PROF. DR. HENRY B. WOLFE
UNIVERSITY OF OTAGO, INF. SCIENCE DEPT.
COMMERCE DIVISION
NEW ZEALAND
VIRUSES: WHAT CAN WE REALLY DO ? *14*

Viruses: What Can We Really Do?

by Dr. Henry B. Wolfe

Introduction

It is virtually impossible to know everything about any facet of computing as it changes on almost a daily basis. Having said that I believe that it is worth sharing some of the knowledge that I have gained as a result of 5 years of study and experimentation with viruses and virus defense strategies as well as having personally tested nearly 50 anti-virus products.

1. Myths and Misinformation.

The greatest disservice that any anti-virus product vendor can perpetrate on their customers is to make them believe that the use of their product reduces ALL risk of virus infection.

NO ANTI-VIRUS PRODUCT REDUCES THE RISK OF INFECTION TO ZERO!

2. An Introduction to Viruses

Computer viruses, although not new, have become notorious for doing damage to many computer systems both micro and main frame. The object of this discussion is to assist you to develop an understanding of how viruses, in general, work and an awareness of the risks posed by them. This understanding will be based on the known techniques of virus design. As time goes on the level of sophistication has, and will, continue to increase and the basic methods and activity of computer viruses may also change with this increase in sophistication.

2.1 A Brief History

Attack software has been around since the late 1950's when at MIT various technical individuals - then known as hackers - used to play a game called Core Wars. Each would write a program designed to take over computer memory and destroy their opponent's control of memory. The one who ultimately gained total control of the computer's entire memory was the winner. Over the years there have been many scenarios developed as to how this type of attack software might be used to damage, defraud, and otherwise interrupt the normal day to day operations of computer systems.

One real life example involves MicroSoft which is one of the largest software houses in the world. One package that they produced was called "Access". The master copy of the package was deliberately infected by an employee of MicroSoft and copies were subsequently distributed to consumers. As the buyer set up the new software a message appeared on the screen - "The weed of crime bears bitter fruit. Now trashing your program disk.". All files on the consumer's computer were lost. MicroSoft originally denied any responsibility until a reporter published an article referring to this event. InfoWorld (October 28, 1985) reported that MicroSoft admitted that one of its programmers had put in a worm, but "without permission" and further assured that the offending code had been removed.

The introduction of the micro computer has made possible experimentation by individuals who in the past would not have had such opportunity. This experimentation has led to the development of a new generation of attack software now known as viruses.

2.2 The Larger Picture

Mainframe and mini computers are not excluded from attack. The nature of their operation makes it more difficult but not impossible. The story of Markus Hess is a good example of one kind of attack. Hess made clever use of the many interlinked networks in Germany and the U.S. He exploited the Gnu-Emacs editor installed on many Unix systems. This enabled him to copy a modified version of "atrun" into the operating system. "Atrun" was then automatically executed by Unix and it assigned super-user privilege to Hess' account. He would then copy the original version of "atrun" back to the system. With super-user privilege he was able to browse, modify and copy files at will in any account within the exploited system. Often the information found in one system lead him into another and so on. It took a concerted effort over several months on the part of many individuals and law enforcement agencies to finally apprehend Mr. Hess and his confederates. In this type of attack the object was to obtain otherwise denied information.

Another type of mini/mainframe attack is more closely related to virus type attacks discussed here. In November 1988, Robert Tappan Morris, Jr. let loose a virus on the Internet system (a network of thousands of computers within the U.S.). This attack made use of a program which infiltrated the computer by taking advantage of the Sendmail electronic mail program (Unix). The virus imported other programs which, when executed, collected usernames and passwords for all accounts on the system. It then used these accounts to infiltrate other nodes on the system. The virus was detected because of a design error which caused it to create copies of itself rapidly. This saturated the network and caused affected systems to crash (approximately 6,000). Most virus attacks are not meant for personal gain. This attack was indiscriminate and not mounted for personal gain. Morris was prosecuted and received a $10,000 fine and required to perform 400 hours of community service.

Attacks on larger systems are less common because there are fewer large systems, access to system files is denied to most users, there are fewer individuals with the appropriate expertise required to make a successful attack, and the means of mounting an attack is restricted by the ability to access a target system.

It should be noted that the larger systems serve many users. User confidence is very important to the integrity of these systems. Many incidents go unreported in order to protect that confidence.

The Apple Macintosh is a very popular family of micro computers. It has not been immune to virus attack. There are many documented viruses which are designed to attack the Macintosh. Tens of thousands of machines have been affected. Scores and "nVIR" represent two of the viruses currently active among Macintosh users. These are applications infectors. The Scores virus was designed to attack specific software packages originating from the Electronic Data Systems (a General Motors subsidiary). The "nVIR" is not so specific and infects all programs within a system.

The Macintosh as well as other unique families of micro computer have been infected by viruses designed to exploit the weaknesses of their respective systems. The techniques employed may vary from family to family of micro computers but the main concepts of how they work are the same across all types of computers. Various estimates suggest that there may be as many as 85 million IBM compatible MS/DOS based micro computers.

2.3 An Assessment of Risk

There are many risks associated with computer virus infection. Some have argued that hardware can be damaged by direct virus software intervention. It may be possible with some hard disk controllers to issue a command for the read/write heads to go to a non existent track. This would cause the read/write heads to crash into the spindle around which the disk rotates. The only recovery from this incident is to have the disk disassembled, repaired and assembled and then hope that the data on the disk has not been corrupted.

Another hardware risk may be to the video display unit. It may be possible to alter the video attributes either temporarily or permanently via the CMOS configuration memory so that the system thinks that it has a different video display unit. Continued use with the wrong attributes could cause the unit to burn out.

A third type of hardware risk is that of excessive use. Some virus strains have been known to increase the number of disk accesses for each access issued. This translates into the premature wearing out of the hard disk due to excessive use (ie. instead of each read or write occurring once, the virus causes it to be done ten, twenty or more times).

Finally, hardware can be made to slow down. This is not the same as excessive use, although that risk does cause a system slow down. Hardware slowdown results in a sluggish system. Hardware slowdown does not necessarily do physical damage or reduce the life of the system components.

Another type of attack is not aimed at destroying hardware or software. It is designed to disrupt normal operation in each session. This technique is exemplified by one strain of the "ISRAELI" virus. Its sole function is to alter all executable files with the extension .COM or .EXE so that they become terminate-and-stay- resident (TSR) programs (".COM" and ".EXE" are the MS/DOS extensions for two types of executable files) . Eventually memory is overflowed and the system crashes.

The next type of risk is to data and software. The most common is loss of data. A large number of viruses attempt to either reformat the hard disk or destroy various components of the vital system files (ie. boot track, file allocation tables, partition table, directories, IO.SYS, MSDOS.SYS, COMMAND.COM, etc.). This usually results in preventing access to the hard disk. While there is some software that, in some instances, can recover from a disk reformat and other types of data attack, the most common result is that the hard disk must be reformatted and all software and data files must be reloaded from your backup.

The second type of data risk is that of data modification. This is much more insidious than the more direct approach described above. In this type of attack application software or key data files are modified such that characters are swapped or one character may be substituted for

another. The outcome is that the computer's sole product - information - becomes unreliable. This type of attack is very difficult to identify because the system remains stable. The hard disk is NOT reformatted nor is there any other readily identifiable sign. This is more dangerous in that at the point the data becomes noticeably unreliable, regular backups may have also been corrupted.

The end result is that decisions which rely on data or computer driven analysis of data will be based on erroneous information. If the manipulated file happens to be a Cash Flow Projection the results could be fatal for a business. If the values modified happen to be distance and altitude data in a flight controller's computer it may mean that two airplanes collide. If the values that have been corrupted are the parameters used in a hospital critical care monitoring unit, the result could be fatal for the patient.

At the very least a virus attack will cause an interruption of normal day to day operations. If a business relies heavily on its computer, this may also lead to a loss of confidence not only from within the organisation but also from its customer base and other business associates.

The types of attacks on computers described above can be prevented. As society becomes more dependent upon the reliable use of computers, it is imperative that their integrity and reliability be protected. Many businesses could fail as a result of decisions made based on computer analysis of erroneous or unreliable data. The continuity as well as the accuracy and reliability of hardware and software performance becomes vital to the survival of business organizations.

3. Virus Makeup

A virus in this context is a program that is characterized by combining the following attributes:

A. The ability of the virus to modify software not belonging to the virus by binding its program code into this target software. This may be specific (e.g. software belonging to one company) or general.

B. The ability to recognize a program that has already been modified, and prevent duplicate modification to programs already "infected".

C. Viral modifications allow infected programs to assume the above attributes - ie. to be able to reproduce.

Most viruses are made up of at least two parts. The first part is the marker bytes or signature. These bytes identify to the virus program that a target program has already been infected thus preventing it from being infected again. The second part is the virus kernel. This piece of code contains the routines necessary for reproduction. The virus first checks to see if the target program has been infected. If it has not, then it attaches these two pieces of code onto the beginning or end of the target program. The next time the target program is executed the virus will be activated.

Other more sophisticated types of viruses achieve a similar end but they do not corrupt the target program nor do they necessarily have a unique signature. These viruses append their code to the target program thus expanding its size. Their mission is to lay dormant, infecting other programs where possible until a triggering event. When the triggering event occurs, the

virus then executes the part of its code that performs its dormant or primary function. These functions may be as innocuous as displaying a message on the screen or as harmful as burning out the video display unit.

In the July/August issue of Computer Security (The Newsletter for Security Professionals) mention is made of a new virus approach. The report comes from England and describes a two part virus. Each part is a virus that alone can do its damage. When one detects the other's presence on the host system, they combine or mate to produce offspring. The offspring are new, distinctly different viruses and can infect systems in their own right. Incestuous mating with their parents apparently produce other different strains of the virus.

4. How Viruses Work

Viruses have a number of features in common. I will now attempt to describe the inner workings of viruses in general.

The first thing to recognize is that a virus must be executed in order to propagate and do its damage. In an MS/DOS PC environment executable files are commonly suffixed with the extensions - .COM or .EXE. This is not an exclusive condition and it is not commonly known, for example, that Lotus 1-2-3 worksheet (.WK1) files may also contain virus code.

Once the system is infected the virus program hides itself in one or more of the following places:

A. The Boot Sector.

B. System Programs eg. MSDOS.SYS, IO.SYS, COMMAND.COM, etc.

C. Applications Programs eg. .COM or .EXE files.

D. Companion viruses create a .COM version of each .EXE file. Virus code resides in that .COM file. Because the hierarchy of MSDOS dictates that .COM files are executed BEFORE .EXE files, the companion virus will be executed first and the .EXE version is then called from that program -- AFTER the virus has become active.

E. Lotus 1-2-3 Worksheet files eg. .WK1 files.

F. In the case of A & B where the code size cannot be increased, extra code may be placed in sectors flagged as "BAD" on the hard disk or in the unused portion of sectors used by other programs or data.

In order to understand how a virus works it must be assumed that it has taken control of the computer by being executed. The first thing an application infector does is to search out other executable files. It does this by browsing each and every active disk unit attached to the computer. Within each unit it searches each and every directory and subdirectory until it finds a .COM or .EXE file.

When the virus finds an executable file, it analyses that file to determine whether or not it is already infected. This may be done by looking for the virus' signature or marker bytes. Most viruses will have a unique signature. If the executable file has this signature, the virus recognizes the infection, ignores that file and moves on. The virus then proceeds to search the current directory and disk for another executable file. The process is repeated until it finds an executable file that has not been infected or until all files in all directories and sub-directories on all disks have been browsed and analyzed.

If the executable file is not infected, the virus proceeds to infect it. This is done by implanting its marker bytes and either overlaying virus code in the executable program or by appending that code to the end of the program.

One of the ways that some viruses can be detected is by the fact that program files increase in size. Some viruses such as the 1701 and the 1704 viruses have been named for the amount that they increase infected programs by.

If the virus is a boot sector infector it attempts to locate each and every boot sector for each and every disk unit attached to your system. If it is successful in locating a boot sector it then modifies it so that at bootup time the system will load the virus as one of the first programs active in the system. It would be likely that it would remain active thereafter.

Virus code can also be hidden in flagged BAD sectors on the hard disk. Some viruses have the capability to store their malicious code on disk and then flag those sectors as bad blocks. DOS will not see that there is information hidden there. It will only see that the blocks are bad and otherwise ignore them. When the virus once again gains control, it can then retrieve the code stored there by issuing an absolute read of specific disk locations. Because the blocks are not bad or damaged, the read will be successful.

The final step that the virus carries out is to check for its triggering event. That event can be just about any condition or combination of conditions. It can be the number of infections. It can be the number of times that a particular command, such as TYPE, has been executed. It can be that the attack part of the virus is executed on a particular date; for example, Friday the 13th. This is left up to the imagination and ingenuity of the author.

If the conditions are not right for triggering the attack part of the virus, then it becomes dormant. In most cases it still sits in the background monitoring and checking for new uninfected executable files and for the triggering conditions.

If all of the conditions have been met for triggering, the virus executes its code. Typically this results in the loss of access to the hard disk. That usually means that data files and programs are either lost or destroyed. Some viruses rewrite the boot sector (a clear definition of the boot sector and its use will be found in Appendix B). Without a valid boot sector the computer cannot boot from the hard disk.

Some viruses also destroy the partition table. Without the partition table, the machine thinks that it has no hard disk and, therefore, the hard disk becomes inaccessible.

Some viruses destroy system files such as MSDOS.SYS, IO.SYS, and/or COMMAND.COM. If any of these files are destroyed, the system will not boot from the hard disk.

Certain viruses delete every program called for execution, or data files opened for access. Still other viruses attempt to do a low level format of the hard disk. If it is successful, all of the files on the hard disk are lost. It will also not be bootable.

This, in basic terms, is how a virus works. Most follow the same form but have different agendas and different triggering conditions. There is, however, one other method by which a virus can get onto the system which has not yet been addressed. That is via electronic mail. Many micro computers now have the capability to connect to larger mini's and mainframe computers. In this mode the larger machine has control of many functions of the micro computer. Viruses can be passed to the micro through this type of communication. The virus still must take control of the system before it can further infect and become a problem.

This introduction has endeavored to provide an overview, in less technical terms, about how viruses work and also to describe a few of the more common varieties currently circulating amongst micro computer users on a worldwide basis. There are several hundred viruses that have been identified and analyzed. Of those the most common variety (about thirty-five percent) are boot sector infectors.

5. What is the actual threat?

Infiltration routes are pirated software, bulletin boards, shareware, floppy disks supplied by computer magazines, public domain software, shared PC's (at home), service engineers, and shrink-wrapped software.

There have been at least three documented cases where prominent computer magazines have unknowingly included software on disks as a promotion bonus (infected disks). One included a mutation of the Stoned virus, another included Disk Killer and the third was infected by the Cascade. Needless to say, one should be careful when making use of software provided "free" with the computer magazine of your choice. If you intend to make use of software obtained from these sources it would be in your interest to scan them first with a good virus scan program.

A large number of executives use their home computer for business activities. Quite often this home machine is also used by their children. There are many documented cases where the executive has done his work at home and taken the finished disk back to the office only to infect other PC's within his organization. Be wary of this possibility. Good enforced policies can deter this or at least reduce the risk to your organization.

Service engineers have also provided some organizations with little surprises. They often carry their own diagnostic disks and can easily pick up and pass on viruses moving from customer to customer. To avoid this risk you can insist that all diagnostic disks (or any others required by service engineers) are provided by your organization, that those disks are copy protected, and under the control of and held by a responsible individual within your own organization.

Shrink-wrapped software implies that the product came directly from the vendor and is clear of virus infection. There have been documented cases of shrink-wrapped software arriving already infected. In most of these cases the product appears to have originated in Taiwan - but not all. The risk from this avenue is admittedly small but even products that are shrink-wrapped should be scanned to further reduce that risk.

IBM's Thomas J. Watson Research Center, has compiled statistics relating to actual reported virus incidents. This topic is important as it may help the reader to assign a level of real risk from computer virus infection and then act accordingly to reduce that perceived risk. IBM has over 400 different viruses in their collection. Of these only 60 were reported in actual incidents. The rest may only be in the collections of researchers. These statistics showed that 72% of all reported infections were accounted for by only seven viruses (the New Zealand or Stoned virus is by far the leader and accounted for more than 25% of reported infections). Unfortunately, no one is willing to estimate how many PC's had been infected during the past year.

All of the anti-virus developers share one thing in common. They have developed a product that they believe is infallible and provides adequate protection. While their initial intentions are unquestionably to provide, to their clients, maximum protection from virus infection, the biggest disservice that they can do is to encourage their clients to believe that they are now safe from virus infection and that there is no longer a significant threat.

Anti-virus protection can be achieved but it must be done in a considered and logical way. That means making use of anti-virus products, enforced policies and common sense. All of this should be tempered with the knowledge and understanding that there will still be a risk. Even more important is the need for users to be educated as to what is at risk and its possible impact on the organization and their job security.

6. Classes of anti-virus products - strengths and weaknesses.

There are several classes of product each with their own inherent advantages and disadvantages. Some of these are: hardware only, hardware software combinations; software monitors, shells, scanners, checksummers, and removal programs. Without exception, all require that procedures for operation be strictly adhered to. Each has its place in the scheme of risk reduction. It is intended that after reading this the user will be better informed as to the types of anti-virus strategies offered in the marketplace. This knowledge should provide a basis on which to select the strategy or combination of strategies that is appropriate to their environment or organization.

6.1 HARDWARE ONLY: This type of product is normally used on a quarantine system. A quarantine system is a PC which is used exclusively for testing INCOMING software and would usually be found in larger organizations. Ideally, the "quarantine" PC would have two hard disks as well as a special circuit board (such as Write Guard) which provides the capability to copy protect one of those hard disks. The operating system and other permanent software would reside on the protectable drive (C:).

Incoming software would be installed on the unprotected drive and used in the normal way. If an alarm is raised during normal operation of the new application software (due to an attempt to write to the protected drive) one can reasonably assume that the executable being tested is either infected (and trying to infect the protected drive) or that it normally writes to other unspecified drives. The latter is far less likely and raises serious questions about the product itself.

This type of product can provide excellent protection in the form of verification that incoming software is not infected. The risk with this defense, as is true with all other products, is people and requires that procedures be strictly followed, policed and enforced.

6.2 HARDWARE/SOFTWARE COMBINATION: This combination offers a strong option when well designed. This type of product includes either a circuit board (such as Thunderbyte) or plug that attaches to one of the I/O ports (such as PC-cillin) and software which is tied to that device. These products attempt to defend against common as well as "stealth" type viruses (those viruses that make direct calls to the BIOS rather than going through the interrupt system as the majority of software does) and ensure the integrity of the anti-virus software by providing incorruptible executables in circuit form.

This product would normally be installed on each PC and provide individual protection for that PC. Once again, the main risk with this type of product is people as outlined above.

6.3 MONITORS: This software (such as CompuGUARD or Protec) is memory resident (TSR - terminate and stay resident) and watches various activities during execution of all other programs on the PC (such as interrupt calls and I/O processing). Many viruses can be caught in this way. This class of product may or may not be virus specific. If the monitor is signature based rather than strategy based - it will recognize the signatures of known viruses only and will not detect those viruses for which it has no signature string.

This type of product can easily be circumvented by some viruses. Direct calls to the BIOS are not normally detected by a strategy based monitor and are definitely not detected by signature based monitors (unless the virus has a known signature). This type of product provides somewhat less protection for the user.

6.4 SHELLS: This product modifies DOS and/or places itself between application programs and DOS (such as Certus or Jarrin). This makes it TSR and it oversees activities initiated by executables watching for telltale attack strategies. Some shells also incorporate an executable checksum that is checked prior to running each application. If the checksum is identical to that program's checksum on the master list (created when an executable is installed on the PC) then it is assumed that it has not been modified (infected) and execution is permitted. If the checksums are different an alarm is raised and the user must react.

This type of product can provide a high level of protection when checksums are used in its design. The main risk with it is when procedures are not followed - once again people is one of the main weaknesses.

6.5 SCANNERS: This type of product (such as VIRUSCAN or F-PROT) is based on the fact that some viruses contain unique strings of characters (signature). The use of signature strings is one method used by virus writers to enable an incoming virus to recognize when it has already infected an executable. If the virus could not do this it would repeatedly infect executables until it filled all available disk space. Scanners make use of this technique and search all executables comparing for known virus signatures. When a known signature is found the scanner reports the fact and identifies the virus and the user must react accordingly.

Scanners, probably the most commonly used class of anti-virus product, provide a degree of protection but it should be noted that there is a whole family of viruses that have no signature. This type of virus is self encrypting and will be stored in a different form on each PC. Most of these cannot be detected by scanners. It is also important to note that viruses are being created at approximately one every two to three days. By their nature, scanners are almost never up to date and will only be able to "see" those viruses that have signatures in their list (the current number of viruses and variants ranges from 2,700 to over 3,000 depending on who you talk to). There is no definitive all-inclusive collection of viruses and, therefore, a scanner's success depends on the collection used by a particular vendor to develop their signature list.

None of this negates the importance or usefulness of signature based software products and the use of a scanner is usually the first line of any defensive strategy. But the user needs to be aware of the inherent weakness or limitations when using this class of anti-virus product and not be lulled into thinking that they are foolproof.

6.6 CHECKSUMMERS: This type of product makes use of CRC (cyclic redundancy check) or proprietary algorithms to create a numeric value called a checksum (such as V-ANALYST). This value is recorded in a database (file) which is used by the detection software. Each time the system is powered up (or at other user specified times) the detection software checks all or selected executables (by calculating a new checksum and comparing that number against the appropriate value in the checksum list). If any do not match an alarm is raised and the user must act accordingly.

This class of product can provide a high degree of protection - if procedures are always carefully followed. They do not provide for detection or identification of incoming viruses. They only identify change and that only after the fact.

6.7 REMOVAL SOFTWARE: Some anti-virus products also provide the companion facility of being able to remove or disinfect those executables that have been infected by identified viruses. This capability can be important in after the fact situations (one or more executables are already infected) where there is no backup copy of the infected file. They are designed to handle specifically identified viruses and either remove the code added by the specific virus or sever the linkages required by the virus to operate. In either case the software is returned to a state that is usable without being able to infect other executables. This class of product provides no protection but does provide an important repair capability for those faced with the task after a successful virus attack.

6.8 COMBINATIONS: There are a large number of products that offer various combinations of these techniques (some of the products cited have multiple capabilities as well). This is not meant to be an all inclusive list of products nor product types. There are also inoculation programs and products that encrypt an executable so that it will not execute if it has been tampered with.

Every individual or organization has different requirements and environments within which protection is necessary. I believe that a combination of two or more of the techniques described above will provide protection to an acceptable level and have implemented such a scheme in my own environment. One key to successful protection is training and indoctrination. It is important that PC users know what is at stake and how each of them can reduce the risk of virus infection.

7. Backup and recovery from a successful virus attack.

Successful recovery from a virus attack can be attributed to two basic policies. The first is backup. Without proper and adequate backup, complete recovery from many types of virus attack will not be possible. Important data will be destroyed and no utility or amount of clever manipulation will be able to retrieve it. Without backup it will be lost forever. The basic policies should cover the backup and storage of original software. In most cases this need only be done once for each software version installed. Regular backup of data files (non-executables) should be performed at intervals that make sense for the organization and the application. In some cases backup of particular data files could be daily or hourly. In other cases it might be weekly or monthly or some other interval. These backups should be scheduled in a formal way to ensure that they are carried out. Backup schedules, logs and procedures need to be formally documented. These procedures must be reviewed regularly so that the ever changing computing environment is protected. Moreover, the use of backup logs and policing of those logs can ensure that adequate backup is in place to allow recovery from virus or other types of computing disaster. This type of protection is often overlooked or forgotten. If it is, eventually the organization that fails to keep adequate backup will pay a price for not doing so.

The second policy is adequate recovery procedures. These procedures should be carefully thought out and described in detail - perhaps in cookbook style - so as to assist non technical personnel in the recovery process when specialists are not available to take charge. The instant that an event is detected, it should be reported to the individual responsible for this process and the system on which the incident has been detected should not be used for further work. Where it is connected to other systems, that connection should be immediately severed. The designated person should then be given control of the situation and allowed to carry out the recovery procedure according to the formalized procedures or at least supervise this activity. It is very important to keep the amateur "experts" out of the loop. In most cases they cause more problems than they solve.

8. The wildcard in the equation - people.

Inevitably, people (your staff) either make a policy successful or cause it to fail. The key to success in this case is to involve them all in the virus protection process. This can be done with a good training program and with regular reinforcement sessions and updates which keeps the important issues before them. **If you are not involved then you don't care or see the need for proper security procedures!** If you don't understand the consequences of a successful virus attack and cannot see what the impact may be on your job security then prevention policies will seem remote and unimportant. The same is true for backup and recovery procedures. This aspect of computer security should be clearly linked to virus attack training programs so that all staff can see the importance and value of the proper and regular execution of backup and recovery procedures.

9. Procedures and policies for safety.

In summary, three general rules that can reduce the risk of virus infection are:

A. Never Boot from a floppy disk.

B. Do not practice software piracy.

C. Make use of write protect tabs on floppy disk.

These precautions alone will be enough in many environments but with the rising popularity of data communication and bulletin boards they will not suffice. Additional precautions will have to be implemented to ensure the continuity and integrity of micro computer system performance.

Remember, that adequate regular backup is a must in any business environment. All of these policies work in concert. To implement only one will provide little in the way of risk reduction.

Specific policies and procedures must be formalized, put in place and enforced/policed. Without that enforcement, the best protection products available can fail.

10. A suggested method of detection and defense.

In my own computer operation, I use a combination of products. I have set aside one PC as a quarantine machine and have WriteGuard in place on the C: drive. As software is received, I first scan it for known viruses and use both VIRUSCAN and F-PROT (if one doesn't find it the other one will - or at least that's my theory). The vast majority of common viruses will be caught immediately by this step and reduce or eliminate the time needed to install and test it on the quarantine machine. If no alarm is raised during the scan process, I then install the software with the C: drive protected. If after a reasonable period of operation, the product raises no alarms, I then wipe it from the quarantine machine (with a low level format of the second hard disk) and install it on the intended PC. I am well aware that it is possible to evade even this detection process by building a time delay into the virus so that the replication imperative is not immediate.

IN SUMMARY:

Throughout these discussions we have tried to offer suggestions that will help those not familiar with computer virus attack. We have discussed some history and described various virus strategies. The main types of anti-virus products have been outlined including the pros and cons of each. Hopefully, after these discussions, the reader will be in a position to make an informed choice of anti-virus products and strategies. We have described some basic rules or procedures, that can be followed easily, which will reduce the risk of virus attack.

If the reader takes away a better understanding of the computer virus phenomenon, then this document will have served its purpose.