Close

Copy

Licensing Agreement

Close
Copy

InVircible, ResQdisk and *ResQpro* are trademarks ™ and copyright © 1990-96 with all rights reserved to *NetZ Computing Ltd. (NCL), Israel.* You, as the purchaser of license rights to the software, are herein referred to as the *Licensee*.

The installation or use of this software package, or part of, constitutes acceptance and full agreement to the terms and conditions of this warranty and licensing agreement, in so doing the Licensee agrees to be bound by said provisions. This is a legal agreement between the Licensee as the end user -- and NetZ Computing Ltd. who is the owner of all rights to the software. Usage of the InVircible software indicates acceptance of the terms and conditions.

License: InVircible is licensed, it is not sold. The Licensee is obtaining limited rights to use the InVircible software. The Licensee is licensed to use InVircible on a single personal computer. Site licenses are available for individuals or organizations with multiple installation requirements.

Site License Certificate: The Purchaser of a Site License will be issued a License Certificate by NetZ Computing. The holder of the Certificate is entitled to free upgrades of IV during the License Term, as well as hotline support through a NetZ Authorized Agent. The possession of the NetZ License Certificate constitutes the sole proof of a genuine and legally acquired Site License.

Alteration or reverse-engineering of the programs is not authorized, such attempts terminate this license agreement and may result in unpredictable damage and may have irreversible effects.

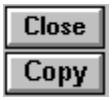
Limited Warranty / Limitation of Remedies: Defective diskettes will be replaced, at no charge, if they are returned freight pre-paid, within 30 days of the original purchase date. The sole remedy available to the Licensee will be limited to the replacement of defective diskettes as outlined, or a refund at the purchase price of this software program package.

InVircible is provided "as is". The Licensee assumes all responsibility for the determination of the suitability of this software for Licensee's configuration and the results and performance of this software program package. NetZ Computing, the distributor, and their suppliers do not warrant, guarantee, or make any representations regarding the use of, or the results obtained with, the program in terms of correctness, accuracy, reliability, legality, or otherwise.

Disclaimer: In no event shall NetZ Computing Ltd., nor its Agents, nor anyone else involved in the creation, production, or delivery, of this product be held liable for any damages, loss of profit, consequential, or other damages, that may arise out of the Licensee's use or inability to use the InVircible programs. Some states do not allow excluding or limiting implied warranties or limiting liability for incidental or consequential damages, so the above limitation may not apply to the Licensee. This limited warranty

and license agreement is governed by the laws of the State of Israel.

Using the Manual



This hypertext manual provides essential information for the installation and use of InVircible. In addition to background and program documentation, the manual provides guidelines for minimizing ongoing threats to the well-being of your PC's and network health.

InVircible is user friendly, intuitive and will provide long-term virus protection. After installation InVircible functions unobtrusively without the need for ongoing user actions.

Most of virus attacks will be detected and removed from your PC by just following InVircible's instructions on the display. For those situations where complex or compound virus attacks are detected, InVircible's utilities and this manual's guidelines provide for a user directed thoroughly complete restoration.

InVircible may be installed and used with assistance only from the online user help, so many InVircible users operate without the manual. A Quick Start section in this manual guides the user who prefers to install immediately and review the manual in detail later.

Commands throughout this help file are displayed in **magenta bold typeface**.

Browsing and Printing

The InVircible on-line manual for Windows is organized in five browsing sequences for convenient reading and printing with the **<<** and **>>** buttons, available in your Windows Help system.

The *Main* browsing sequence will take you through a quick starter, the installation of IV in various environments, the preparation of the rescue diskette, how to use the InVircible menu shell - **IV**, the integrity module - **IVB**, the correlation scanner - **IVX**, the disk recovery module - **ResQdisk**, and the virus scanner and remover - **IVSCAN**. Common virus problems and the way to handle them are also discussed here.

The *Virus* sequence will take you through a primer on computer viruses, their various categories, what they are and how they work and what their risks are.

The *Anti-Virus* sequence surveys the various antiviral program types, what is generic virus capture (detection) and how it works, the basics of generic integrity analysis and recovery, what is auditing and how it helps in maintaining systems integrity, how virus cooperative methods work, and what are virus analyzers and how they work.

Integrated *Anti-Virus* and multilayered antiviral protection is explained in detail, you will also find suggestions how to optimize your antivirus strategy in the stand alone PC and *network* environments, specific generic antiviral methods are discussed, and virus screening methods for new software are explained.

The *Anti-Virus* sequence ends with a discussion on viruses and antivirals in the non-

DOS environment (OS/2 and Windows NT), antiviral TSR and blockers, the inoculation of application programs and how to protect from the new Word macro viruses. There is also a topic about *hands-on training* for yourself and computer help desk personnel in antiviral strategies and recovery techniques.

The *HD and data recovery* sequence covers in details the methods and techniques used by InVircible's ***ResQdisk*** and ***ResQdata*** to recover hard disks from virus infection and catastrophic damage (loss of partition or total loss of access) as well as the recovery of files. Among the many topics covered are: IDE / EIDE configuration setup and recovery, troubleshooting hard drive access problems, the reconstruction of the MBR and of the boot sectors for DOS and Windows 95, special configurations such as Dynamic Drive Overlay (DDO) and Compaq non-standard disk configuration, fixing damage to file and directory structure and how to improve your data survivability and recoverability.

The topics covered under *hard disk recovery* are a **must** for computer help desk personnel, computer maintenance providers and data recovery experts. The implementation of some of the methods described require the registration of the professional version of InVircible - ***ResQpro.***

The *Miscellaneous* sequence starts with the licensing agreement and provides useful information such as additional antiviral utilities and where to find them, product upgrades and support and a full list of command line options and syntax for all the InVircible modules.

A Quick Starter

Close
Copy

Fast Installation

Insert the distribution floppy into drive A: or B:. To start the quick installation procedure run **INSTALL /FAST** from the distribution floppy.

The InVircible INSTALL first examines your system for viral activity and checks for boot and file infections of the hard drives. After these initial tests, InVircible then installs itself on your system to a default directory and the IV commands will be added to the **autoexec**. Next, all files on the hard disk drive partitions will be "secured" by IVB. This initial scanning and securing of a large number of files may take a few minutes, so please be patient! Whenever a choice is provided, the default selection is the recommended choice.

The preparation of a Rescue Diskette is critical for effective recovery from disk crash and viral attacks. The preparation of the rescue diskette is needed before a disk incident or virus infection may occur. Therefore, prepare the floppy immediately after the installation was completed and the computer was rebooted.

Routine Use of InVircible

IVINIT will run a quick check on any initialization of the computer on the integrity of the boot block and the operating system. IVINIT can also remove most boot viruses right at start up, after prompting for your permission.

Next, IVB will automatically run a DAILY check of your programs in the local partitions on the installed hard drives.

IVB is the primary tool for monitoring the integrity of files and for their recovery in the event of a virus infection. IVB will report changes in binary-executable files. Changes are expected if you replaced a program's older version with a new one and the filenames are reused. IVB will suggest the renewal of the database file in case the changes are a new version of a program. The current signatures file will be backed up when renewing the signatures file, thus you may always correct a mistake one stage back, in case the changes were viral.

It's good practice to screen newly acquired software with an up-to-date on demand scanner. Off-line screening with a scanner need to be run just once, before installing the software to your hard drive or to a file server. Ongoing and on-line protection is provided by InVircible. It's worth scanning new software floppies with IVSCAN in addition to a third party scanner as IVSCAN provides additional generic capabilities not found in common scanners. It may detect generic infectors that were missed in the pre-screening process. After installing new software to the hard disk, run IVB to secure the newly added executable files.

Application programs do not change normally unless they have been upgraded. A **consistent change pattern** reported by IVB usually indicates a viral infection. IVB

provides this information both on screen and in a report file (IVB.RPT) placed in the partition's root directory. A report of only a single file having become modified requires you to suspect a false alarm. Viruses rarely infect only one file, so an indication of a single modified file may not necessarily indicate the presence of a virus. In most cases it is not a virus doing! Furthermore, virus activity usually gives itself away by several indications such as memory stealing, baiting, spoofing or Piggybacking.

When a genuine infection is found, then boot from the clean rescue diskette and first assess the extent of the infection, before taking any corrective actions. IVB should be preferred whenever possible for removing viruses from programs, provided the programs were secured before becoming infected. Only as a last resort, use IVSCAN or a third party scanner to disinfect.

Completely new viruses can still be found using IVX and be renamed so that they will not constitute a threat to your computer and applications. The renamed files can then be replaced from backup.

A Primer On Computer Viruses

Close
Copy

What are Computer Viruses?

Computer viruses are computer code written by real programmers, there is nothing mysterious about them. There exist already thousands of them - more than 8000 at the beginning of 1996 - and several new ones are written every day. They have to be purposely designed and written, they don't just appear from nowhere, neither evolve or 'mutate' in a spontaneous process. Being some sort of graffiti, the purpose of their writers is to spread their 'work' to as many as possible computers. The basic and common characteristic to all computer viruses that distinguish them from other programs is their ability to replicate and spread from one machine to another. Replication is what gave viruses their name.

Computer viruses are written to operate in a specific operating system environment. There are DOS viruses, Unix, Mac viruses, Windows and even Windows 95. To this date there aren't any Netware specific viruses nor other net operating system's viruses. The total majority of computer viruses, and the ones to be concerned about operate under DOS.

Computer viruses are passed the same ways we obtain software; by sharing software, downloading programs from the net, with newly purchased PC and hard disks, in vacuum wrapped new software, and sometimes even on preformatted floppies.

Yet an infection will start spreading in a computer only upon the execution of an infected program or boot sector. The most common way (70 to 90%) of how an infection starts is by attempting to boot from a boot-infected floppy. The floppy may contain just data, it need not being a bootable one, just enough that its boot sector was infected. A computer with an infected hard drive will infect other floppies. Just reading the floppy in an infected machine's drive suffices for infecting its boot sector.

Infection by a file infector will usually occur upon running an infected program. Additional programs will become infected either upon execution (memory resident viruses) or by executing another, yet infected program (direct action viruses).

Virus Risks

Close
Copy

Contrary to common belief, computer viruses almost never cause immediate and irreparable damage to programs or to data. Most don't even cause irreparable damage, ever. Viruses insert their own code into boot and partition sectors and into programs in a way that **preserves the functionality** of their host. Functional code, even if treated by a virus, retains all its pre-infected data and code which makes the removal of the virus possible.

Since the purpose of a virus writer is to get his virus spread then it should not give itself away. A prematurely destructive or disruptive conduct will disclose the virus presence and the user will notice and take action that will stop further spreading of the virus. Therefore, destructive viruses are not common and not found in the wild. It also implies that those found in the wild are necessarily the least offensive, as they could spread unobstructed.

In result, the more successful viruses found in the wild are all 'well-behaved' and are therefore recoverable.

Viruses, although not all of them may have a payload that activates on predetermined conditions. For example, Big Caibua displays an obscene animation at certain hours in the afternoon, Natas has one chance in 500 to trash the hard drive on booting, Michelangelo trashes the first 255 cylinders of the hard drive on March 6 every year and Emmie commits suicide by trashing the partition sector on any date in 1993.

Yet most viruses have no destructive payload at all and are more of a nuisance than a real threat. For example, the 'Concept' Word macro-virus that affects Word documents causes no damage, it just adds a reproducing (hence a virus) macro to Word's global template. Others may interfere with the proper functioning of your programs, although they weren't intended to do so. Boot viruses tend to interfere with the functioning of Windows and some will prevent Windows from loading or hang.

While your computer may continue functioning for a long time being infected with a virus without even noticing, you may cause severe damage and even lose access to your programs and data by an unsuccessful attempt to remove a virus. MBR infections may easily turn into real disaster while trying to remove them. Unfortunately, boot-partition infectors are also the most prevalent ones, from seventy to ninety percent of all viral incidents.

File viruses tend to cause more massive infections than boot infectors, after all there is just one boot or partition sector per drive, while there may be thousands of executable files on the hard disk or on the network server. Yet the damage they cause is lesser than boot-partition infectors. A file infector won't cause loss of access to the hard drive, unless it contains a deliberately destructive payload. As stated, deliberately destructive viruses aren't widespread and it's very unlikely that a destructive virus will infect your computer. Destructive viruses get extinct shortly after being released as their replication is a decaying process. Similarly to boot infectors, the removal of file infectors by

inappropriate methods or means can cause a great deal of work in restoring the computer or server back to normal operation.

The risks mentioned emphasize the need for reliable virus detection and dependable recovery means. This is all that InVircible is about.

Categories of Computer Viruses

Close
Copy

Most viruses belong to one of the following two categories:

- Viruses that infect through the boot sector or partition sector (MBR), generally known as '**boot infectors**';
- Viruses that infect through programs, or '**file infectors**';

There are also **bipartite** or **multipartite** viruses that combine the properties of these two categories (boot and file). Examples of multipartite are *Junkie* and *Natas*, while *Tequila* is a bipartite as it does not infects the boot sector of floppies, just the MBR of the hard drive as well as EXE files.

The **cluster infectors** emerged in late 1991. These viruses manipulate the file allocation table (FAT) pointers through the directory entries in order to spread. *Necropolis* (1963) and *Dir-2* are cluster infectors and in 1995 *Dir.Byway* was added to this category.

A new type of viruses could be application specific **macro viruses**. It is yet unsure whether the Winword macro viruses are the first of a kind or will remain an episode in computer's virology.

A last type are the '**companion viruses**', these are a variant on the file infector. A companion virus takes advantage of the properties of the operating system by spawning a companion to the victim program.

Below is a quick definition of terms that are used to describe viruses and techniques employed by viruses:

Boot Viruses

Boot viruses attack the hard drive's boot sector, the master boot record (MBR, sometimes referred to as the partition sector) and/or the boot sector of floppies. When the computer is turned on, it runs a **bootstrap** program contained in these sectors, first the partition bootstrap and then the system bootstrap, to ready itself for work. In case of a boot infection, one of these programs may simply be a virus. Boot viruses will remain active in memory while the computer is on. During this time they will infect write enabled floppies put in the floppies' drive.

File Infectors

A file infector attacks executable program files, usually those having a COM or EXE extension name. Sometimes also files having the structure of an executable are targeted by viruses, regardless of their extension name. File infectors may corrupt non-executable files as well but they cannot spread this way.

Many file viruses are **memory resident** (TSR). After an infected file is executed, they will remain resident in the computer's memory until the computer is turned off. While in

memory they will continue to infect other programs and may interfere with normal operations. If the computer is turned off they will lie dormant in an infected file until the program is executed and then load themselves back into memory until the next time the computer is turned off.

Multipartite Viruses

Multipartite viruses infect both executable files and boot-partition sectors, sometimes the boot sector on floppies too. Some multipartite become infectious only after rebooting the computer from the infected MBR, like *Tequila*, others are equally infectious if loaded from file or through the boot process.

Cluster infectors

The cluster infectors are basically file infectors, with a unique infection mechanism. Cluster infectors are inherently **stealth** viruses because they hide the increase in file length, although the virus code is actually appended to the file. The way it is done is by inserting the extra clusters in the file's clusters chain, by manipulating pointers in the directory entry. Two well known such viruses are *DIR_II* and *1963* (Necropolis). DIR_II is almost extinct as it does not survive in DOS 5 and higher environment. Cluster infectors are also excellent **piggybackers**.

Winword Macro Viruses

Close
Copy

A macro virus is a piece of code written in an application's macro language, that is capable of replicating itself. Macro viruses will reproduce only if the automatic execution of macros is enabled in the targeted application. By definition, macro viruses are **application specific**, to distinguish them from DOS viruses.

Unlike DOS viruses, that can use DOS and BIOS services (documented as well as undocumented), application specific viruses can only use commands that are available in the specific application's macro language. Not every application that uses macros is exposed to macro malware abuse, to this date only Winword 6 has been successfully targeted by macro viruses.

Applications' macro language is rather limited compared to the set of instructions available to the operating system as it should only provide for the application to perform its tasks. Winword is not expected to configure a hard drive or to manipulate the boot or partition sector. Although not absolutely impossible, the formatting of your hard drive by a Winword macro virus is unlikely.

Another distinction of the Winword macro viruses from the DOS ones is that macro viruses affect **data** rather than the application itself. To read how to protect from Word macro viruses click here.

Techniques Used by Computer Viruses

Close
Copy

Stealth.

Stealth viruses are so named because they actively seek to hide themselves to prevent detection. **Stealth** viruses are categorized as '**semi**' and '**full stealth**'. Semi stealth viruses will subtract the size of the virus code (in bytes) from the infected host file at any time that a directory (DIR) command is executed. The full stealth ones will remove themselves from the file, to re-infect it again when the inspection is over.

Stealth viruses are successfully handled by generic methods, both for detection and recovery.

Some boot viruses use stealth techniques too. Stealth boot viruses will spoof disk viewers and editing tools, except for IV's ResQdisk. Examples of stealth boot viruses are *Monkey, AntiEXE* and *B1-NYB*. Natas and Gingerbread Man are stealth multipartite viruses that use boot stealth too.

Boot stealth is successfully handled by generic tunneling methods (SeeThru) for both the detection and recovery of the affected areas.

Encryption

Most of modern viruses use encryption to make their presence less obvious. An encrypted viruses usually has a short common encryptor with the rest of the virus code being encrypted. The detection of encrypted viruses cause high false alarm rates due to the ambiguity in the detection of the short common encryptor string.

Encryption is easily handled by generic integrity detection and recovery methods.

Polymorphic Viruses

Viruses that use **variable encryption** are said to be polymorphic. As their name implies, the virus code differs from copy to copy, but this is also true for plain encrypted viruses. Yet in polymorphic viruses, the encryption algorithm changes also from copy to copy. Polymorphic viruses presented a new challenge to scanners as instead of looking for relatively simple signatures they need to look now for more complex and fuzzy patterns. **Heuristics** is used in many scanners to detect the polymorphic viruses. This increased their false alarm rate and also rendered scanning more difficult and slower.

Like with signatures scanning, heuristics are also based on prior knowledge on the mutation engine used for creating a specific virus. The mutation engine is that part in the virus code that is responsible for 'mutating' the encryption algorithm. There are several such encryption engines, created by individuals and virus writing groups. The most known is the Dark Avenger's mutation engine, also known as *MtE*, created by a Bulgarian virus writer that authored several viruses in the late eighties.

Polymorphic viruses are easily handled by generic integrity detection and recovery

methods.

Companion Viruses

Companion viruses take advantage of the precedence DOS gives to COM over EXE files in the order of execution. If there are two files bearing the same name, one with a COM extension name and the other with an EXE extension, then DOS will execute the COM file first. A companion virus is basically a Trojan that "infects" EXE files by **spawning itself** into a companion COM file, bearing the same name as the EXE file. Sometimes the companion virus file will have its attribute set to 'hidden', to avoid its detection by the DIR command. A variation on the theme is the **path companion**. The difference between the latter and an ordinary companion virus is that while the latter companion is spawn in the same directory as the targeted EXE, the path companion may have its spawn in any directory contained in the path that precedes the targeted directory.

Both types are handled successfully by IV's generic methods.

Anti-Virus Program Types

Close
Copy

Antivirus (AV) programs belong to one of the following categories: on-demand scanners, TSR scanners, activity blockers, and generic AV. While scanners and blockers are generally understood there is some confusion about what is meant by generic AV. For most users, as well as the majority of computer security experts, generic AV brings to mind only 'integrity checking.'

The term generic, from the Latin 'genus,' implies that a method is applicable to a group or kind, and exclusive of others. While checksum integrity checking, CRC included, may apply to viruses as well, they are not inclusive to ONLY viruses. The checksum or CRC of a file, or of a program may change for many reasons most of which are not connected with virus infection. A few common, non-viral reasons for the changing of a file, and of its checksum or CRC, are its replacement with a newer version, self configuration, and plain corruption in result to truncation and cross linking. There are other causes for file changes, as well.

InVircible's integrity scheme focuses on virus-exclusive integrity analysis rather than on checksumming. It provides IV with higher probability of detection of genuine virus activity, combined with much lower false alarms that plague the ordinary integrity checkers.

Generic Virus Capture

Close
Copy

When a virus strikes, then the knowledge that something is wrong will let you break the attack and recover the system to operational status the quickest possible.

Since viruses are real programs and they have to execute their code somehow, then they necessarily leave traces. Virus capturing, the equivalent of detection in generic terminology, is based on the sensing of phenomena indicating the possible presence of a virus. This is different than finding a specific signature in memory or in a file which is the method used in scanners and TSR. The range of useful phenomena for virus capturing is broader and includes self-baiting and self integrity checking, verification of memory stealing, launching bait sequences, sensing Piggybacking or file killing, integrity checking (yes, this too), and the sensing of active deception. It's also possible to use tunneling for capturing boot viruses, either by tracing the origin of certain interrupts, or with hardware access.

Generic methods do not require a priori knowledge of specific viruses, as is true with signature scanners. All viruses will disclose their presence to one or more of the sensing methods mentioned. This has important implications. Virus capturing methods can detect the activity of both existing and new viruses which isn't possible using known virus signatures.

The Generic Integrity Analyzer

Close
Copy

Integrity analyzing deserve special attention because it's the most powerful of the generic methods. A properly designed analyzer can tell you more about a virus, in a matter of moments, than any other tool. When properly used the integrity analyzer can determine the size of the attacking virus, whether it uses stealth, if it's full or only semi-stealth (there is a difference in recovering from each), and if the file is recoverable.

To understand how integrity analyzers work it is instructive to compare them to integrity checkers. The latter calculates a number (CRC) that represents every byte in the processed file. Changing a single byte anywhere in the file will result in a different calculated CRC.

For example, DOS's SETVER.EXE program contains the version table of programs written to a special section in the SETVER.EXE file. Entries can be added or removed from the SETVER table. Changes to the SETVER table are legitimate as it is the purpose of the program, and adding an entry or deleting an entry does not indicate infection. An integrity checker will detect changes to the SETVER table without distinguishing between changes due to a virus or a legitimate one. An antivirus integrity analyzer can tell the difference.

The programs that interest virus writers are mostly of two types, COM and EXE. Every program has an entry point, this is where DOS starts reading the instructions and executing the program. The entry point is usually indicated by a 'jump' or 'call' instruction at the beginning of COM files, or by a set of pointers, contained in the header of EXE programs.

When a virus infects a program, it either appends, prepends, or inserts its code into the file and then modifies the entry pointer(s) to start execution from the virus code. Let's assume, for example, that we take a snapshot of a few bytes at the entry point of the uninfected file. If we compare the entry point of an original and infected file, and the bytes at the same relative offset to the entry point, we'll see that they have changed. Normally we will find three kind of changes in the infected file when it is compared to the uninfected one. The header (or the jump address at the beginning of the file) has changed, the code itself at the entry point has changed, and the location of the entry point has changed, as well.

The above demonstrate that just a few significant bytes can reveal more information about virus infection than a complete file's CRC. Moreover, while the latter is incapable of revealing whether the change was caused by a virus, the data used in generic integrity analysis inherently contains this capability. We need, of course, a few more parameters to confirm whether the change is viral, or due to non-viral factors such as corruption, or perhaps installation of a newer version of the program. A possible way to discriminate between the options is to include in a file's integrity database parameters that should NOT change as a result of infection by a virus. The presence of these parameters in the modified file necessarily indicate a virus infection, and their absence implies that the change is not viral.

Let's now consider the potential of InVircible's integrity analyzers, IVB, for determining the specific file alterations and methods of operation used by a particular virus, and how to handle them. Many of the newer viruses use stealth, although not all have full stealth capability. Discriminating between semi-and full stealth depends upon whether the integrity checker can detect changes when the virus is active in memory. If no changes are detected, then the virus is full stealth and cooperative methods can be used to remove it. If changes can be seen when the virus is active in memory, but no change can be seen in file size, then the virus is semi--stealth. Some semi--stealth viruses will show a **decrease** in file size by exactly the length of the virus code. This is the result of an unsuccessful attempt to conceal their presence.

The conclusion, which is perhaps surprising to some, is that CRC methods are not the best suited for antivirus integrity checking. A generic integrity analyzer is better adapted for this purpose. It can identify changes in program files specific to virus presence and also potentially critical virus characteristics.

Generic Recovery from Virus Attacks

Close
Copy

Generic recovery is the restoration of infected programs to their original, pre-infected state. Some security experts recommend a system should be recovered by deleting the infected programs and replacing them with clean ones from backup. The main reason these experts recommend replacement instead of restoration, is because they claim that you can't be sure the restoration results in a byte for byte identity with the original program. The fact is, however, that this can be easily verified. In the highly technological world in which we live, there is no room for superstition or speculation, certainly not if the facts can be verified. For a privately owned PC, or for a critical application PC, replacement might be the simplest course of action. But in business network environments, where time is of prime importance, fast recovery of executable programs may be imperative. Critical data files can always be restored from backup if there is a concern for their integrity. In fact, rapid program restoration can speed the process of restoring critical data files from backup by returning a system to operational status more quickly.

Computer viruses are deterministic code and they function in a deterministic way, too. Virus names like *Satan Bug* and *Creeping Death* are just folklore and have nothing to do with unnatural powers. The exactness of the recovery of a program from a virus can be verified easily by comparing, byte for byte, a restored program to a clean one from backup. In case of a massive infection, generically restore a few infected samples and compare them to clean originals to determine whether complete restoration is possible. If it is, then you can be sure that the restoration of the rest will be complete, as well. This results from the deterministic nature of a specific virus's method of infection and the inherent logical structure of executable files.

An advantage of IVB is that file integrity authentication and file restoration can be accomplished using the same database files. This capability results from the generic nature of the processes involved. It also further demonstrates the value of generic integrity analysis over checksummers. The databases of the latter do not contain the information required to restore infected programs. A checksum (or CRC) is just a 16, 32 or 64 bit number. How can you restore a file with the knowledge that its pre-infected checksum was 1234 and when infected it is 4321? By contrast, when critical program file characteristics have been sampled and stored in databases, it is possible to use this information to restore files to their original condition, byte for byte.

Cooperative Recovery Methods

Close
Copy

A special category of generic recovery methods are the cooperative ones. These apply only to full stealth viruses of both the boot and file infector types.

The principle involved is extremely simple. The recovery process takes advantage of the fact that a full stealth virus, either boot or file, will present the correct, uninfected data of the inspected sector or file, when the virus is active in memory.

To recover from a stealthed boot infector (MBR infectors are referred to as "boot infectors," as well), simply copy the stealthed image of the infected sector and rewrite it to the same place using tunneling techniques. The advantage of cooperative recovery, over the undocumented 'generic' technique known as FDISK/MBR, is that with the cooperative one you write **exactly** what was there in the master boot sector in the first place, while in many cases you might cause more harm than good with FDISK. InVircible implement this cooperative recovery method under the name 'SeeThru' technology.

Another effective file recovery technique is by **cooperative integrity restoration**. A new integrity database is established while the virus is still active in memory. Then the computer is rebooted and the programs are restored from the database made when the virus was resident. This technique is effective only against full stealth viruses. It has been implemented successfully against *Tremor*, a common virus in Germany, and works against other full stealth viruses such as *NATAS*, *Die_Hard*, *Hemlock*, *Nightfall*, *Invisible Man*, *Uruguay*, and all strains of *Frodo*, as well.

Virus Analyzers

Close
Copy

The problem that haunts security experts when they face a new, or modified virus, is that it usually takes days, sometimes weeks or even months, until antivirus developers have an algorithm available to restore systems from the new virus. We have seen instances when whole enterprises were halted for days because of attacks from new viruses.

Generic methods have a lot to offer in such situations. First, generic recovery will allow return of systems to operational status in the shortest possible time. In most cases, programs can be completely restored using the integrity database without waiting for the disassembly and analysis of the virus. In the case of destructive, overwriting viruses, the integrity database can be used to identify which files need to be removed and replaced.

The correlation scanner, a breakthrough in generic AV technology, can be used to spot infected files that were not found by the integrity analyzer and the source of the infection, too. Since an attacking virus can be of an unknown type, then no scanner will find it. The file that brought the infection into the system will not have a pre-infected, "clean" record in the database since it was already infected when the database for it was created. The correlation scanner will find the original infector, and other infected files, by similarity to an infected sample identified by an integrity analyzer, captured through a baiting process, or designated by the user. IVX, InVircible's correlation scanner was recently enhanced so that a library of signatures can now be used with it, as well.

The generic correlator identifies similarities in the processing that a file undergoes during infection (a virus infection is a 'process'), in the cryptographic model used in the virus code, and the matching of a signature string. The correlation scanner has proven itself effective with plain, encrypted, and even polymorphic viruses.

The correlator may replace the scanner in many cases, and can be used to disinfect a computer, without needing updates, and with no delays. It is a tool that empowers users to restore their systems independently of antivirus security experts who may not always have the resources to respond quickly to user needs and requests.

Integrated Antivirus Protection

Close
Copy

Having reviewed the basics of generic antivirus technology, let's now discuss an overall antivirus strategy. An effective antivirus defense should consist of several elements. Because of its broad effective spectrum, generic protection has a pivotal role in an integrated AV defense strategy. It can capture a virus that passes through a known virus screening process and act as a buffer before loading an activity blocker or TSR scanner, if one is used.

On-demand and TSR scanners can detect only viruses that are contained in their signature database. Often, this does not include all known viruses. Therefore, virus scanning should always be **preceded** by running a generic probe and integrity analyzer, especially before scanning a file server. There is always the risk that a new fast infector, or even a known one not included or accurately detected, is active in memory. Signature scanning, without first performing a generic integrity analysis, can infect a large number of executables inspected by the scanner.

The backbone of an integrated antivirus strategy is the integrity analyzer. First, you need to run it regularly in order to keep its database up to date. Secondly, it's arguably the most effective means to detect an infection in its earliest stage. Thirdly, it will be the first to capture the presence of a fast infector before it can infect any significant number of programs, even if the virus is new. And lastly, it provides a very effective way to quickly and fully recover a system to its original, pre-infected condition, in the event of an infection.

Virus scanners are a convenience, for offline scanning new software before installation, and for scanning floppies.

The core component of a comprehensive AV defense can be modeled upon the military equivalent of a "fail safe." A "fail safe" is a counter defensive process that will operate despite unknown and variable conditions and methods of attack.

Generic methodologies best implement this concept as they function without regard to the specific details prevailing at the time of the viral attack, and use multiple, overlapping and partially redundant mechanisms for virus capture and, if needed, system recovery. Deliberate redundancy means that one or more of the generic methods will operate effectively despite variability in virus hosts or methods of viral action. In contrast, signature based antivirus can only handle known threats, contained in their database.

Therefore, a more defensible AV strategy combines generic capture and restoration methods with known virus scanners. A cost effective benefit of this approach is that scanners will not need to be updated as frequently since the main purpose of updates is to detect new viruses. The generic methods acts as both a buffer and safety net to protect systems until preventative methods using scanners are possible. Since it can take anywhere from days to months for signature detection and cleaning algorithms to be made available, generic methods provide a highly significant "fail safe" to overall AV

defense strategies. Even more importantly, advanced generic methods such as those presented, here, will provide both virus capture and system restoration immediately.

The use of antivirus TSR and activity blockers is debatable. They always adversely and, at times, significantly impact system performance and available resources. They can also interfere with normal program operation, and conflict with other applications. Yet there are users that won't give them up. Sometimes this decision is based on the mistaken assumption that TSR provide protection beyond what is available from scanners. In fact, the latter are invariably more comprehensive and reliable than their respective TSR, from the same vendor.

From a practical standpoint, when generic, TSR scanner, and activity blocking methods are used the latter two should be run after completing the generic tests and integrity analyses. Generics and AV TSR do not coexist well. The latter intercept generic probes as if they were viruses. This is especially true of activity blockers.

InVircible Installation

Close
Copy

Installation to Hard-disk

Insert the InVircible diskette in drive A or B. Run **drive:INSTALL** to begin the installation, where **drive** is A or B.

The INSTALL program will prompt for the directory to install to. Type the full pathname of the preferred location or just press Enter for the C:\IV default, when prompted.

INSTALL has the following additional options:

- **The Rescue-Disk sub-menu**, for the preparation of the diskette.
- **The License Installation/Removal sub-menu.** Enables either the installation or removal of the InVircible license to/from the hard drive. This option is available only if you possess the original distribution diskette. The process is reversible.
- **The Batches sub-menu.** Enables the insertion/removal of the IVTEST command to/from batch files in specified directories. The installation of the IVTEST probe in batch files is optional. IVTEST will be installed in its "quiet" mode. IVTEST messages will be displayed only if suspect activity is detected.
- **On-line Registration.** Enables the on-line licensing of InVircible by telephone. Accessible by pressing **F10**. Call your IV dealer for details.

Installation To or From a Network Server

InVircible is a universal anti-virus, recommended for both single user PC as well as for all popular network brands.

INSTALL enables the installation of InVircible to a file server or to a workstation in a network. The default is installation to a single user PC's hard disk.

Select where to install InVircible (singe user PC - the default -, to a file server, or to a workstation in a network) from the Install sub-menu. Note that the installation (or removal) of the license to the hard disk can be done from the original diskette only.

All stations connected to the network and equipped with a hard-drive need individual licensing. If you prefer working in the Sentry mode then do not install the license record to workstations. The file server does not need the installation of the registration key. Any workstation in the network having a hard disk may be infected prior to logging-in to the network. Therefore, any such station needs its own InVircible installation.

In Non-DOS Environment

The InVircible INSTALL procedure should preferably be run from a plain DOS environment. If INSTALL is attempted from an unsuitable environment then a warning message will request the user to boot from DOS and try again.

In a Corporate Environment (Sentry Mode)

There are instances when only the detection and alerting functions of InVircible's are needed. To install InVircible in the sentry mode (detection only), just skip the key transfer step in the installation procedure by running **INSTALL /FAST** or by leaving the write protect tab of the InVircible diskette in the protected position. The retraction of the license from the hard disk back to the diskette will also switch InVircible to the sentry mode.

The sentry mode is indicated in both IV's Caution Panel and in the upper left corner of the program's panel.

Unattended Installations

There are instances where a PC should operate uninterrupted in an unattended or remotely controlled installation, like in bulletin board systems (BBS). The daily inspection of IVB, when automatically rebooting may pause if IVB finds changes in programs. To bypass the pause use the **/NOSTOP** switch on IVB's command line in the **AUTOEXEC**, as follows: **IVB C: DAILY /NOSTOP**

IV Licensing Utility for System Administrators

System administrators can authorize a remote user, online, from a site license master floppy. The special **LICENSE** utility is available only to purchasers of a site license, from your authorized InVircible agent.

The Rescue Diskette

Close
Copy

A Rescue Diskette is essential to disaster recovery and effective virus protection. The IV **ResQdiskette** is one of IV's strongest features.

The Rescue Diskette is a bootable floppy, containing the necessary InVircible programs, the hard-disk's boot-areas image files, the data stored in the CMOS, a copy of your AUTOEXEC.BAT and CONFIG.SYS and some useful DOS files such as **FDISK, SYS**, and **UNFORMAT**. It enables to recover the boot and system areas of the hard-disk, its track zero, the CMOS settings, as well as the recovery of files in case of infection.

Preparation of the Rescue Diskette

Install InVircible normally to the hard disk and reboot the computer to make the installation effective. Insert a formatted (**FORMAT A: /U**) diskette into drive A and run **INSTALL /R**. First, the system and InVircible files will be transferred to the diskette, to make it bootable. Then, the boot areas of your hard-drive will be backed-up to the diskette. Write-protect the diskette.

Partitioning device drivers (e.g. Disk Manager v6.03+), disk-compression (*Stacker*, *SuperStor*, *DoubleSpace*), password drivers etc., required to recognize partitions or drives at booting time, should be installed onto the Rescue Diskette.

There is no point in preparing a Rescue Diskette for a computer without a hard disk. The Rescue Diskette is individual to the specific hard-disk for which it was prepared. Rescue Diskettes should never be swapped between computers.

Customizing Your ResQdiskette

You may add utilities such as your favorite ASCII editor, special drivers and configuration files to your ResQdiskette. IV will take care of Stacker and DoubleSpace disk compression. Other drivers should be added manually by the user.

Format a floppy in drive A, copy the drivers and utilities you need to A:, and edit **a:\config.sys** and **a:\autoexec.bat**, if needed. IV's rescue disk procedure will edit it's commands into your startup files without erasing them. Start **INSTALL /R** and answer "No" when the program prompts if to erase the existing data.

Making a Disk Manager Rescue Floppy

To prepare a boot rescue floppy for a drive with Disk Manager's ***Dynamic Drive Overlay*** you should use **Ontrack's DMCFIG** utility, provided on the Disk Manager utilities floppy. Run **DMCFIG /D=A**. Alternatively, you may copy DMDRVR.BIN and XBIOS.OVL from the Ontrack floppy to the rescue floppy and edit an **a:\config.sys** on the rescue floppy with the statement: **DEVICE = DMDRVR.BIN**.

The DMCFIG utility will do it automatically. Now proceed with **INSTALL /R** as explained

above.

When Should You Renew The ResQdiskette?

The Rescue Diskette contains the PC's 'personal' data: a copy of the installed operating system, partition and boot sector images, a copy of track zero and the CMOS setup parameters. Whenever one of these is changed, such as when changing a DOS version, or modifying the disk partition, then renew the ResQdiskette. It is recommended that you renew the ResQdiskette when upgrading InVircible.

There is no need to renew the rescue diskette when new directories or programs are added on the hard disk. This is taken care of with the DAILY inspection of IVB.

WARNING: The Rescue Diskette is unique for a specific hard disk. Rescue Diskettes should never be swapped between computers.

Optimizing Your Anti-Virus Strategy

Close
Copy

Effective anti-virus protection starts with the installation of InVircible on a clean computer. If found infected during first time installation, then first disinfect the computer with IV (preferred) or with a third party scanner and install IV when clean.

Once installed, IV detects viral deviations from the clean initial state and let restore programs and boot sectors to their pre-infected state.

In the Single User Environment

Install InVircible to the hard drive, as instructed above. The INSTALL utility will edit the AUTOEXEC file, after prompting for permission, and add the necessary commands for automatic tests to be run on the computer initialization. None of the InVircible programs will load itself and stay resident in memory. Once it completed its function, it will unload and leave the maximum of available memory for applications.

From now on, InVircible will detect and correct most problems right at initialization. These include partition or boot sector tampering and certain operating system changes.

It some cases, InVircible may just alert, without correcting, on the presence of a suspected activity in the computer. Such alert is accompanied by a message describing the generic nature of the problem and the recommended corrective action.

On first booting at any given-date, IVB will scan for integrity through all the hard disk local partitions. On successive initialization on the same date, only the root directory will be scanned. Added programs will be automatically secured on the daily scan. If weekly integrity checking is preferred, change the DAILY argument to WEEKLY, in the AUTOEXEC file.

CD ROM, network and remote drives are not checked by the daily command.

Make frequent data backups. Data should be backed-up at reasonable intervals. Programs should be backed-up only once or kept preferably on the original distribution disks or CDs.

To improve the survivability and recoverability of your programs and data, it's recommended to regularly run a FAT backup program in your AUTOEXEC.

Such utilities are Central Point's **MIRROR** (available from MS-DOS 5) and **IMAGE**, from the Norton Utilities. MIRROR is convenient since its complementary program, **UNFORMAT**, is still contained in later versions of DOS. To restore an Image FAT backup, Norton's UNFORMAT.EXE should be used instead. Both UNFORMAT utilities are backed up on the IV rescue diskette, if they can be found in the DOS search path. UNFORMAT.EXE will be renamed to UNFORMT!.EXE on IV's rescue floppy, for not to confuse it with UNFORMAT.COM. Run **MIRROR** or **IMAGE** from the last line of your AUTOEXEC, before loading Windows, if you do. See elsewhere in this hypertext for

more details on FAT mirroring and recovery.

The InVircible system uses a non-TSR virus-activity probe and sampler, called IVTEST. Anti-viral TSR are limited only to known viruses, and they may be obtrusive at times. Yet, there is need to verify from time to time that a viral process is not spreading in the system. This is achieved by the automatic running of IVTEST, from frequently used BATCH files. The INSTALL utility will add the **IVTEST /Q** command (in quiet mode) into batch files in a specified directory.

Antiviral Protection in Network

Close
Copy

This section describes how to implement InVircible for the protection of networks.

Any workstation equipped with a hard disk, should have its own InVircible installation. The InVircible distribution diskette for network installations has multiple license keys, according to the number of licensed users. The license key should be installed only to workstations that have a local hard drive.

It is recommended that the network administrator run a daily IVB inspection of selected directories of the server. This can be done by adding command lines in the network schedule file or script. IVB returns an errorlevel exit-code. Errorlevel 0 means no findings, and anything else may indicate virus activity. Test for errorlevel 1 in batch or script files for suspect activity alerting.

The Virus Problem in Networks

To this date, there are no known viruses that affect the operation of Novell's Netware software. Yet, DOS viruses may affect files stored on file servers. An infected object, such as a DOS shared application, when executed, can further spread the infection between workstations.

As a general rule, boot and MBR infectors, excluding multipartite ones, will not propagate through the network. The effects of such viruses are retained locally and will not extend outside of the affected workstation. A file server can even be started with its MBR infected by *Stoned* or *Michelangelo*, and none of the workstations will suffer anything or even notice there is something wrong with the server hard disk!

On the other hand, many file viruses can survive the Netware login process, or become active when already logged onto the network, and infect files on either the local disks, or on the file server.

Virus Propagation in a Network

File viruses are usually categorized as direct action viruses and memory resident viruses. Some memory resident viruses are also known as "fast infectors" or piggybackers. All three terms refer to the propagation mechanism of file viruses. Direct action viruses infect additional file(s) every time an infected file is executed. Since the virus is not memory resident then the infection of more files will not occur on the execution of a program that is not actually infected. The criteria and pattern upon which additional files become infected can be anything: i.e. infecting files in the current directory, down the directories tree, along the DOS search path, at random etc.

Memory resident viruses usually load into the workstation's memory, when the first infected file is executed. From then on, any file that will be executed and fits the infection criteria of the particular virus (that is now memory-resident), will become infected itself. Many memory resident viruses have a direct action mode of infection in

addition to the memory resident infection mode, as described above.

Piggybackers are memory resident viruses that will infect any file that is "opened" or "closed" by DOS. The programs that are the most likely to become carriers of fast infectors are virus scanners as they 'open' and 'close' every single program on a drive while searching them for viruses! It is thus essential that antivirus scanning programs, whether scanners or integrity checkers, be equipped with anti Piggybacking features. For the moment, only InVircible has this ability.

Here is how viruses propagate in a network. Suppose a virus from one of the workstations found its way to an application on the file server. If another workstation executes the infected program, then that station's memory - and eventually its hard disk - will become an additional source for the spreading of the virus. Later on, the newly infected station will infect additional applications on the server.

The most common agents that cause massive infections in file servers are unaware network supervisors and their excessive use of virus scanners! The classical scenario is as follows: A virus manages to infect the network administrator's workstation. Most managers unjustifiably rely on their antiviral TSR and scanners, or on NLM. With a false sense of security they overlook and reject the possibility that the scanner itself may be the source of an infection. Sooner or later, the supervisor will log-in with full read-write rights (supervisor or not ?!) and scan for viruses with a piggybacked scanner. The results are obvious.

Here is how to prevent virus propagation through the network:

- **Start by protecting the workstations.** The latter should be checked for virus presence prior to connecting to the network. Especially those PCs that have local hard disk, as their chances to become infected are infinitely higher than those that boot from a network startup floppy, or from ROM-DOS. Antivirus that are initialized from the network, upon login, are looking in the wrong place, and too late.

- **Virus scanning of file servers should be run sparingly and with utmost precautions.** Since Piggybacking cannot be sensed on file servers, then always run first generic probes to assure that the workstation you are working from is clean and the scanner is not being piggybacked by a virus. Scanners should always be tested first on a local hard disk (never from a floppy) to give a piggybacker the opportunity to disclose its own presence.

Protecting a Network

The strategy of how to protect a networked system from virus attacks consists of a few simple steps.

First, have InVircible installed on all stations that have a hard disk. This way, every workstation will be verified before it has a chance to connect to the network and contaminate the server. To facilitate the task, InVircible can be installed directly from the file server, to workstations having a hard disk. IV is provided with the IVLOGIN.EXE program. The supervisor should install IV into a dedicated directory on the server and invoke the **IVLOGIN** program from within the users login script. The following illustrates how to implement automatic IV installation in a Novell network. Suppose IV was installed to F:\PUBLIC\IV. Run SYSCON and select "Supervisor Options" from the

menu, then select "User LOGIN Script". Add the following line into the script: **# F:\PUBLIC\IV\IVLOGIN.EXE**

IVLOGIN will accept the following command arguments: If a random signature filename is desired, then add **/RANDOM** after the IVLOGIN filename. If a specific signature filename is preferred then add **SIG=MY_NAME**. No spaces are allowed in the SIG= argument.

Instead of virus scanning, better is to secure and verify the file server content with IVB, the integrity analyzer and generic restorer. This can be done by issuing the command: **C:\IV\IVB F: DAILY**

It is recommended to run InVircible's **Off-line Backup** periodically, once a week, or twice a month. The Off- line Backup will produce a copy of the directories tree on floppy, with their respective signature files, in case these were lost or tampered with. The off-line backup can be run from the IV shell, using the **Alt-O** key combination.

Refusing Access if Not Checked Daily

IVLOGIN can be used to test whether the logging workstation did run the **IVB DAILY** check and refuse access to the network if it was not checked on the same date. The date information of the latest DAILY check is contained in the file C:\IV.INI. **IVLOGIN /Q** is used to query for the latest daily (or weekly) check. Follows a batch file (name it IVDAILY.BAT) that can be called from the network login script. In the following, the IVLOGIN program is assumed to exist in F:\LOGIN\IV.

```
F:\LOGIN\IV\IVLOGIN.EXE /Q
IF ERRORLEVEL 1 LOGOUT
```

Disinfecting a File Server

Whenever a virus is detected in a network, then act in an orderly manner. Spurious activities tend to cause more harm than good by turning recoverable damage into permanent one.

While the virus is still active, obtain good virus samples on your local disk. Make sure the COMSPEC points to the local disk (type SET to verify where to the COMSPEC variable points). Correct if necessary with **SET COMSPEC = C:\COMMAND.COM**.

Run **C:\IV\IVTEST**, from your local disk. If lucky, IVTEST will generate a couple of samples: VIR-CODE and VIRUSAM.PLE. These files are important for finding other already infected files trough correlation, especially if the infection was caused by a variant of an existing virus or by a new virus.

Proceed first with the disinfection of the local disk(s). Use IVB from the rescue diskette if necessary. After having recovered programs with IVB next try locating the source of the infection with IVX and let it neutralize the affected programs.

When the local disk is clean, login to the network manually (execute the login batch line by line). Avoid the execution of any file from the server itself.

Systems administrators: Do not run LOGIN from the server, as the files in the login directory may be infected. Keep a clean copy of the F:\LOGIN directory on your hard disk, or on an IV Armored floppy and run LOGIN from the floppy or from the hard drive, in order to log in as 'supervisor'.

Now assess the extent of the virus spread on the server by using IVB, IVX or IVSCAN (where applicable). Before cleaning the server, shutdown all workstations in an orderly manner. All stations should be disconnected and shut down, since the virus might be memory resident. All the stations with hard disk should be disinfected individually before connecting them again to the network.

It is possible to reject the login of an infected workstation using the errorlevel returned by IVTEST. The following is an example of a batch file to call from a Netware login script. Note that **errorlevel are returned to DOS**, thus checking errorlevel return should be performed from a batch file, if running under Netware.

```
IF EXIST C:\IV\IVTEST.EXE C:\IV\IVTEST/Q
IF ERRORLEVEL 1 LOGOUT
```

The file server should be restored and cleaned in the following order. First use **IVB /R** for the fastest and best restoration of affected programs (provided they were secured beforehand). Next use **IVSCAN /R**, if the virus is known to IV, to restore those files missed by IVB (no signatures). If the virus is unknown to IVSCAN, then correlate with **IVX** to find out all files missed by IVB. Either rename the suspected files or delete them. Both options exist in IVX. Print the IVB/IVSCAN/IVX reports after each step. You may need to delete a few files that IVB could not restore, by using **IVB /D**.

Replace the files that were renamed or deleted by IVB/IVSCAN/IVX (as in the reports' printout). Restart the network and resume operation.

Network Shared Drives, Viruses and Antivirus

Effective antivirus methods can be implemented through network shared drives like with Microsoft's workgroup and Windows NT. Suppose a stealth virus infected drive C on computer \\AA. Suppose the infected computer is connected to computer \\BB and that AA's drive C is shared with BB and defined as D. Drive \\AA\C is the same as \\BB\D (a double back-slash precedes a computer name in network notation. \\AA\C means drive C on computer AA).

Let's suppose an infected program is executed on \\AA, a virus thus becomes resident in the memory of \\AA. As a result, files may now become infected if executed from \\AA while the virus is active in \\AA's memory. To push the example a bit further, let's also suppose that the affecting virus is a stealth one, so that it cannot be "seen" by an antivirus program when inspecting while active. This is only true if the inspection is run from \\AA.

Yet the infected files on \\AA\C can be verified and even cleaned by running the antivirus program from \\BB, on drive D, as the latter is actually drive C of \\AA. Since no virus is running in \\BB's memory, then the environment is clean and the active virus on \\AA will not interference with the inspection and cleaning, provided it's run from \\BB.

Sharing drives can help in inspecting and cleaning viruses across a network.

The Windows NT Environment

Close
Copy

Susceptibility to Viruses

This section deals mainly with the Windows NT environment. Still, most of the topics discussed under apply to OS/2 as well.

There exist no Windows NT specific viruses. Since writing one is extremely difficult, then it's also unlikely that there will be any in the foreseeable future. Still, Windows NT is susceptible to DOS file infectors when in DOS compatible mode, like when in a Command Prompt session.

Boot Infectors and Multipartites

Whether a hard drive can be affected by a boot or partition infector depends uniquely on whether the boot process is controlled by an IBM-compatible BIOS, hence whether interrupt 13h is supported during the boot process. The installed operating system, or the hard drive's file system (FAT16, FAT32, NTFS, HPFS etc.) are irrelevant to the possibility of the hard drive's MBR or boot sector to become infected during the boot process.

Yet things get more complicated when the 32 bit OS (NT or OS/2) is started and running. For compatibility reasons, the newer operating systems still support the older BIOS interrupts, yet they do it in an indirect way. For example, Windows NT virtualizes interrupt 13h, but only for standard read-write services of the floppy drives. Non-standard formats, like those used in copy protection schemes are not supported under NT, not even for floppies. Hard drive direct sector access is not supported at all under NT. On the other hand, OS/2 will allow some direct disk access, but only when in full screen DOS mode. Yet the latter cannot be relied upon.

Consider what happens in case the hard drive caught a boot infection. As the computer starts booting, the virus will first load into memory. Then, depending on the specific virus and other system configuration parameters, the boot process may either hang or continue. Boot infectors require that BIOS interrupts are available in order to replicate. This would be the case under DOS. Yet boot infectors will have no effect and won't replicate once NT or OS/2 managed to start, even if booted of an infected MBR or boot sector.

Multipartite infectors and droppers are either ineffective or only partially effective under NT. An infected program, or dropper will disclose itself by violating the system's integrity when attempting to directly access the hard drive sectors. Yet the memory resident portion of the virus may now become memory resident and infect other files.

File Infectors Under NT

File infectors behave quite unpredictably under Windows NT. Some might hang, others may infects like under DOS, and still others may just corrupt files instead of infecting them in result of an 'unsuccessful' attempt to infect.

Because of the irregular behavior of viruses, integrity monitoring is the most suitable method for controlling them under Windows NT. Generic restoration from integrity database is optimal under these conditions, and in many cases generic restoration will also be the only working method.

Memory resident viruses can be passed between Command Prompt sessions. Many viruses seek to infect the command interpreter, if not yet infected. The command interpreter is the file designated by the COMSPEC environment string, shown by the **SET** command. The virus will be removed from memory when closing the shell but will reappear every time a Command Prompt shell is opened.

Protecting an NT Workstation or Server

The major virus threat to NT workstations and servers are boot sector infectors. As explained above, boot infectors may prevent Windows NT from starting. The remedy is the InVircible rescue diskette. It will let to remove a boot virus from the hard disk and restore the system to operational status.

To prepare a rescue diskette for an all NT hard drive proceed as follows. Format a DOS bootable floppy on a DOS machine and copy RESQDISK.EXE to the floppy. Boot the NT machine from the floppy and run the following from the A: drive:

```
RESQDISK /B
RESQDISK /Z /B
```

Images of the MBR, the boot sector, the CMOS data and a copy of track zero will be created on the rescue floppy. After having prepared the rescue diskette, change the boot sequence in the setup from A,C to C,A to prevent accidental infection in case an infected floppy was forgotten in drive A. Since NT cannot be accessed from external boot then there is no point in letting A being the boot drive.

Where dual booting is available, then boot from the hard drive to DOS and prepare the rescue diskette the normal way, by running **INSTALL /R**.

Install InVircible to the hard drive using **INSTALL**. If preferred, you can install IV centrally from the server to all workstations by calling **IVLOGIN** in the NT login script. After the installation was completed, import **IV-DAILY.PIF** to the NT Startup group. The latter's default command line is: **IVB C:\ DAILY AUDIT /NOSTOP**. The command line options can be modified with the Windows' PIF Editor.

The NT scheduler can be used for running various IV checks at a predetermined hour and date. The following shows how to run a daily IVB audit check using the NT scheduler. Note that full pathnames are required in both the AT schedule commands as well as in the batch file that is executed by the scheduler.

In C:\IV, create a batch file and name it DAILY.BAT. Edit the the following line in the DAILY batch:

```
C:\IV\IVB.EXE C:\ AUDIT /NOSTOP
```

Next, shell to the Command Prompt and create a new scheduler task by entering the following command:

AT 12:00 /interactive /every: C:\IV\DAILY.BAT

The NT scheduler will now run the DAILY batch, at noon time, on every date.

Disinfecting an NT Workstation or Server

The disinfection of an NT workstation or server from a boot virus can be performed only under DOS. You may need to change the boot sequence from C,A to A,C in order to be able to boot from the IV rescue. The boot chain (the CMOS parameters, and the partition and boot sectors) can be restored from backup on the IV rescue. In case there exist no IV backup, then the MBR can be reconstructed using ResQpro. Yet, it is not recommended that inexperienced users undertake the reconstruction of the MBR for other than DOS partitions.

The restoration of infected files by InVircible's is straightforward. Use either of the following procedures:

- Preferably, disinfect from a remote user, having full share privileges on the infected drive. Use **IVB /R** to restore the affected files to their pre-infected state. To disinfect, the remote user needs to have IV 'Qualified' permission.

- Where remote access is not available, then you can disinfect a workstation or NT server locally by running InVircible from a write protected floppy. Yet, since the command interpreter could be infected then you need first to overwrite the infected command interpreter with a clean one. It is good practice to have a backup copy of the NT command interpreter on a floppy. Serach for COMMAND.COM and CMD.EXE in your C:\WINNTxx\SYSTEM32 directory. Have a back up of both on a floppy, before they become infected.

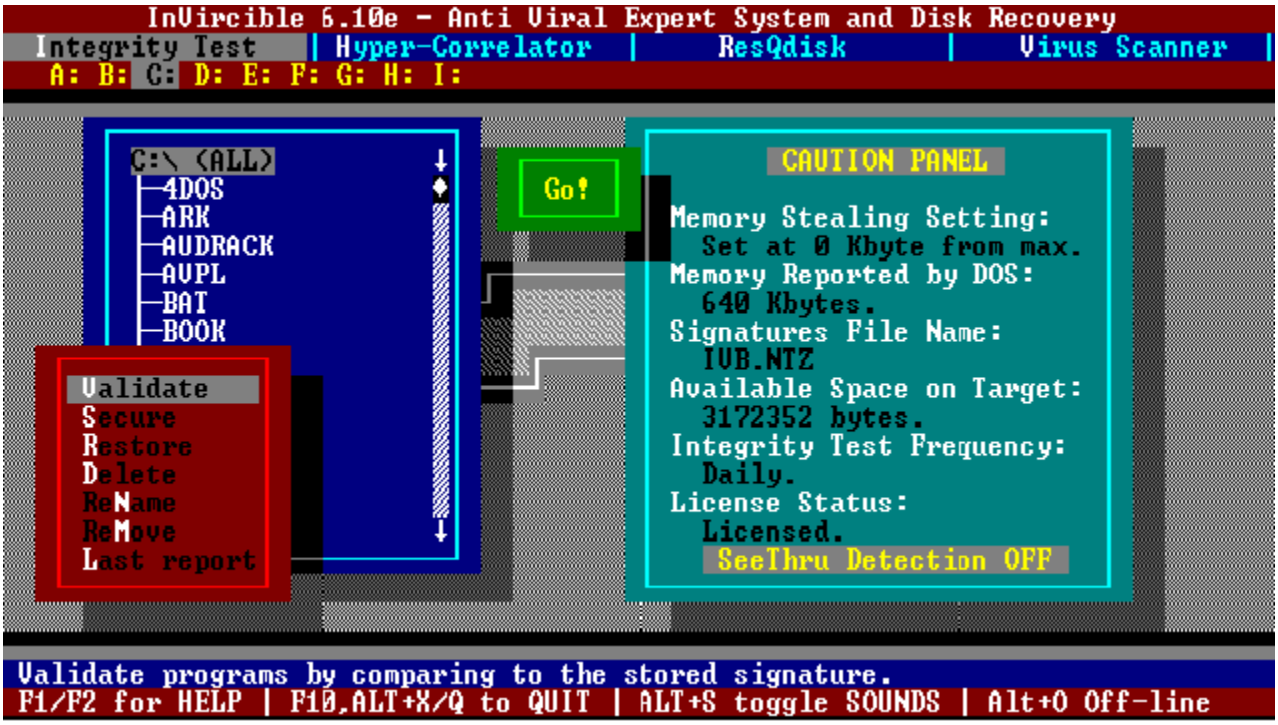
  Use Windows' File Manager to overwrite the command interpreter from backup. Since File Manager doesn't use the command interpreter then there is no risk to reinfect the environment by doing so. Now you may safely shell to the Command Prompt and restore the files by running **A:IVB /R**. The IV floppy or the hard drive need to have IV 'qualified' permission to enable the IVB restore option.

Functionality of InVircible Under NT

ResQdisk will not run under Windows NT as none of its functions is supported undr NT. ResQdisk may be invoked from the NT command prompt, yet the program will simply abort. The other IV modules function normally under NT, less their hard drive boot inspection routines. The latter are inhibited when running under NT.

IV - InVircible User Interface

Close
Copy



InVircible's User Interface

IV may be started in different modes by using the command line options. By default IV will start in the Integrity Check mode, on the current drive. IV's command line syntax is:
IV [/switch] [drive:]

Switches:

- **/V** to start in the Virus Scan mode,
- **/R** to start the ResQdisk boot block maintenance utility.

The "drive" option will home IV to a different than the current drive at startup. The switches and the drive option may be used in combination in any order.

There are several ways to change options and modes with IV, just use the keyboard or the mouse anyway you like. Any "sensible" action will respond accordingly.

IV Special Keys

Alt+key combinations: the Alt key combined with either I, H, R or V will switch to the Integrity Check, Hyper- Correlator, ResQdisk or Virus Scanner respectively.

Ctrl+key combination: the Ctrl key, when combined with a character from **A** to **Z** will

change the current drive to the one selected, if available.

Alt+G shortcut control: the shortcut hot-key is labeled "Go" when a mouse is found or "Alt+G" when the mouse is inactive. The shortcut control may be used to proceed with the selected mode without further selections. Press **Alt+G** or click the mouse on the green "traffic light" at the top center of IV's display.

The IV Caution Panel

The CAUTION PANEL is at the right-hand side of IV's main screen. It displays various parameters on InVircible settings as well as selected warnings, related to possible virus problems.

The first two entries are dedicated to memory stealing alert. In case DOS reports less than the expected memory, a message indicating the number of missing kilobytes will pop up. The message will prompt for confirmation to set the alert threshold to the difference detected. Once set, IV and the InVircible programs will ignore missing memory, if equal to the preset threshold. The threshold is reset automatically to zero when IV detects more DOS memory than the current threshold setting.

A Low memory message will be displayed in case missing memory is detected and the user didn't confirm the setting of the threshold. There are instances when less than 640 kbyte of DOS base memory is normal, such as with certain settings of the setup or in certain compatibles, especially when booted from a diskette (Acer, some Dell models, HP etc.).

The third entry is the current database file name. This filename may be changed through IV by selecting Rename from the Integrity Check menu. IV will prompt for a name, or ask to press Enter to reset to the IVB.NTZ default filename. See also in the IVB extra security topic.

The fourth entry indicates the available space in bytes on the target drive. In case the available space on disk decreases, a Disk space lost! message will come up, indicating the number of bytes "lost". In case lost space is reported, one should be aware of the possibility of virus Piggybacking. IVB or IVSCAN will stop the scanning process in case piggybacking is detected.

The fifth entry indicates the frequency of the full disk Integrity Check. If InVircible was properly installed, it should be either DAILY or WEEKLY. If not, re-INSTALL InVircible as detailed elsewhere.

The last entry checks for the Sentry-licensed mode status and will indicate "licensed" or "detection only". The empty field at the bottom is reserved for warning messages.

Browsing Through Last Reports with IV

The last report generated by either IVB, IVSCAN or IVX can be viewed on screen through IV. Select the Last Report in the Integrity Check or Virus Scan mode. The IVX report can be viewed in either modes by pressing the H key and then Enter. The report files are written to the target's root directory and named IVSCAN.RPT, IVB.RPT or IVX.RPT. The report file can be redirected to the printer through DOS command, either

PRINT filename or **TYPE filename > PRN**.

IVB - The Integrity Expert System

Close
Copy

IVB will secure, authenticate and restore executable programs from virus infection and virus-like modifications.

IVB takes a snapshot (signature) of critical information from each executable file and stores the information in each directory in a database file. The process is called **securing**. The information is then used during **authentication** (also called **validation**) to determine whether a file was modified by a virus. The database is also used to **restore** files that have been modified by viruses.

Viruses are characterized by unique properties: They replicate into other programs, they modify the host program and they normally affect more than one program. All these anomalies are unmistakably detected by IVB.

Common viruses will usually increase the size of a file. If IVB detects that more than one file is "modified" or "changed in size", then virus activity is indicated.

A single modified file, or an inconsistent change pattern, may be a new version of an existing program. IVB will usually detect a version change and suggest to resecure the directory.

IVB, when licensed, restores programs that have been secured. The IVB command-line syntax is:

IVB [-] [drive:\directory] [/option]

The default drive and directory are the current ones. The various options are:

none	to **authenticate** (the default),
/S	to **secure**,
/R	to **restore**,
/D	to **delete** and
/V	to **remove** the signatures database files.

Let IV guide you through the options.

The **delete** option will erase files that cannot be restored, such as overwritten or ruined by a Trojan or virus.

The [**--**] switch (space followed by a hyphen) indicates to IVB to operate on the specified directory only without including the sub-directories.

IVB will automatically secure added files when run in the default mode. The **/S** (**secure**) mode is to be used in case IVB indicates a modified file, which is a replacement rather than a modified one. Use the single directory switch to resecure a single directory by

selecting it through the IV shell.

IVB is tamper-resistant and will detect and replace a tampered record of a particular file, rather than void the database. When renewing a database file, the current signatures file is backed up with the extension name '.000' and the backup is given a read-only attribute. This should let to recover from an erroneous replacement of the database file, should this happen.

IVB processes executable files only. Files with the following extensions are processed automatically by IVB: **BIN, COM, EXE, OV?, SYS, NLM, VLM, 386** and **VXD**. IVB has provisions for adding up to five (5) file types to its processing list. See in the IVB.INI file topic how to do this.

IVB will prompt with the following options before attempting the recovery of a file: **Restore** this single file, **Skip** or **All** - attempt to restore all recoverable files that were changed.

IVB Extra-Security Feature

IVB has extra security features which allow the user to define the name of the integrity database files. The default name of the database files is IVB.NTZ. This feature will prevent the destruction or modification of the IVB database files by a dedicated virus. It allows the user to define a unique filename, other than the default. To set a different name to the database files use the **Rename** option from the IV shell. The database filename in use is indicated in IV's Caution Panel. To force IVB using a specific dabase filename from the command line use the syntax:

IVB [directory and mode] /X forced_filename

The /X switch overrides the database filename currently used.

Off-line Backup of IVB's Database

There are instances when an off-line backup of the database- files tree is desirable. Making a backup is easy using the Off-line Backup option. Use IV to produce an exact replica of the database file's tree on a previously formatted diskette in either floppy drive A: or B:. If necessary, the database files may be restored to their original locations by the **Restore** option of the Off-line Backup mode from IV. The off-line backup function can be entered from the IV shell by pressing **Alt+O**.

The IVB.INI file

IVB.INI is a text file that contains parameters to initialize the IVB program. The file should reside in the same directory as IVB.EXE. IVB.INI can be edited with an ASCII editor. The following are the various entries allowed in the INI file:

Signature file name: To set a signature filename other than the default, write a line as follows: **SIG=filename**. Only one such item is allowed in IVB.INI.

Including new file specs in IVB's Checklist

To include a specific file type in IVB's checklist, add a line as follows: **INCL=*.XYZ.** DOS wildcards (*,?) are accepted in this entry. Up to ten include specs are allowed in IVB.INI.

Excluding files from IVB's Checklist

There are instances where a secured file may change for non-viral reasons. You may then wish to exclude that file from IVB's list, without giving up the processing of other files with the same extension name. An example for such file is RANDSEED.BIN, a random seed sequence, used by the well known PGP program (Pretty Good Privacy), by Phil Zimmerman. See next how to do this.

To exclude a file from being processed by IVB, add a line containing the following: **SKIP=filename**. Do not specify a full pathname, just the filename (up to 12 characters). Wildcards are not allowed in this entry and you may specify up to ten (10) filenames that IVB should ignore.

IVB Audit Mode

Close
Copy

IVB provides for the auditing of specified directories and drives. The audit function is based on the IVB integrity database and runs concurrently with normal IVB integrity checking. The audit feature is useful in monitoring for changes in software inventory. The following events are reported to the audit log:

New files. IVB automatically updates the integrity signature record when new files are added to an existing directory or creates a new record if one does not exist yet. Files that do not have a matching record in the database are reported as new in the audit log.

Changed files. Modified files are reported in both IVB's integrity and audit logs.

Missing files. A "missing file" is a file that exists in IVB's database but is not found in the directory.

Starting the Audit Mode

The audit function is switched on by either of the following:

On demand auditing can be started by adding **AUDIT** to the IVB command line. If an audit log for the specified drive already exists then new entries will be appended to the existing log. The **AUDIT** command is used to **initialize** an audit log for a specific drive. If no audit log exists then a new log file will be created. For example, to initialize an audit log for drive C: run just once **IVB C: AUDIT**.

Automatic auditing (the default). Background auditing starts automatically on condition that the audit log file exists and a "new file" was found.

Auditing is available in registered copies of InVircible only. Auditing runs only in the default mode of IVB (integrity check) and is disabled in all other modes. The extra security switch (**/X**) if effective and may be used in audit mode.

The switching on of the **Audit** flag on screen indicates that and audit entry was logged to file. The audit log will then be presented on screen when the run is completed. In case integrity changes were found, then IVB.RPT will be presented instead. To inhibit the presentation of either report, use the **/NOSTOP** command line switch.

In normal daily use run IVB in the default mode, after having initialized the audit log. IVB will then alert on modified or new files only.

An audit log for a specific drive needs to be initialized just once. Local hard drives and floppy drives will have their audit log written to the root of the specified drive. The audit log of network and CD-ROM drives is written by default to the local C:\ directory (root of C:). The default name of the audit logs is **IV-AUDIT.{D}** where D is the letter of the audited drive. For example, IV-AUDIT.{G} is the audit log of drive G. In case G is a network or CD-ROM drive then the log will be found in C:\. If G is a partition on a local hard drive then the report will be found in G's root.

Audit logs have their attribute set to read-only to protect them from being accidentally erased.

Using the Audit Feature

Auditing is useful in system management and security. On a private user PC, auditing will help you keep track which program was installed and when, or which program is missing since when. In case of an infection, the audit log combined with the IVB integrity report and the IVX report, could possibly point to the particular software that brought the infection. Just check in the audit log for a "new" file or files that were added on the date the infection started, or shortly before.

In the corporate and network environment auditing helps monitoring new software uploads to the server. It is recommended that the network administrator uses a special signatures filename for auditing network drives. The special signature filename may contain high ASCII characters, from ASCII 224 to 255, for example. Filenames containing such characters are hard to handle and are relatively safe from being tampered with by users.

Where higher security is needed, then it's recommended to use IVB's off-line backup (press **Alt+O** when running IV.EXE), in conjunction with the above procedures.

IVX - The Correlation Scanner

Close
Copy

IVX is a generic statistical correlator, based on the comparison of files to a sample known to be infected. This way, IVX detects all the files that were infected by the same virus and can even trace down the source of the infection. IVX is ideal for isolating new viruses, or disinfecting a machine with no previous installation of InVircible on it.

InVircible generates its own samples through IVINIT and IVTEST, either or both Virusample and Vir-Code. A file designated by IVB as changed, may also be used as a valid sample.

IVX runs from either the command line or from the IV shell. The command line syntax is:

IVX [pathname of sample file] [drive:\directory to search]

Wildcards (*.?) are allowed in the sample's pathname.

The IVX report is displayed in a bar-graph format, when the correlator completes its scan. You may also want to review the IVX report through IV.

Suspected files may be renamed through IVX to non executable extension names: COM to IVC, EXE to IVE, BIN to IVB etc. IVX will prompt before actually renaming a file. The renamed files may be safely copied to a floppy and sent or e-mailed for further analysis and evaluation.

The detection threshold of IVX can be adjusted in a range from 1%, for highly polymorphic viruses, to 100%. The default detection threshold is set to 20%.

Tips for Using IVX

The correlator is a powerful tool in the hands of a trained user and its findings are usually conclusive from the very first run. Yet, you should be aware that virus writers make great efforts to conceal their virus' presence. Therefore, do not expect a perfect match, but rarely, when using IVX with real viruses. The results of IVX should be regarded as relative, i.e. the files with the higher correlation are the more likely to be infected. Plain viruses may exhibit anything from 40% to 100% correlation to the sample, while highly polymorphic viruses may have as low as 4%. The conclusive evidence is that non-infected files are in even lower correlation with the sample.

Before launching IVX, spot a few infected files that could be used as samples. The easiest is to look for them in the DOS directory. Frequently used files are usually the first victims of an infection. Boot from a clean DOS floppy and run IVX from a write protected or IV Armored floppy.

When dealing with a virus that infects both COM and EXE files, prefer infected EXE file as the sample, as it contains more relevant information than infected COM. The results will usually be more conclusive than with a COM sample.

Before renaming suspected files through IVX, try a few runs with different detection thresholds. Review the report to chose the best threshold. The best is when all infected files are captured and the report contains no false positives. Yet it's preferable to replace a few extra files in case of doubt, than to risk reinfection.

Examples:

IVX bar-graph for Jerusalem.Standard

The first example is of the *Jerusalem* virus. It infects both COM and EXE files. The sample used in this illustration is Virusample file, created automatically by IVTEST. The report shows 100% correlation of the COM files and 60% for EXE, since *Jerusalem* is a plain non-encrypted virus. If we had used Vir-Code instead, the results would be 100% correlation for EXE and 60% for COM, as Vir-Code is an EXE style sample, while Virusample is usually a COM one. The threshold was set to 50%, to filter out false positives.

IVX bar-graph for Maltese Amoeba

The second example was taken from *Maltese Amoeba* (alias *Irish*) and is especially interesting. *Irish* is heavily encrypted, although it is not considered as polymorphic. DELTREE.EXE is used as the sample. Note the dispersion in the correlation factor. Note also that the largest common correlation is contributed by encryption (the middle shading). As a rule, encrypted viruses will show no contribution from string matching, with most of the similarity due to encryption, and sometimes some contribution due to structure matching. The threshold was set to 30% in this case.

Practicing the Use of IVX

It may be worth to practice with IVX so that you know what to do when the real thing happens. You may use IVX to trace down files that were processed in the same way. For example, all InVircible files, except IVHELP, were processed by LZEXE runtime compression. Try the following:

```
IVX C:\IV\IVB.EXE C:\
```

All the files that correlate higher than 65% are treated by LZEXE as well. Now, if you correlate with IVHELP.EXE as the sample, IVX will find programs that were processed by PKLITE. You will find a few of in the DOS directory.

Another compression scheme that can be used to practice IVX is DIET. Correlate to Frisk Software's F-PROT.EXE (if you got a copy) and set the threshold to 50.

IVX - Advanced Options

IVX, the hyper-correlator of InVircible, can be used in one of the following modes:

- As a ***statistical correlator***, to detect statistical similarities between files.
- As a ***signature extractor***, for establishing a scan signature from an infected sample.
- As a ***scanner***, from a signatures library.
- For ***searching a text string*** in files.
- As a ***signature scanner***, from a user defined signature.

All modes can be entered from the interactive dialogue menu. Only the statistical mode can be entered directly from the command line.

The Statistical Correlator

This mode is started by designating an infected file to compare to. If there exist a ViruSample, Vir-Code or both, then IVX will pick them automatically for sampling, in the "Auto" mode. IVX will analyze the sample file and establish its correlation parameters in three categories: structure, crypto model and signature. IVX extracts a signature from the sample file, where possible. The signature is automatically added to the IVX.LOG file, in the IVX directory.

Extracting Signatures

Finding a good signature needs common sense, patience and experience. Yet, it's not always possible to find a valid signature for every virus. With some viruses, IVX will extract a valid signature on the first run, others may need the user intervention.

The following are valid signatures, extracted automatically by IVX, as stored in IVX.LOG, on the author's hard drive.

```
BB8513E3262CAEA8B892 ;      Big Caibua
8BCC0EFA178BE78EC08D ;      Die Hard
378B0E4B8B053CD5043D ;      Frodo, 100 Years
072E8E1645002E8B2643 ;      Jerusalem.1808, Standard
```

Each line contains a single signature, made of ten bytes (in hex notation), then a ';' mandatory separator, and a description. IVX will write the full pathname of the sample file as description. In the above, it's obvious that the sample are all from the file viruses directory on the author's drive. The last entry in each line (the name of the virus) was added with a text editor. Free text is allowed past the ';' separator, as IVX will ignore anything to its right.

To decide whether a signature is good, simply scan for that signature through several

infected files. If IVX finds the signature in all, then the signature is good. There are instances when a signature extracted from an infected COM sample is not found in EXE files, and vice versa. Sometimes, the signature taken from the EXE sample is good for both, or the one from the COM sample is good for both. There is one way to find out: Trial and error. It's possible that you'll need two signatures for the same virus, one for EXE, another for COM files.

There are instances when 'wildcards' may be needed. Suppose we found an exotic new virus and when correlated for in the statistical mode, no match is found for a signature. Yet, when looking in the log file we may see that the signatures 'almost' match, except for a few bytes that change value in the different signatures. You may then replace these bytes by a '??' wildcard. IVX will ignore the byte at the specific position. Let's look at an example. Suppose we obtained signatures for the following samples, all from the Exotic virus infections:

```
8BFF0EFA178BE78EC08D ;   SAMPLE-1.COM
8BCC0EFA178BE7FFC08D ;   SAMPLE-2.EXE
8BCC0EFA178BE78EC08D ;   SAMPLE-3.COM
8BFF0EFA178BE7FFC08D ;   SAMPLE-5.EXE
```

A good signature for the Exotic virus will look as follows, with bytes 2 and 8 replaced with the ?? wildcard. The same can be used when adopting user defined signatures from other than IVX sources.

```
8B??0EFA178BE7??C08D ;   Exotic virus signature.
```

Using IVX in Signature Mode

User defined signatures can be entered in three different ways:

- As an ***existing IVX signature***, from the ***IVX.LOG*** library, or by adding a new, or editing an existing signature in the library.
- As an ***ASCII string***, from the keyboard. The search for ASCII is case sensitive and looks for a perfect match.
- As a ***hex signature***, from the keyboard. Wildcards are allowed.

Start IVX, select the 'Signature' mode and IVX will prompt for one of the above. Pressing 'A' will start the ASCII entry mode, 'H' will start the hex entry mode, and Enter will open the LOG file for browsing and selection.

The 'Signature' mode can be used in a staggered scan strategy. The probability of an occasional occurrence of the same signature with 10 valid bytes is fairly low. Even six valid bytes still provide a fair margin against false alarms. Therefore, the signature scan mode of IVX can use sometimes for the final scan, to eliminate the 'noise' that may exist in the statistical mode. For that purpose, a good signature should be extracted first through the statistical mode, and validated as explained. Then the final run is executed in 'signature' mode, with 100% match and no false positives.

Using IVX for Virus Alert

A common user with no background in virus or antivirus matters, when hit by a new virus, like Big Caibua (April 1995), can issue a virus alert on the net within minutes. After

he/she confirmed the validity of the signature, of course.

The message should contain the signature extracted by IVX, as found in the IVX.LOG file. Conversely, all users of InVircible (or IVX) can add the signature to their IVX.LOG file and scan for the new virus through their programs.

IVX can also be used as a signature scanner for user defined signatures obtained from other sources. There is no limitation on the number of signatures that can be stored in the IVX.LOG file, except your patience for browsing till you find the right one.

ResQdisk - The Hard Disk Recovery Tool

ResQdisk is a compact yet useful utility capable of repairing damage to the hard disk partition or boot sectors, removing known and unknown partition and boot viruses, restoring self booting capability to the hard disk and regaining access to the disk, in cases where access was lost.

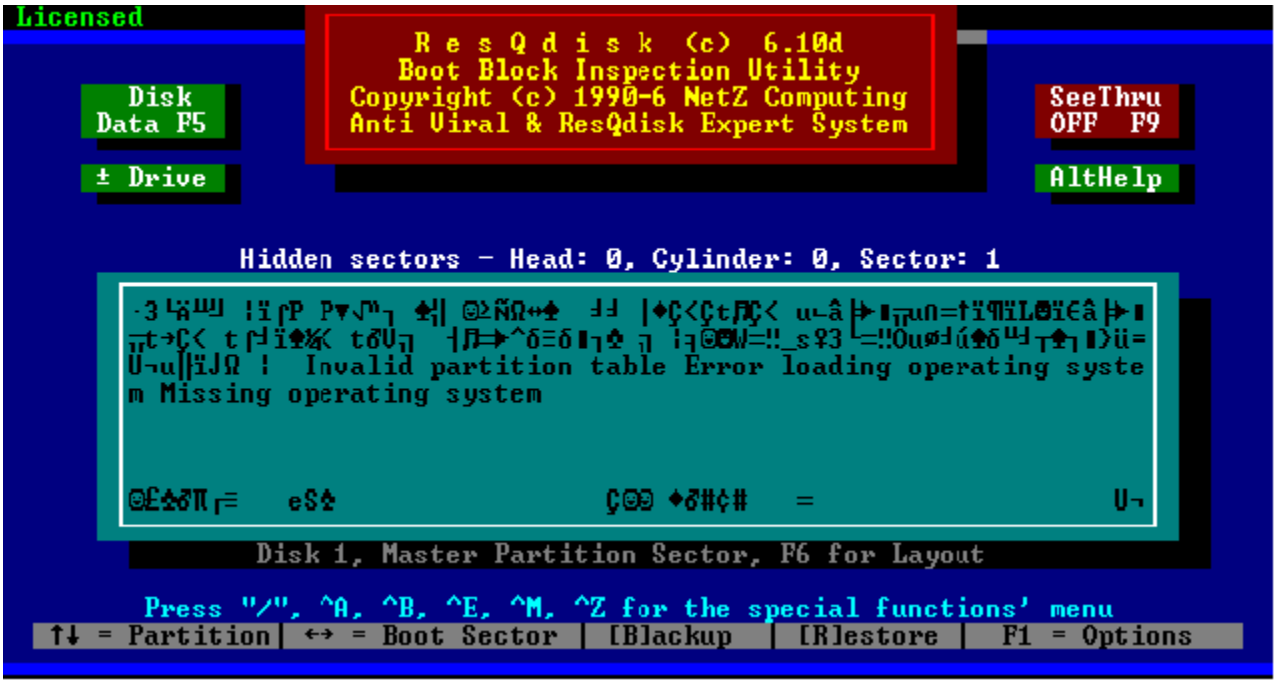
ResQdisk deals uniquely with hard disks drives that were prepared and configured by DOS FDISK or Windows 95.

DOS disks are organized in sectors containing each 512 bytes data storage units. Any DOS hard drive has a master boot record (MBR), sometimes referred to as the partition sector and at least one boot sector. Every partition has its own boot sector. Floppies have only a boot sector. Both partition and boot sectors are usually referred to with the general name 'boot sectors.'

Boot sectors are favorite targets for computer viruses, since they contain an executable programs known as the *bootstrap* that loads at boot time. A damaged or zeroed sector will return an "*invalid drive specification*" message, fail to boot, deny access to the hard drive or simply read trash from the disk, depending on the kind of damage to the sector.

For maximum safety, backup and image files of the boot sectors should be written to a rescue diskette since an inaccessible disk will deny access to backups written to itself. The ResQdisk hotkeys and options are displayed at the bottom of its user interface. The user is encouraged to familiarize him/herself through browsing the hard drive with ResQdisk. Extensive online help is available when in ResQdisk by pressing **Alt+H**, or **Alt+G** for the online manual / guide.

Caution: Hard disks have unique configurations of the boot sectors. Do not attempt to swap or copy boot areas between hard-disks. This might end up with an unreadable contents of the transplant's receptor.



The ResQdisk user interface

Navigating with ResQdisk: The direction keys are used to navigate ResQdisk to read the configuration sectors of your hard disk(s). The content of the sector is displayed in the window, in black on cyan background.

Use the following keys to browse through the sectors:

Home	Go to the master boot record (MBR) at sector 0,0,1.
PgDn	Go to next partition sector, it there exist one.
Left arrow	Go to the active boot sector.
Right arrow	Go to next partition boot sector, if there is one.
Up & Down arrows	Browse trough sectors on track zero, head zero.
+/- (plus/minus) **keys:**	Change between hard drives 1 and 2.

Refreshing the MBR Bootstrap

ResQdisk has embedded routines for the refreshing of the bootstrap program of the MBR or the boot sector, in case they have been damaged or cannot be recovered.

The refreshing of the MBR is the equivalent of running **FDISK /MBR**. The refresh method can be used for removing generic partition infectors. To refresh the MBR press the **Home** key, then **F1**, then **F4**.

The bootstrap refresh procedure should be used with utmost care. Do not use the **Home-F1-F4** sequence in the following cases, or you may loose access to the hard drive:

You cannot access the hard drive when booted from a clean DOS floppy. This

could be the case when the hard drive is infected by a virus that modifies the partition data, or you are using security software that encrypts the MBR, or you are using dynamic boot overlay.

The disk is booting to other than plain DOS or Windows 95. The refresh sequence should not be used if running under Boot Manager or Windows NT.

Refreshing the Active Boot Sector

Boot sector infectors, not to confuse them with partition infectors, can be removed from the hard drive by transferring a fresh copy of the system. Booting clean from a floppy and running **SYS C:** will refresh the bootstrap code of the active partition's boot sector. Viruses that can be removed this way are Form, Boot-437 and Da'Boys, to mention the most prevalent ones. Yet, executing a system transfer is not always practical or possible as you will need a clean copy of the same OS version that is currently installed and running on the hard drive.

ResQdisk will let you refresh the active boot sector without requiring a copy of the installed operating system. In most case you won't even need to boot clean to remove a boot sector infector.

The ResQdisk procedure refreshes the boot sector the same way as DOS's (or Windows 95's) **SYS C:** but without transferring the system's files. ResQdisk will search for existing system files in the root of drive C: and chose the correct boot style from MS-DOS, PC-DOS or Windows 95. Where no system files are found, ResQdisk will use MS-DOS as the default boot style.

Refreshing the Boot Sector Via Interrupt 13h

The sequence for refreshing the active boot sector is: **Left arrow, F1, F4**. This method may be used on hard drives with a standard configuration and geometry. Here is how to test if the INT 13h boot refresh method can be used safely on a particular drive.

Press **Left arrow**, then **^A** and analyze the sector '**as boot sector**'. If the boot sector parameters make sense then the **F1-F4** method can be used on that drive. The INT 13h boot refresh method should not be used on drives with dynamic boot overlay (DDO). ResQdisk has an internal safety lock against such eventuality, still, it can be circumvented if you insist.

INT 13h boot refreshing is especially effective against stealth boot infector as SeeThru is available in the INT 13h mode.

Refreshing the Boot Sector Via DOS

The active boot sector on drives using DDO cannot be accessed with interrupt 13h as it is stealthed and virtualized by the DDO. Instead, it can be accessed with DOS, provided the dynamic overlay was loaded and is running in memory. To let ResQdisk access the active boot sector via DOS press **^B** or **Alt+B**. Select the **logical drive** which's boot sector you want to access. A menu will open. The **SYS** option will refresh the boot sector's bootstrap of the selected drive.

Caution! The boot style (MS-DOS, PC-DOS or Windows 95) is selected upon the system files found in C:\. **Do NOT use this method to refresh the boot sector of compressed (Stacker, DoubleSpace etc.) drives.**

Proceed as follows to test whether the DOS boot refresh method can be used safely on a particular drive. Press **^B,** select the drive and then **Read** to load the sector to the clipboard, next press **^C** (show the clipboard) then **^A** and analyze '**as boot sector**'. If the boot sector parameters make sense then the DOS boot refresh method can be used on that drive.

The DOS boot refresh method is effective for removing non-stealth boot infectors without requiring a clean boot. To this date, the most prevalent boot infectors of hard drives, like *Form, Da'Boys* and *Boot-437* do not use stealth. Therefore this method can be used against them in the majority of cases.

Restoring the Hard Drive's CMOS Parameters

InVircible monitors changes in the setup parameters of the hard drive(s) in the CMOS data. The purpose of monitoring the CMOS content is to let resume operation rapidly in case the setup data was accidentally changed or zeroed. This could happen because of a weak battery, electrical surge, a user setup error and sometimes even bad or corrupted software.

The CMOS data backup file is monitored for changes every time IVINIT runs, as part of its routine tasks. The CMOS data is also backed up to the rescue diskette. To restore the CMOS parameters from backup, use **^F3** or **Alt+F3** from ResQdisk.

The IDE Analyzer and Setup Calculator

ResQdisk has a built-in IDE analyzer and setup calculator. This feature is especially useful with IDE/E-IDE drives, in case the CMOS data was changed or nullified and there is no available data on the drive parameters. Select 'autodetect' (if available) or a random disk type from the setup table, boot from a DOS floppy and run ResQdisk. Use the setup feature by pressing **F5**. It will identify the manufacturer's parameters of the hard disk as well as the IDE model.

For SCSI and MFM drives, ResQdisk's setup panel will show the drive configuration as detected by the BIOS.

Backing up and Restoring Boot-Partition Sectors

Sector can be individually backed up or restored. To backup or restore a sector, go to the sector with the navigation keys and press "**B**" for backup or "**R**" for restore. IV's rescue diskette contains a backup for each critical sector.

Basic ResQdisk Hot-Keys

The following are a list of ResQdisk's hot-keys which give access to further selection through menus:

Analyze functions (^A, Alt+A): The hot-key will open the analyze menu, with options

to assess the sector as a partition or as boot sector. In the **partition mode** the partition table data is displayed. In the **boot mode** ResQdisk will display the BIOS Parameters Block (BPB) of the shown boot sector.

Boot sector functions (^B, Alt+B): There are instances where it is preferred to use DOS interrupts to access the system boot sector, rather than BIOS interrupt 13h. Such is the case when you need handle the boot sector of a Disk Manager partition. The **^B** menu has the same options as **^E**, but uses DOS instead of BIOS. The SYS option has the equivalent effect of a SYS C: command, on the boot sector, but without transferring the system files.

Edit functions (^E, Alt+E): The hot-key will open the sector editing menu. The currently displayed sector can be loaded to the clipboard and rewritten to another location. In addition to read and write, there are options to read a sector from file, to write a sector to file and to decrypt an encrypted sector. The latter is especially useful to recover from the Monkey virus.

Track zero menu (^Z, Alt+Z): The whole track zero, head zero can be backed, restored, or compared with the **^Z** menu.

Restoration Functions Hot Keys

The Ctrl+function menu is displayed when pressing the slash "/" key. The following options are available under Ctrl+function:

^F1 Force a new master boot record (MBR), recalculated from the standard DOS boot sector found in sector 1,0,1. ResQdisk will use the PART.NTZ image file instead, if it exists. **^F1** will assume a SINGLE partition existed on the hard drive, unless there exists a PART.NTZ image with several partitions data. For disks with multiple partitions see **^F4**, below. This function requires that the copy of IV is registered.

^F2 Force a new active system boot sector, recalculated from the master boot record data. ResQdisk will use the BOOT.NTZ image file instead, if it exists. This function requires that the copy of IV is registered.

^F3 Restore CMOS setup data. ResQdisk will use the data stored in the CMOS.NTZ file if it exist. Otherwise ResQdisk will do nothing.

^F4 Scan the entire disk for additional partitions and rebuild the MBR accordingly. This option is available only to registered ResQpro users. The ResQpro function, **^F4**, can be enabled for a single on-line session by IV support agents. See the agents list.

^F10 Password to enable ResQdisk in Pro mode for a single session. Users in need can contact an authorized IV agent for online assistance. The agent can enable ResQdisk features for the current session and guide the user to help him regain access to the hard drive. The password is accessed by pressing **^F10**. Read the password to the agent and he will respond with a matching one that will enable ResQdisk.

IVSCAN - The Virus Detection and Removal Scanner

IVSCAN is a basic virus scanner with enhanced generic features. IVSCAN is not a substitute for an up-to-date virus signature scanner and it should not be used as such. When used properly following the screening procedures described in this manual, integrity based analysis using InVircible provides for a more reliable, thorough, and comprehensive detection of both known and new viruses than is possible using a known-virus signature scanner.

If you feel the need for a virus signature scanner to screen new software, or for disinfecting a machine before installing InVircible, then we recommend that you look for one of the many shareware or commercial scanners available from internet anonymous ftp sites, and online services such as CompuServe. It is essential that you understand the limitations of using such scanners. They will only detect viruses contained in the scanner's virus signature database. And, there is also the very real possibility that viruses known to the scanner will nonetheless be missed if they are active and in memory, or worse, spread throughout the machine by the scan process.

As a general rule, integrity based detection and restoration by IVB is superior to detection and disinfection by IVSCAN, or by other scanners, since integrity based analysis does not require identification of the virus. Integrity based detection and restoration is also exceptionally precise since it is based on information about the file itself, not specific viruses.

Scanners, IVSCAN included, may misidentify a virus strain for another or can be fooled into "finding" one virus while it is another. Disinfection based on such error will usually ruin the program and it may take a long time to identify and replace all of the corrupted files. In addition, many viruses add 1 to 15 bytes of junk code that are not removed by scanners. While this doesn't cause a problem in most cases, there are programs that check their self integrity and will stop functioning if not restored to their original condition byte for byte. IVB will do this better than IVSCAN and the vast majority of other scanners, as well.

While disinfecting a computer, with IVSCAN or else, the user must be aware of the possibility that a virus might be active in the computer's memory (memory-resident). Since a virus in memory has taken control of the computer, a general rule is to reboot an infected computer from the write-protected Rescue Diskette, or from a clean write-protected DOS diskette. The disinfection of a hard-disk must be performed only from the Rescue Diskette and not from the suspected drive itself, unless InVircible's programs specifically recommend execution in an infected environment (e.g. retro-Piggybacking).

Detection of, and disinfection from common viruses is done by IVSCAN. The program is activated via IV or directly from the command line. The command line syntax is:

IVSCAN [-] [where and what to scan] [/Option]

Parameters in [ ] are optional. If no option is chosen IVSCAN will detect viruses without removing them (the default). The default scan path are the current directory and its sub-directories. The use of wildcards (*,?) is allowed in specifying where and what to scan for.

The options are:

- **/R** to restore the file (if it can),
- **/D** to delete infected files that IVSCAN cannot recover,
- **/B** to recover or replace a suspected boot sector in floppies, or MBR on hard drive.
- **/EX** to scan executable files only.

The [-] parameter, when used, will limit IVSCAN to process the specified directory only.

Examples: To scan the entire C: partition for viruses in the Detection-Only mode type: **IVSCAN C:**

To use IVSCAN for detecting and eliminating viruses automatically only from COM files in the DOS directory, type: **IVSCAN C:\DOS*.COM /R -**

The space before the hyphen is required. If the hyphen is not used, this command will cause IVSCAN to begin at the specified directory and scan all its sub-directories.

If IVSCAN indicates "use IVB to restore" and the affected files were properly secured, then restore them with IVB. Use the Delete option as a last resort. A list of infected, deleted and restored files will be written to a IVSCAN.RPT file in the root directory of the processed drive, to help you replace the deleted files.

Before attempting the recovery of a file IVSCAN will prompt you with three options: Recover, Skip or All (disinfect all).

The Detection of Potential Droppers

Viruses are sometimes released in the wild by a technique known as a 'dropper'. The purpose of a dropper is to install a boot or MBR infector to a floppy or to the hard disk, or start an infection by either loading the virus to memory, or to drop it into a host file. The dropper could be the virus itself, but in a different form than its spontaneous offsprings, to avoid disclosing itself. A method that has been used many times is by applying runtime compression to an infected file and then removing data that gives away the compression.

IVSCAN will look for runtime compressed files, showing an attempt to conceal this fact, and alert the user that it found a potential dropper. The decision whether it's a genuine dropper or a benign file is left to the user. You can use the screening method to resolve this problem.

Generic Removal of Boot Viruses

Boot viruses are transferred from one computer to another by accidentally booting from an infected floppy (the infected diskette is left in drive A and the computer is rebooted), even if the floppy is not bootable, just infected! IVSCAN provides for the replacement of the boot sector in floppies with a standard DOS boot sector. Just type: **IVSCAN drive: /B**

IV contains a complimentary program, FIXBOOT, for bulk cleaning of the boot sector of floppy disks. FixBoot will install a generic clean boot sector to a diskette, in drive A or B. FixBoot can be used to prepare a clean bootable floppy, when the hard drive is infected with a boot infector. Insert a formatted floppy in the drive, transfer the system files with the DOS command SYS A: (or B:) and finally run FIXBOOT A: (or B:). Remove the floppy from the drive without attempting anything else, not even a DIR command, and write protect the floppy.

The above technique usually works even in the presence of stealth viruses such as Monkey.

Hard Disk Recovery

Recovering the data on a hard drive to which access was lost can be compared to detective work. Reestablishing access to the data consists in finding what were the configuration parameters of the hard drive at the time it was still functioning and restoring these parameters to original. The following is a list of these parameters, in the same order the computer reads them. This is also the order in which they should be verified and corrected, if necessary.

- **The hard drive setup in the CMOS.** These apply only to IDE and E-IDE drives. SCSI and the older MFM drives are identified by the hardware during the POST test.
- **The master boot record (MBR)**, also known as the master partition sector.
- **The boot sector**(s), especially the boot sector of the active boot partition, where there is more than one partition.

The hard drive setup in the CMOS. An IDE / EIDE drive (Integrated Digital Electronics, Enhanced IDE) is represented as a collection of sectors from 1 to N, where N is the capacity of the drive in bytes, divided by 512. IDE can be represented as different combinations of cylinders, heads and sectors per track, provided that the product of the three numbers equals the total number of sectors available on the drive as indicated by the manufacturer, or is smaller thereof. The manufacturer's parameters of the hard drive do not necessarily represent the actual physical geometry of the drive.

The translated geometry of an IDE drive can be selected at the time the hard drive is configured. Once set, it's important that the same setup parameters are maintained throughout the life of the hard drive. Inconsistent setup parameters from those that the drive was configured with will result in file structure problems such as file allocation errors or trash directories and erratic file names.

The hard drive setup parameters in the CMOS, as well as those used by the BIOS can be read with ResQdisk by pressing F5.

The Master Boot Record (MBR). The MBR is the first sector located on track 0, head 0 of every hard drive. It contains the bootstrap program and a table of parameters for up to four partitions residing on the same physical hard drive. The table is also known as the partition table.

Each partition data block consists of 16 bytes, altogether 64 bytes for up to four partitions. Each block contains information about whether the partition is active (bootable), the partition type, the head, cylinder and sector numbers where the partitions starts, where it ends, how many sectors are there before that partition and how many sectors are there in the partition itself.

The partition table layout can be inspected with ResQdisk. Press F6 to view the data in the master and second partition sector (where there is one).

The Boot Sector. Each partition starts with a boot sector. A boot sector consists of a bootstrap program and data that describes the partition to which the boot sector

belongs. The data is also known as the BIOS parameters block (BPB) since it's used by BIOS interrupt 13h to access the partition. Only physical local drives (fixed and removable) can be accessed with interrupt 13h. Network and virtual drives, like compressed volumes, RAM drives, assigned and **SUBST**ituted logical drives cannot be accessed with BIOS INT 13h. It's important since ResQdisk can access the hard drive with both interrupt 13h as well as with IDE port access.

The boot sector BPB data can be viewed with ResQdisk with the F7 function key.

Troubleshooting Hard Drive Access Problems

You may be having access problems to a hard drive for any of the following reasons or combinations thereof.

- **Hardware failure.**
- **Wrong hard drive type or wrong parameters in the setup.**
- **Bad partitions sector.**
- **Bad boot sector.**
- **Corruption to the FAT, the root directory or to both.**

A hard drive access problem will manifest itself by the message "invalid drive specification" when you try accessing its partition(s).

First, check that the hard drive is properly identified in the CMOS setup. Press F5 when in ResQdisk to access the setup calculator. IDE drives will have non-zero parameters in both the BIOS and IDE column, SCSI and MFM drives have only BIOS parameters.

Some older and slow IDE drives may not be identified properly as IDE, yet there is nothing wrong with them, these are just IDE pre-standard drives. The setup data gives a first indication that the hard drive is recognized by the hardware (for SCSI too!). Now you need to determine if the configuration of the partition and boot sectors fit. Both partition and system boot sectors are referred to by the general name 'boot sectors'.

Resetting a Bad or Forgotten Password

A bad, or forgotten password may deny access to setting of the CMOS parameters. The password data is usually stored in a reserved area of the CMOS. It's possible to reset the CMOS data to zero by momentarily short-circuiting its battery supply, yet this requires technical knowledge and opening the computer's cover. You may use ResQdisk for the same purpose as follows. You will need another computer with no access password in its setup, preferably one with a BIOS from the same make as the locked computer. Copy RESQDISK.EXE to a bootable floppy. Insert the floppy in one of the running computer's drives, change drive to where the floppy is and run RESQDISK /B (create a backup file of the CMOS data). Now boot the inaccessible machine from the floppy just prepared, start ResQdisk, press Control+F3 to restore the CMOS from the fake backup file and reboot the computer. Most chances are that the CMOS checksum will now be wrong, resetting the CMOS to zero and letting to enter new parameters.

Identifying Hardware Problems

ResQdisk's default is the analysis of the first hard drive MBR. In case the MBR cannot be accessed then a message "cannot read the drive" will pop up.

Check with ResQdisk (**F5**) that a hard drive is defined in the setup and can be found. If the drive is an IDE / EIDE then the right hand column should show sensible parameter and the drive model should be indicated at the top center. SCSI and MFM drives will only show the BIOS parameters, the right hand column will be blank.

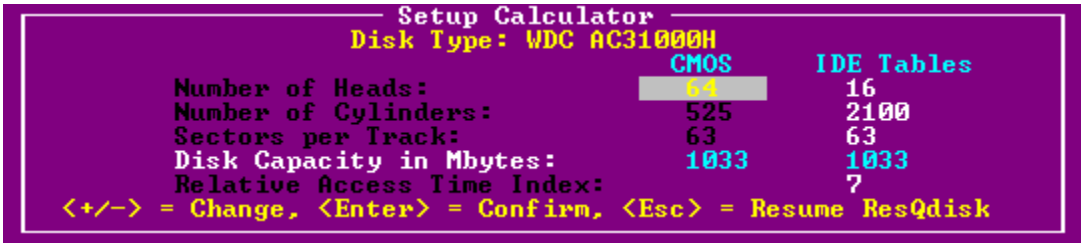
If the BIOS detects the presence of a hard drive and ResQdisk cannot read the MBR then there is most probably a hardware problem. Look for a defective controller, a bad connection of the power cord or the data flat cable, or a defective drive. Switch the power off and check that all cables are connected properly and that the power is connected to the hard drive. Make sure you can hear the hard drive spinning when power is switched on.

Erratic reading or writing to the hard drive could indicate a hardware conflict. Disconnect all IDE devices like a CD ROM by removing its IDE controller (usually the sound card). A modem could also be the source of an IRQ conflict with the hard drive controller. Reinstalling the devices one by one will help troubleshooting hardware problems.

Finding the Hard Drive Original Setup

An IDE drive having 12 heads, 987 cylinders and 35 sectors per track can be configured with different parameters. The drive will function properly with any of the following combinations of heads, cylinders and sectors per track: 15/720/38, or 32/762/17 etc. Note that the product of the three numbers is the same (or smaller) than the total available number of sectors as indicated by the manufacturer. ResQdisk will let you find the original setup parameters with which a hard drive was configured.

Start the BIOS setup procedure and let it auto-detect the IDE parameters. Older BIOS's may not have an IDE autodetect option. Select then any disk type that will let the computer boot from a floppy, the correct disk type has no importance at this stage. Start ResQdisk's setup calculator (F5) and write to paper the IDE manufacturer's parameters. Change the CMOS setup accordingly and reboot.



IDE analyzer and setup calculator, function F5

Verifying Configuration Compatibility

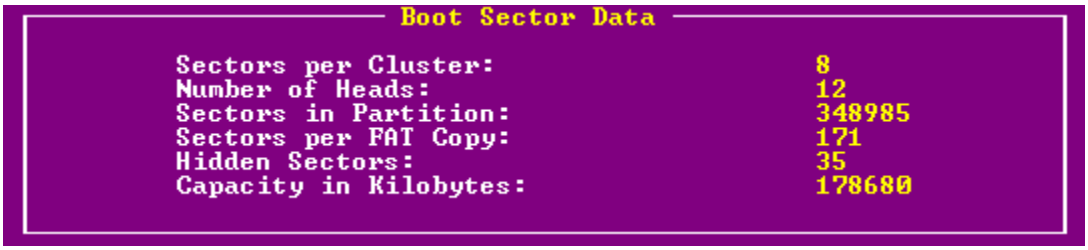
Run the following compatibility test even if the hard drive appears to function properly and self booted. Start ResQdisk and check that the **F5, F6** and **F7** functions all indicate the same configuration parameters.

Figuring Out a Non-Standard Configuration

Suppose that a drive with 12/988/35 heads-cylinders-sectors was configured and formatted as a 32/763/17 one. Also suppose that the CMOS setup data was zeroed and the autodetect found the manufacturer's IDE table and the data was written to the CMOS. Suppose DOS returns this message when trying to access the C: drive: "Invalid media type, Abort, Retry, Fail".

Partition table layout, function F6

Start ResQdisk and analyze the MBR data with **F6**. You will notice that the "Reserved sectors" indicates 17. Now try assessing the boot sector with the left key arrow, then **F7**. You should see garbage as the wrong setup parameters will point ResQdisk into the FAT, instead of the boot sector.



Boot sector layout, function F7

Yet if you browse track zero with ResQdisk you should notice a familiar boot sector pattern in location 0,0,18. The irregular place of the boot sector suggests that it could be the reflection of the lost boot sector, as interpreted with the wrong setup. Use **Alt+A**, "As boot sector" to analyze the sector content and you will soon find the missing information: the number of heads (32) and the sectors per track (17, in the "hidden sectors" field). With this information, and the manufacturer's IDE data as found with ResQdisk F5, you can now calculate the last missing parameter. The correct number of cylinders in the setup should equal the product of the IDE parameters, divided by the configured number of heads, divided by the number of sectors per track:

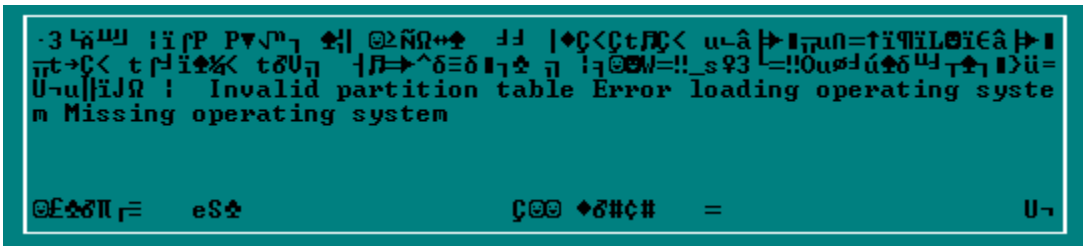
12 x 988 x 35 / 32 / 17 = 762.9

Restore the 32/763/17 parameters in the CMOS setup and the drive should resume normal operation.

Reconstruction of the MBR

-
-

ResQpro and advanced users.

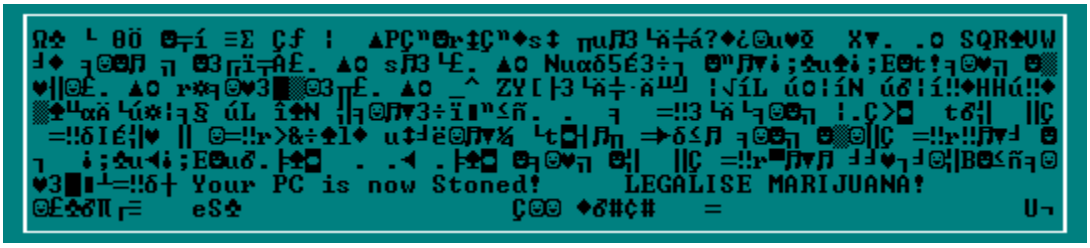


View of typical MBR

Let's take a look at the content of the master partition sector, often referred to as the master boot record (MBR). The MBR is always the very first sector on a hard disk, where it can be found by the BIOS program when booting the computer. The above is displayed in ResQdisk's clipboard when observing the MBR. There is no need to "understand" the meaning of the funny characters above, it's enough that the computer knows what to do with it. Yet, it's worth becoming familiar with the look of a normal MBR, so that you can tell what's wrong with it and fix it.

The MBR shown above is typical to a partition created by DOS' FDISK. The MBR will differ slightly depending on the partitioning program used when configuring the disk. Yet the general structure is maintained in all.

A normal, bootable MBR contains a bootstrap program (the upper part, with the failure messages), the partition table data with up to four partition records of 16 bytes each, and the "U" signature at the sector's end.



MBR infected by Stoned-Marijuana

A glance with ResQdisk at the MBR will reveal the nature of the problem. You should be able to read failure messages in the bootstrap code of a healthy MBR ("Invalid partition table, Error loading operating system, Missing operating system" in the example above). The lack of error messages may suggest that the bootstrap code was overwritten by a virus. If ResQdisk flashes "probably a virus" then you can be quite sure it's virus code. The substitution of the ".3" string at the beginning with a Greek ***delta***, ***theta*** or ***omega*** may also suggest a virus infection, although not necessarily so! Some rare or old configuration schemes have one of these characters at their beginning. You will later see that the DOS boot sector starts with a delta and it isn't infected. A Greek delta just

denotes a JUMP NEAR instruction in assembler language.

An empty partition table indicates that no partition is defined. Some viruses like Monkey overwrite the partition's data with nonsensical data. Pressing **F6** will show the content of the partition table in legible format.

The "U" signature is required by the BIOS and the OS in order to interpret the sector as a 'boot' one. Both partition and boot sectors must have that signature at the end of the sector.

Reconstruction of the MBR

There are several ways to reconstruct an invalid master boot record (MBR). The best is of course to restore the MBR from backup, provided one exists.

Many MBR infectors, although not all, will relocate a copy of the original sector to somewhere else on track zero. *Stoned*, *Michelangelo* and *NoINT* will relocate the MBR to sector 0,0,7, *AntiEXE* relocates the MBR to sector 0,0,13, *B1-NYB* to sector 0,0,17 and so on. To recover the original MBR just copy it with ResQdisk (**Alt+E, Read**) and paste it back to 0,0,1 (**Alt+E**, then **Write**). Others, like *Monkey*, encrypt the MBR before relocating it elsewhere. Monkey relocates the encrypted MBR to sector 0,0,3. ResQdisk Pro contains a special routine to decrypt *Monkey*'s encryption (**Alt+E, Decrypt**).

Before restoring a relocated MBR, check with ResQdisk's analysis tool that the sector belongs indeed to that hard disk. Press **Alt+A** and select "As partition". The parameters displayed should make sense and should fit the setup parameters (press **F5** to see them), as well as the boot sector ones (press **F7**).

When the original MBR cannot be found and a good boot sector exists (left arrow key) then the MBR can be reconstructed from that boot sector. Press **^F1** to force a new MBR based on the boot sector's BPB (BIOS Parameters Block). Take care to work from a floppy with no backup image files (no PART.NTZ, nor BOOT.NTZ), otherwise ResQdisk will use the backup. The image may not belong to this disk!

Finally, when both the master partition and the boot sector are damaged then use **^F4** (or **Alt+F4**) to search the disk for existing partitions. ResQdisk will pause when a potential partition or boot sector is found and display its table data, as a partition sector first. The presentation can be toggled to "boot sector" with **Alt+B**, in case the partition table does not make sense. A physical partition may start by either a partition sector or a boot sector, both are valid and recognized under various operating systems. Write the data down to paper, you will need it to finalize the editing of the reconstructed MBR, in a while. The partition type has special importance, write it down as ResQdisk does not retain the data found when reconstructing the MBR, it takes into account only the sector coordinates.

When done with the search, ResQdisk will display the reconstructed MBR and save it to the clipboard. You will be prompted to write the MBR to disk. You may write it to disk immediately or later if you prefer. It is convenient to store the clipboard content in an unused sector of track zero. Select the sector you wish to use for the purpose, press **Alt+E** and select "Write". Now you may edit that sector before finally writing the new MBR back to sector 1.

Manual Editing of the MBR

ResQpro and advanced users.

Although there is room for up to four partitions in the MBR tables, only two partitions are usually written to the master partition sector (the one in sector 0,0,1), regardless of the number of the partitions existing on the drive. The MBR will contain the first active boot partition and a second partition containing the rest of the drive, as an extended DOS partition. Further subdivision of the extended partition will be contained in the extended partition sector. You can use ResQdisk's inspection tools to find whether a partition is defined in a lower hierarchy partition sector, or it should be defined in the master partition sector. Partitions other than DOS types like Novell and Unix are usually defined in the master partition sector.

Special Partition Types

The following are a few common partition types. The highlighted number corresponds to the 'type' byte in the partition's record.

01	DOS-12, found in older DOS versions based on 12 bit FAT.
04	DOS-16, primary DOS, for a partition to 32 megabyte.
05	Extended DOS.
06	Big-DOS, primary DOS partitions, larger than 32 megabytes.
07	HPFS, NTFS (for OS/2 and Windows NT).
10	OS/2 Boot Manager.
18	Compaq diagnostics and utilities.
80	Older versions of Disk Manager.
84	Disk Manager Dynamic Boot Overlay.

Editing the MBR

ResQdisk Pro assumes a standard DOS FDISK partition scheme, by default, in which the active primary DOS partition always comes first, starting on head 1, cylinder 0, sector 1. For partition configurations that differ from standard, manual editing of the MBR will be required. A couple of nonstandard configurations are mentioned below.

Disk Manager Dynamic Drive Overlay (DDO). Look for the DDO topic for a discussion of large capacity IDE drives and dynamic boot overlays.

Compaq Models Partitioning

Most models of Compaq come with a unique partition configuration. The first 3 megabytes of the hard disk are allocated to a non-DOS partition reserved for Compaq diagnostics and utilities. The examination of the hard drive with FDISK (option 4, display partition data) will show the non-DOS partition. The primary DOS partition will usually start on cylinder 5 to 7, depending on the number of heads and sectors per track. Since DOS does not recognize the special partition then the first logical drive (C:) is allocated

to the other partition.

Scan the hard disk with **^F4** to reconstruct a new MBR. Next change the following with ResQpro's MBR editing utility (**Alt+M**).

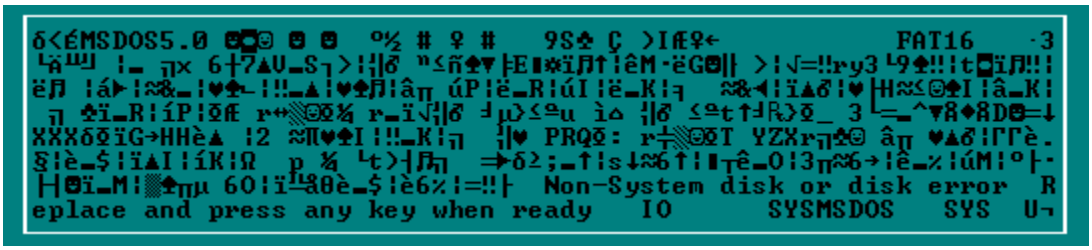
Partition 1: Change the partition 'active' flag to inactive. The partition type should be changed from 4 (DOS-16) to 18 (Compaq's special type).

Partition 2: Change the active flag from inactive to 'active' and the partition type from 5 (extended DOS) to 6 (primary large partition DOS).

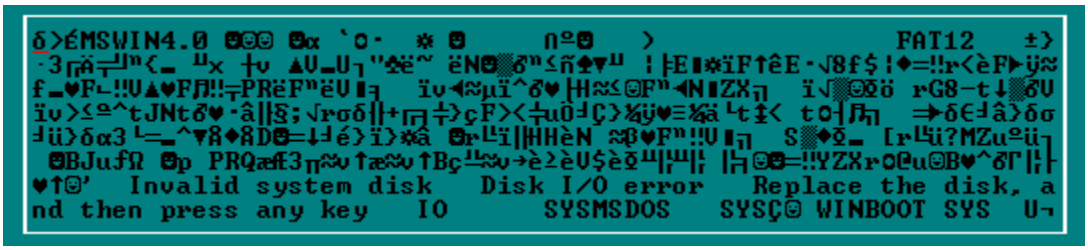
After reconstructing the partitions sector from scratch you'll also need to reconstruct the boot sector, described next.

The Active Partition Boot Sector

Copy
Copy



Typical MS-DOS boot sector



Typical Windows 95 boot sector

Like with partition sectors, a boot sector contains a bootstrap program, parameters data and the boot signature double byte.

The boot sector is the first sector in every partition. It contains information about the operating system that owns the partition as well as the necessary information about the structure of the partition. The latter is contained in the BIOS Parameters Block (BPB) which makes part of a boot sector. When the computer is started, the BIOS takes care of loading the bootstrap program contained in the MBR. Control is then passed to the active partition boot sector, its bootstrap program is read and executed. Finally, control will pass to the operating system and DOS or Win95 will start loading.

Observe the following features in a boot sector: The name of the operating system that governs the partition (note that DOS 6.x has a DOS 5 boot sector OEM), then the BPB (the gibberish on the top line). The OS bootstrap program occupies the rest of the sector. A "U" signature terminates the boot sector code. Note the boot failure messages contained in the bootstrap program, as well as the names of the system files. Note the difference in the OEM and in the system filenames between a DOS boot sector and the Windows 95 one.

The BPB data can be assessed using **F7** when running ResQdisk or with **Alt+A** and "analyze as boot sector".

Symptoms of a bad boot sector. A bad boot sector, unlike an infected one, will not let the operating system boot sequence continue and access to the logical drive partition will be denied. An overwritten boot sector, or bad BPB data will return the message "drive reading error, Abort, Retry, Fail" when attempting to access the partition.

Another common problem with a bad boot sector is the failure to boot the operating system, although the active partition can be accessed from external boot.

CAUTION! Do not write, delete or rename ANYTHING, nor 'SYS' a disk that has access problems or file structure damage before you have fully assessed the situation.

There could be a mismatch between the partition configuration and the BPB data contained in the boot sector. Any alteration to the FAT in such case will result in severe data integrity loss, sometimes beyond repair! File and directory structure damage requires utmost care, or you may not be able to recover the data.

Precautions in Hard Disk Recovery

Avoid disk repair utilities in the first stages of a disk recovery. Utilities to avoid are SCANDISK, NDD, DISKFIX or alike. You may use CHKDSK, but without the /F (fix errors) switch. Preferably, run all programs from a floppy to avoid writing to the hard disk.

Avoid the use of a program management shell such as DOSSHELL, Norton Commander or their like. Temporary 'piping' files are sometimes created when calling a program from a shell. These files could be written to disk and overwrite the FAT or part of the root directory. Run all programs from plain command line syntax.

Don't be misled by an apparently empty partition or one that seems to contain 'garbage'. The missing data may all be there, latent and intact. Be careful not to overwrite it and ruin the chance of recovery. A mismatch between the partition geometry and the BPB may manifest or present itself as an apparently empty or trashed partition.

Boot Sector - MBR Mismatch

When the boot sector's BPB and the partition table's data disagree, which is correct? Sometimes you may find that a logical drive partition's boot sector parameters and its partition (MBR) do not agree. Follow this rule: If a drive partition is accessible and reads correctly (directory and file structures appear intact, CHKDSK indicates no allocation errors or cross linking) then the boot sector is correct. The partition (MBR) and the boot sector may be discordant in some parameters without affecting operation, yet they should both be consistent about the partition type and the boot sector's physical coordinates (the head, cylinder and sector where the partition begins). For the other parameters, the boot sector BPB values take precedence over the MBR's partition values.

Number of Heads and Sectors per Track

Whenever a boot sector exists and can be found, then you can find the number of heads and sectors per track with which the partition was configured. Just read the BPB data with **Alt+(A)**nalyze, As boot sector. You can read both parameters directly or calculate them from the BPB. The hard drive parameters (number of heads, cylinders and sectors per track) in the CMOS setup and in the BPB must match each other for an IDE drive. These parameters are loaded automatically by the SCSI adapter while

booting.

Wrong Boot Sector Style

Booting a partition requires that the boot sector is of the same type as the installed operating system. Boot sectors differ by the operating system itself (DOS, Win95, Windows NT, OS/2 etc.) the OEM (original maker of the operating system) and the version. A worthy piece of information is that MS-DOS 4 to MS-DOS 6.x use the same boot sector, just like PC-DOS 4 to IBM/Novell DOS 7.x use the same boot sector. An MS-DOS boot sector on a Windows 95 system, or vice versa will simply hang the computer in the boot sequence, with no indication what's the problem. Yet you can access a Win95 partition by booting from a DOS floppy and vice versa.

Wrong Data in Bios Parameter Block (BPB)

Erroneous parameters in a boot sector's BPB can have quite odd effects. For example, wrong data in any of the following parameters, may let you access the partition, yet result in allocation errors and massive cross-linking of directories and files: ***number of sectors in partition, number of heads, number of sectors per track*** and ***number of sectors preceding the boot sector (hidden sectors)***.

Click here for reading how to edit the boot sector.

Reconstruction of the Boot Sector

First assess the nature of the problem! ResQdisk Pro provides several ways to reconstruct the boot sector, depending on the nature of the problem. The DOS SYS command may be used to refresh a defective boot sector provided the partition can be accessed. Still, SYS should be avoided in a disk recovery session since a system transfer will take place, files will be written to disk and in case of boot mismatch, access to important data may be lost. Therefore, you may use SYS C: only if drive C: is accessible and there is no damage to file or directory structure. In all other cases use ResQdisk Pro instead.

The standard ResQdisk procedure for reconstructing the active boot sector (from the MBR) is Control+F2. If there is no access to the partition then ResQdisk will search for an existing FAT and reconstruct the sector accordingly. If no FAT is found then ResQdisk will continue with a "blind" recovery. The default boot sector will be a MS-DOS-5 one. Yet, if C:\ is accessible then ResQdisk will assess whether the installed operating system is MS-DOS, PC-DOS or Windows 95 and match the bootstrap program accordingly. Therefore, you may need to fix the boot sector of a Win95 disk in two phases: In phase one you will regain access to the boot partition, with a MS-DOS 5 boot sector as default. You then need to reboot from a floppy, verify that partition C:\ can be accessed and run ResQdisk once more to match the correct type of boot sector to C:. You may use **^F2** again, or ResQdisk's boot sector refresh sequence (Left arrow, F1, F4), or IV's equivalent of DOS's SYS, but without transferring the system files (**Alt+B, SYS**). Now the boot sector should fit the installed operating system and the hard drive should boot normally.

The methods described can be used whether the damage to the boot sector was caused by a virus or else.

Editing the Boot Sector

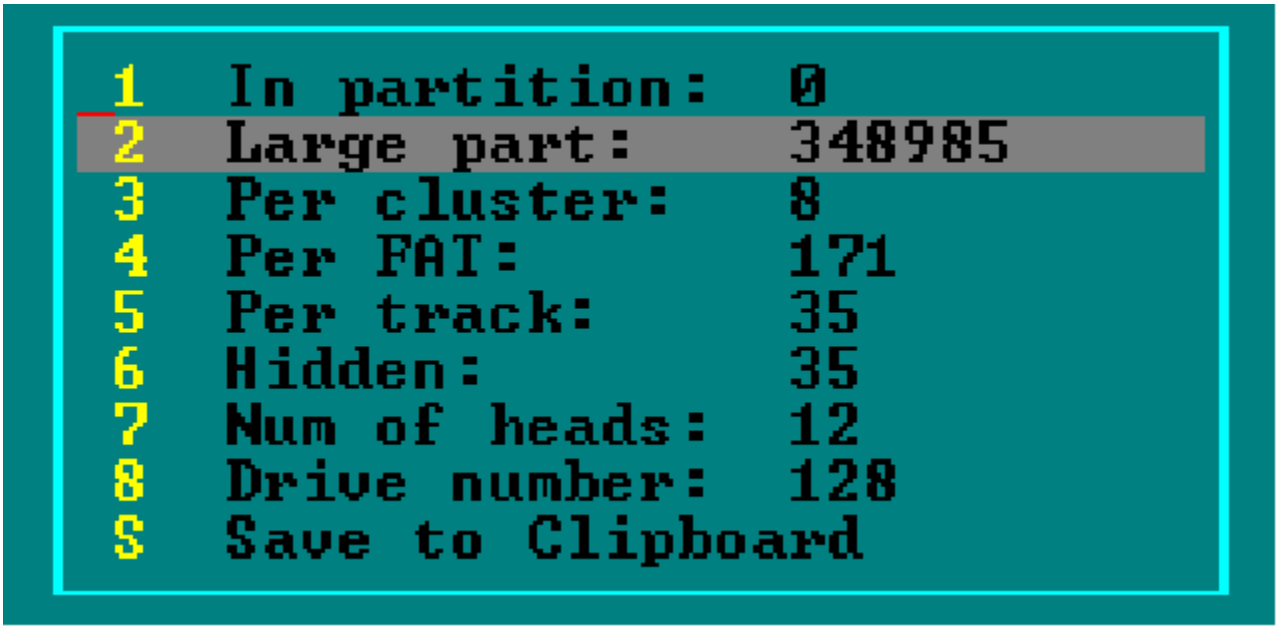
Copy
Copy

ResQpro and advanced users.

Incorrect data in the BIOS Parameter Block (BPB) of a boot sector may cause serious problems in reading and writing data in a logical partition. Here are a few examples: An incorrect number of ***sectors per FAT*** in the BPB or of the ***sectors per cluster*** will result in an apparently empty partition or may show as garbage. A discrepancy between the number of ***total sectors in partition*** and the actual number of sectors in the partition will result in allocation errors as well as cross linked files, when verified with **CHKDSK** or **ScanDisk**.

ResQdisk's BPB editor is especially useful in the recovery of non-standard configurations and partitioning schemes such as those created by Ontrack's Disk Manager or in most Compaq's models.

To enter the **BPB editor**, start by selecting the boot sector of a logical drive. Press **Alt+B** (or **^B**), select the drive letter and **Read** the sector to the clipboard. Now press **Alt+N** (or **^N**) to enter the editor. The **Boot Sector BPB Editor** functions only if a sector is loaded to the ResQdisk clipboard. You may practice with the BPB editor, nothing will be written to disk until you select **Write** from ResQdisk's selection menu. Therefore, be careful not to write to a disk sector unless this is really what you want to do.



Boot sector's BIOS Parameter Block editor

The BPB contains additional parameters, not mentioned above, such as the sector size (always 512 bytes for physical DOS drives), number of root entries (512 in hard drives), number of FAT copies (always two) etc.

Number of sectors in small partition:

A partition that contains less or exactly FFFF (hex, decimal 65,536) sectors is recognized as a "small" partition. The maximum size of small partitions is limited to 33 megabytes (65,536 x 512 bytes, exactly). The number of sectors **in partition** will be indicated in the first field of the BPB editor, above.

Number of sectors in large partition:

This field contains the total number of sectors for large partitions (bigger than 33 megabyte). Note that only **one** of the two field contains a non-zero value, either this one or the previous one.

In case the number is incorrect, in either of the two fields, then enter the correct number displayed in ResQdisk, function F6 under "Total Sectors". Select the field to update according to the partition size as explained above.

Number of sectors per cluster:

The number of sectors per cluster in a partition depends on the total number of sectors in the partition and is always a power of 2. The correct sectors/cluster ratio for a given partition size is calculated by ResQdisk and is displayed on the help line (at the screen bottom) when selecting **Per cluster** and pressing Enter.

Number of sectors per FAT copy:

The number of sectors per FAT copy (there are two identical copies of the FAT) depends on the previous parameters: the total number of sectors in partition and cluster size. Assure these parameters are set correctly in the BPB. The number of sectors that should be in a FAT for a given partition size, with a given cluster size is calculated by ResQdisk and is displayed on the help line (at the screen bottom) when selecting **Per FAT** and pressing Enter.

An incorrect number in this field will result in either an apparently empty partition or will show garbage under a **DIR** command. If in doubt then see next in "***Searching for Existing FAT Partitions***".

Sectors per track:

The value should conform with the one indicated in the setup. This parameter affects the way the partition is addressed with interrupt 13h.

Number of hidden sectors:

The value should either equal the number of sectors per track if the partition begins with a partition sector, or be zero if the partitions begins with a boot sector. This parameter affects the way the partition is addressed with interrupt 13h.

Number of heads:

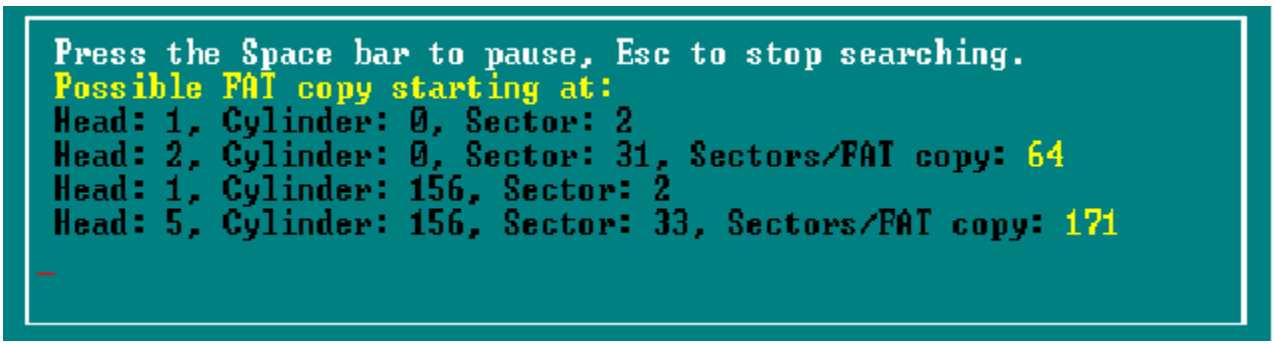
The value should conform with the one indicated in the setup. This parameter affects the way the partition is addressed with interrupt 13h.

Hard drive designation in BIOS:

The first hard drive is designated as 80h (128 decimal) in BIOS notation. The hard drive will not self boot if this parameter is incorrect.

Searching for Existing FAT Partitions

Finding the existing file allocation tables (FAT) on a hard drive can help in the recovery of DOS partitions existing on a hard drive. To search for FAT pairs press **Alt+F5** or **^F5.** The example below is from a hard drive with two partitions.



Searching for existing FAT partitions

Note in the above example that both **first FAT copies** are found on head number 1. Note also that the first copy of a FAT always starts on sector 2, since sector 1 on the same head (and cylinder) is occupied by a boot sector. The location of the FAT reveals important things about the partitioning geometry.

If a partition's FAT starts on head 1 then there should also be a partition sector located on head 0, on the same cylinder. If the first FAT copy starts on head 0 then there is no dedicated partition sector for that partition. The partition is then defined in the MBR or in an extended partition sector, with multiple sub-partitions.

Fixing Damage to File or Directory Structure

Copy
Copy

One of the worst things that can happen to a PC is damage to the File Allocation Table (FAT). The FAT is the table of contents of a hard disk and without it would be impossible to access a single file on the drive. There are many causes for FAT damage, some we can control, others have a random nature and happen without notice. Yet viruses are the least likely to cause FAT damage. From the 7,000+ existing viruses at this date, perhaps only a couple have a payload that will deliberately mess with the FAT. Much more common is FAT damage caused by careless computing habits (switching a computer off before it finished flushing the cache buffers) or sheer ignorance, like playing with a disk editor, or using a disk repair utility where it should be avoided.

File structure damage does not always show immediately. It may start with file allocation errors, or cross-linked files. If not taken care immediately, file structure damage will worsen, it never gets better by itself. Severe FAT corruption will show by the disappearance of files or directories, and the appearance of trash files and directories bearing odd names, containing high ASCII and control characters.

Minor FAT corruption, such as lost clusters, or a few allocation error can be handled with the standard DOS tools like CHKDSK/F or SCANDISK. Crosslinking should be handled carefully as the situation tends to worsen when resolving cross-linked files. It is recommended to delete both cross linked files. If you need to backup one, or both files then backup to a different drive than the one that has the crosslinked files before deleting them.

The real challenge is when massive FAT corruption occurs. You boot the computer and to your surprise there is nothing on the hard drive, or there is just garbage. Or, you may have installed a new Windows application, the installation hung, you hit the reset button, and then the hard drive is a total mess.

This is when you need DOS' most powerful recovery utility: UNFORMAT. UNFORMAT and MIRROR are a pair of complementary utilities that were licensed from Central Point and bundled with MS-DOS 5. Only UNFORMAT is bundled with MS-DOS 6.x. MIRROR creates a backup of the drive's critical areas including the FAT. UNFORMAT restores the FAT from the MIRROR data, as explained elsewhere in this manual. Yet, it is less known that UNFORMAT can help in recovering files and directory structure even without a MIRROR backup, by using the /U switch (don't search for MIRROR files, ignore them if they exist).

UNFORMAT will search the partition for directories and reconstruct their tree, ignoring the content of the root directory. The search will start from the end of the partition down to the root. The difference between UNFORMAT and the common disk repair utilities is the strategy. The latter rely on the root directory and FAT and work their way 'bottom up', while UNFORMAT assumes nothing about the root and FAT and works 'top down'. Therefore, UNFORMAT should be preferred when the root or FAT do not exist or are severely corrupted. InVircible's rescue diskette contains a copy of UNFORMAT.COM just for that purpose.

Improving Data Survivability and Recoverability

Copy
Copy

Copy Prepare an **IV Rescue Diskette**.

Copy Make **regular backups** of your critical data.

Copy To facilitate data recovery **use a FAT (i.e. MIRROR or IMAGE) mirroring utility** in the autoexec.

Copy **Defragment your hard drive regularly**. About once a week will do. The percentage of the recovered data in case of file structure corruption is significantly higher with unfragmented drives.

Copy And last but not least, **practice data recovery** techniques on a test machine. Hands on experience is invaluable when the real thing strikes!

ResQdata, The Data Recovery Module

ResQdata is InVircible's non-destructive file recovery utility as implied from its name. ResQdata's purpose is to retrieve files from a hard drive or floppy when the data cannot be accessed anymore through the file system. ResQdata can be used to advantage in the following cases:

- When a directory or tree section containing important data files was accidentally erased, corrupted or destroyed.
- When part or the whole file system was corrupted or ruined and the problem could not be fixed with disk repair utilities. Or as happens more often, when an attempt to repair the file system resulted in more damage to the file system.
- To recover a file that had been set to zero length by either user mistake or by an application error.
- To recover data from bad floppies, in conjunction with two other NetZ utilities: ResQflpy -- the floppy cloning utility -- and FixBoot, IV's floppy boot sector repair utility.

ResQdata can either search for default data files or for user defined ones. In the default mode, ResQdata will search for a fixed set of data files, including Word, Excel, PowerPoint and Quatro's. In user guided search, ResQdata will look for files having the same characteristics as the sample provided by the user. This way, ResQdata can recover a large variety of database types and formats, even user unique databases, provided the files have typical and recognizable features.

ResQdata works only on drives organized under a FAT file system.

The extent to which ResQdata will successfully recover files depends primarily on the fragmentation level of the sought drive. Contiguous files can be fully recovered, while the recovery of fragmented files will yield incoherent data fragments.

ResQdata has additional and important uses in data recovery. It can be used to verify whether correct drive geometry and parameters were selected. If the drive parameters and boot sectors are correct, then ResQdata should be able to spot files on that drive of the type known to exist on it. If ResQdata finds none, or only few, then this could indicate a bad selection of parameters in the setup, wrong partition data in the MBR or incorrect BPB (BIOS parameter block) in the boot sector. See in the ResQpro section for details on how to fix these problems with ResQdisk.

To actually recover data from a hard drive, ResQdata should be run from the special InVircible ResQ Professional floppy. In its freeware mode, ResQdata will only search for data files on the hard drive but will not recover them. Still, ResQdata will recover data from floppies, for both practicing purposes and real data recovery.

ResQdata Setup

The following are needed for a ResQdata session:

- A source drive from which to read the data. The source drive must be accessible, to the level of being able to change the current directory to the source drive. This would require that the hard drive parameters in the CMOS are correct and that the master partition sector (MBR) and the boot sector(s) have correct data in them. In case a special driver is required to access either the source or the target drives, like the Ontrack DDO or else, then load the driver via the config.sys or the autoexec, as required.

- A save drive, where ResQdata will write the files it retrieves as well as the session report. Any logical drive with sufficient free space on it can serve for the purpose, including floppies or network drives. Most convenient is an external ZIP drive, or you could use a second hard drive, either master or slave for that purpose.

- The ResQdata software. ResQdata must run of the special IV Professional floppy for actual recovery of data from a hard drive.

- Samples data files of the same type as sought on the source drive.

Boot the computer with the same operating system as installed on the source drive before it crashed. This could be important with the newer versions of Windows 95 as there may exist Win95 partitions that are not recognized by DOS, only by Win95.

Boot of a floppy, preferably. Data recovery should be conducted under real 16 bit DOS (not in protected mode), from either DOS 5.0 or higher, or under Windows 95, before Windows started its 32 bit driver.

Check that the source drive is accessible by changing to it. Use ResQdisk Professional if necessary to fix access problems in case DOS returns an "invalid drive specification" message (bad partition data) or "invalid media type" (bad parameters in the boot sector).

Start ResQdata and select the search mode first, to assure that the drive setup parameters are correct. If okay, then ResQdata should be able to spot files that are known to exist on the sought drive. If no files can be found in the search phase then the reason could be different drive configuration parameters from the ones with which the drive was configured. Refer to the disk recovery section in this manual for details on how to fix the problem.

ResQdata will generate a report, display it to screen and save it to the target drive with the name of RESQDATA.RPT. The report contains useful information that could help in assessing the desired startup parameters for recovering your files. It's recommended that you print the RESQDATA.RPT report after each run.

ResQdata Startup Options

When started, ResQdata will prompt whether to search for default files or **use a sample** file, designated by the user. Press Enter for the default, or "S" for sample.

Give the **sample's pathname** when prompted. DOS wildcards (*,?) are allowed. ResQdata will reply with the sample's full pathname if found, or abort if it cannot find the sample.

When in user designated mode, ResQdata will ask to enter an **extension filename** for the retrieved files. Enter a three characters extension name such as RTF, GIF, PCX, DBF etc. ResQdata will add the given extension to the saved files name. The extension in the default mode is selected automatically. MS-Word files will be saved as *.DOC, Excel files as *.XLS, Power Point as *.PPT and Quatro files will be saved as *.WQ1. Files that have been created by Microsoft Office but cannot be classified as belonging to one of the above categories are saved with the extension OFS (for Office).

The next choice in ResQdata's dialog is what **correlation factor** should ResQdata use? The decision whether a matching file was found is based on statistical correlation derived from the InVircible correlator, IVX. There are differences between data files, even if from the same application. Asking for a too high correlation factor will cause ResQdata to miss many instances, while a too low factor will yield excessive hits with a lot of garbage files. The default correlation factor of 4 (from the range of 1 to 5) works well for most databases. Yet there are instances in which you may need to decrease the factor, like when looking for archive files (ARJ, RAR, ZIP etc.). The best factor for a particular database can be found by trial and error. Run ResQdata with different factors and analyze the report on each run.

ResQdata will next prompt for the **source drive**. Just press the drive letter key.

ResQdata will ask if to run in **search-only mode** (the freeware default). If running of an InVircible Professional floppy then the default is the full mode (with file saving to the target drive). Press Enter for the default, or answer "Yes" for searching only, without saving.

Next, you will be prompted for the **target drive**, where to write the report and the recovered files. Select the target drive letter, from A to Z. ResQdata will check that the source and target drives are not identical and abort in case they are. A rule in data recovery is to NEVER WRITE ANYTHING to a drive being rescued, unless its integrity has been checked and confirmed. Writing to a corrupted drive will worsen the problem and overwrite data that could be recovered otherwise.

When in full mode, with the "save files" option set, ResQdata will prompt for the desired **file-length** of the recovered files. A rule of thumb is to chose a length that is larger by a few kbytes than the largest file you expect. See later how to truncate garbage from data files and restoring them to their original length. In case a recovered file was too too short, then retrieve it a second time with a larger selected size, using the report for localizing the cluster at which the file starts.

In full mode, ResQdata will ask if to **prompt before saving** a file. The default is automatic saving of found files. Saved files will be named "1000.ext", 1001.ext" etc., where "ext" stands for the default extension name, or the one selected by the user. Saving upon confirmation is useful when you wish to recover a specific file at a known location.

The last parameter in the ResQdata startup dialog is **from which cluster** it should start

searching. This choice is useful when you need to retrieve a specific file, knowing its exact location from, a previous ResQdata scan or report. Data recovery is continued in next page.

ResQdata Runtime Options

Pressing the **space bar** will pause ResQdata. The search will resume upon pressing 'space' again.

The **+/- keys** are used for changing the search location, in increments or decrements of 200 clusters.

ResQdata will prompt for replacing a full floppy, if a floppy drive was selected as target.

Restoring the Original File Size

Files recovered with ResQdata do not have their correct original size. To successfully recover the data in a saved file, the latter should be larger than the original one by a fair margin.

Yet it's simple to restore the recovered files to their original size. Just open the recovered file with its appropriate application (e.g. open DOC files with MS-Word, XLS with Excel, GIF with PaintShop etc.) and save them to disk.

The normal opening of the file confirms that the data is valid. Saving the file will truncate the excess garbage code and restore it to the correct size. This could also be the proper occasion for renaming the restored file, by saving it under a new filename.

Practicing Data Recovery

You can practice ResQdata by searching various files types on your hard drive, or by actually retrieving files from a floppy. Practicing actual data recovery from a hard drive requires that ResQdata runs of the special InVircible Professional floppy.

To see the effect of defragmentation on recovery, then run the following test. Format a floppy with the /U switch (unconditional formatting) and copy a few Word docs to it. Now, run FORMAT A: /Q (quick formatting) so that the FAT and root directory are blanked (UNDELETE won't be able to see the files anymore). Now restore them from the blanked floppy with ResQdata (use the hard drive for saving the recovered files). You will notice that the files will recover properly. The reason is that the files were written contiguously by DOS, since the floppy was freshly formatted.

Try now a different scenario. Copy files to the floppy until DOS complains "insufficient disk space". Erase a few files of the floppy, picking them at random, and then copy more files until the floppy is full again. Apply FORMAT /Q to blank the floppy and try restoring the files again with ResQdata. You will notice that this time, less files will recover properly. The reason being the fragmentation, caused by erasing and rewriting data to the floppy.

Regular defragmentation of your hard drives is good practice. Except for its other advantages, it also increases the chances of successfully recovering your data when

needed.

Customized Versions of ResQdata

ResQdata will work as is with most database. Yet, it's possible that ResQdata will indicate on certain databases that "the sample has no recognizable features". This doesn't mean that your database format cannot be recovered. You may simply need a customized version of ResQdata, to suit your specific database format.

NetZ Computing will be glad to look at your problem and design a customized version of ResQdata. To investigate whether your data can be restored with such version, please e-mail to netz@actcom.co.il and attach a few samples of your database files (not less than three and no more than ten), in ZIP format.

Primer to Generic Antiviral Methods

Generic techniques can be used to cure damage from groups of viruses such as boot block infectors, stealthy file infectors and cluster infectors.

First, assess the type of infection by its symptoms, then test the selected method on a sample diskette or directory. If successful, then apply the selected technique onto the entire infected disk.

Symptoms of Virus Presence or Doing

Computer viruses always disclose their presence sooner or later. InVircible alerts the user when sensing various effects that can be related to computer viruses. Among InVircible's messages, the following are of special interest:

(1) Missing memory.
(2) A stealth virus found active.
(3) Owns file size "changed by 0 (zero!) bytes"
(4) Piggybacking or file killer detected.
(5) "faked partition (or boot) sector".
(6) "Boot infector detected".

Message (1) indicates memory stealing, attributed mostly to the presence of a boot or partition infector. Certain file infectors do the same, and this possibility should be first discarded. Also, some programs such as DesqView uses a few kbytes kernel in RAM, or certain BIOS use 1 kbyte of RAM when the machine was booted from a floppy, without being related to virus activity.

Message (2) or (3) indicate that a stealth virus is active in the system. These messages are generated by IV's baits or the self integrity check, respectively.

Message (4) indicates that a Piggybacking process was detected and the search should be halted to prevent further spreading of the virus. This message is generated by the either one of IV's scanning programs, IVB, IVX and IVSCAN.

Message (5) indicates the presence of a stealth virus in either the MBR or in the system boot sector. This message is issued by SeeThru, implemented in IVINIT, IVTEST and IVSCAN.

Message (6) will be displayed when a boot infector is found in the MBR or the system boot sector of the hard disk or of a floppy. This message and message (5) may sometimes be displayed both.

What to Do in Case a Virus is Found

It is recommended that inexperienced users ask for assistance or leave disinfection to qualified and trained personnel. Inadequate virus removal procedures can be incredibly

more harmful than continuing working with a virus in the computer!

In case of doubt, then **leave the computer on and working** until help arrives, and don't continue working while the virus is still active. You next best choice is to first backup valuable data and then switch the computer off in an orderly manner. Do not backup programs as they may be infected, and do not switch the computer off before having backed up important data. Act as if you might not be able to access the data on the hard disk the next time you restart the computer.

Collecting Information on an Attacking Virus

It is important to collect as much as possible information on the attacking virus as it will let you make informed decisions of how to best recover from the attack. The following are a sequence of tests to conduct, that will help you finding the facts about the virus.

A virus can be a boot infector (MBR and-or a boot sector infector), it can be a file infector, a companion virus and it can be both (a bipartite or multipartite). There is also the slim possibility that you are dealing with a cluster infector, but this possibility is so rare that you may ignore it, at least in the first stages of your investigation.

First run the IVTEST probe a couple of times. It will indicate whether there is memory stealing, check for boot infection and boot stealth, ant try hook the virus with a series of different baits. Boot infectors will be sampled into a PARTVIR.NTZ or BOOTVIR.NTZ file. A memory resident virus will usually generate a VIRUSAM.PLE file and sometimes a VIR-CODE file too. The former is the result of a baiting process, the latter will result from the self-baiting of IV's modules.

Be aware that viruses can be extremely deceitful. Virus writers are constantly studying the anti-virus products and they develop techniques to evade detection methods. There is a continuously ongoing battle between virus writers and anti-virus developers. You will find that not all detection methods work on every virus, and you should not expect they do. InVircible's generic detection is built with a fail-safe methodology, assuming that the capturing of virus activity by a single sensor suffices to alert the user. With sophisticated viruses this is all that you may get - a single alert, by a single sensor, that something is wrong. An alert should prompt a virus aware user to further assess the situation.

You should always assume the probability of a multipartite infector, even if only boot infection was found, and you need to discard this possibility before proceeding to recovery. Negative results to launching IVTEST does not mean there isn't a file infector active.

A low risk approach would be to boot from the rescue diskette and run IVB, the integrity checker, from the floppy, on all the hard drive's files. The pattern will become immediately evident in case a file infector, or a multipartite was active on the disk.

It's also worth to explore the MBR and boot sector with RESQDISK (from the main menu) whether they have changed, by simply 'restoring' both sectors from backup and watching if anything changes in their presentation on the display. Be sure that you are using the right rescue disk before attempting the above, for the right computer, and that the rescue is up-to-date.

Finally, run the correlator, IVX, to see if there are more infected files than the ones reported by IVB. Select one of the modified files, pointed by IVB as the sample to compare to. The files found in high correlation with the sample and not contained in IVB's report, are the most likely to having brought the infection into the computer.

It may be worth running IVSCAN, in case the virus is contained in IVSCAN's database. If you wish, and have an up-to-date scanner other than IV's, then you may learn the name of the virus, provided it has one, and it is contained in the scanner's database.

At this point you got sufficient information to proceed on removing the virus.

Recovering an Infected Computer, Step by Step

Occasional infections of the boot block are usually recovered by IVINIT, during initialization, when run from autoexec. IVINIT's recovery is especially effective against stealth boot infectors. IVINIT will usually prompt to reboot the computer after disinfecting from a boot infector. Rebooting is necessary to remove the virus from memory too. When no more boot infection is reported after restarting the computer, you then know that the disinfection was successful.

It is good practice to check your floppies after a boot infector was removed from the hard drive, since write-enabled disks that went through the drive are now probably infected. The simplest would be to process floppies in bulk with FIXBOOT. The latter will place a clean boot sector on floppies, regardless of whether they are infected or not.

Not all boot infections are obvious. In case of a boot infection of the hard drive was found and IVINIT did not remove the virus: Boot from the rescue diskette and restore the MBR and boot sector from their image files by pressing F1 then F3, when in ResQdisk.

File infectors are usually removed by IVB, after booting from the rescue diskette. First, run IVB in the default mode (check only) on the affected drive and assess the extent of the infection. Watch that all "changed" files follow the same pattern, i.e. their size increased by about the same number of bytes.

Next, run the correlator, IVX, using one of the infected files as the reference sample. Establish the parameters for the best scan and watch the number of correlating files found. Usually, IVX will find more affected files than IVB. Save a copy of the infected sample on a floppy, you will need it later.

Next run IVB in the restore mode (/R). Select between file by file or automatic recovery ("recover all" option) depending on the pattern of he changes.

Finally, repeat the IVX correlation scan, this time by using the saved sample, from above. This time only the infected files that had no pre-infected signature will show up. You can use the "Rename" or "Erase" option to neutralize these files. The last step will provide a good idea of the source of infection of this particular computer, or file server. Renamed or erased files can be replaced from clean originals or backup. Use IVX, with the saved sample as reference, for the presence of the infector.

Infected samples can be sent by e-mail or on floppy to NetZ Computing, for analysis.

Analyzing Virus Characteristics

Full stealth viruses can be cleaned and removed from files by virus cooperative methods. Therefore, it is important to determine whether a virus is a full stealth one or only semi stealth.

A virus is considered "full stealth" if the infected files exhibit no changes when inspected while the virus active in memory. The following is how to test this. Preferably run the tests on a dedicated test machine, or on a floppy, with the hard disk set to "not installed" in the setup.

First, initialize new integrity signatures of the control group files with the IVB /S command. Next, infect the programs by whatever infection algorithm the virus uses (direct action, Piggybacking, execute program etc.). While the virus is still active in memory, check the files with IVB's default mode. Full stealth viruses will report nothing, while semi-stealth viruses will report "file was modified, no change in size".

Virus-Cooperative Integrity Recovery

Full stealth viruses, except cluster infectors, can be fully recovered through virus cooperative methods. The most straightforward one is integrity recovery. This method takes advantage of the virus properties itself. Before applying this technique to large number of files, test it first on a control group and if successful, apply it on the affected directories and drives. Always prefer operating on selected directories rather than on whole drives or volumes.

With the virus active in memory, secure the affected directory or drive with the command: IVB /S [pathname]. Activating the virus can be achieved by deliberately running an infected program.

When done, reboot from the rescue diskette or from clean DOS and restore all files with the command IVB /R [pathname]. Do not use the same copy of IVB.EXE that was used for securing, even if IVB indicated that it restored itself. Some viruses re-infect on 'closing' the file. Use a clean copy of IVB, from the rescue diskette, for example.

Test a few restored files by executing the programs, to assure the recovery was good.

The following are viruses that respond well to cooperative integrity restoration: All strains of Frodo (4096), Tremor, Natas, Die-Hard_2, Hemlock and more. Natas and Hemlock are multipartite and need cleaning of the MBR too. The MBR infection of Natas and Hemlock is stealth too and can be recovered with cooperative SeeThru.

Virus-Cooperative Piggybacking (Retro-Piggybacking)

Cluster infectors cannot be recovered with the above method because of their peculiar infection mechanism. It consists of inserting the virus code in the file by manipulating pointers in the directory entries or in the FAT. There are actually just a couple of such viruses, DIR_II and Necropolis (1963), with a few strains of each.

DIR_II is almost extinct as it cannot survive in a DOS 5 environment and higher.

Necropolis is in the wild and is still common in many places.

Many full stealth viruses, and cluster infectors in particular, can be disinfected by virus-cooperative or retro-piggybacking. The principle behind this method is the same as in integrity recovery, except that the disinfection is performed with the virus actually in memory. Cooperative piggybacking is embedded in IVSCAN and can be invoked from the command line only. Inverse piggybacking if used in the wrong circumstances can cause irreversible damage to infected files. Therefore this method is not available from IV.

Some full stealth viruses like Tremor, Natas and Frodo can be disinfected by either cooperative integrity or retro-piggybacking, others can be fixed only by one method or the other. The way to find out which method suits a particular virus is by trial and error. Before applying a method to a large number of files, always test first on a control group containing a limited number of infected files, or on a dedicated test machine.

Retro-piggybacking is effective against full stealth viruses only.

Retro-Piggybacking, Method 1

Retro-piggybacking, method 1 is activated by running: **IVSCAN /M1 [pathname]**

This method is effective against full stealth viruses of the file infector type and against Necropolis (1963 - a cluster infector).

Before activating RP method 1, turn off all disk caching as disk caches have a counter effect on retro-piggybacking. The simplest is to bypass config.sys and autoexec.bat by either renaming them, or pressing F5 while booting, if you run under DOS 6 or higher.

The virus must be active in memory when starting RP method 1. Apply method 1 selectively only to directories that contain infected files. When done, reboot the computer without executing any other command, since the virus is still active in memory and the end of the process.

Retro-piggybacking, method 1 is a single pass process. The files will be clean after rebooting. Method 1 is invoked automatically when running IVSCAN /R, and Necropolis (1963) is found active in memory.

Change Attributes, Method 2

Some viruses, DIR_II specifically, remove themselves from system files. Cleaning method 2 takes advantage of this property by changing the files' attributes with the virus active in memory, and restoring the attributes to original when rebooted clean. Method 2 is virus- -cooperative and will function only when the virus is active in memory. To invoke method 2 run: **IVSCAN /M2 [pathname]**

When done, reboot the computer immediately, as the virus is still active in memory. You will notice that all the executable files seem to have "disappeared". They are still there, but their attribute was changed to a hidden, and you can't see them with the DIR command, nor run them by invoking the program's name. To restore the attributes to original, run from a clean copy of IVSCAN: **IVSCAN /M3 [pathname]**

Cleaning method 2 is invoked automatically when running IVSCAN /R, and the DIR_II virus is found active in memory.

Virus-Cooperative (SeeThru) Boot Block Recovery

The master boot record (sector) of the hard disk is a favorite target for viruses. Those that use stealth are especially easy to recover from by cooperative generic methods. Stealth boot infectors usually hook BIOS interrupt 13h, the one used to read and write to absolute disk sectors, in order to conceal their own presence.

InVircible uses low level tunneling techniques that "sees through" virus stealthing of interrupt 13h. SeeThru is widely used in InVircible for the detection of boot infectors and their disinfection.

IVINIT will suggest the recovery of the MBR in case stealth infections of the MBR is detected. IVSCAN will prefer cooperative SeeThru for the recovery of a stealthed MBR and ResQdisk will permit the manual recovery of the a stealthed MBR, providing visual control of the process.

Cooperative SeeThru is safer than using the undocumented FDISK/MBR command. The latter will install a standard bootstrap program in the MBR, overwriting the current one, while retaining the existing partition data in the sector. While this procedure may help in many cases to remove an MBR infector, it may in others cases cause loss of access to the hard drive or cause it loose self booting capability. The FDISK/MBR undocumented command should not be used in case of infection by an overwriting MBR infector (Monkey for example), or when a special boot driver (DM 6.03+, EZ-Drive) is used, with access control systems, Boot Manager, OS/2 partitions, Windows NT or with other than FDISK partitions.

Instead, the method based on cooperative SeeThru will reinstate the original MBR. InVircible will capture the original MBR as returned by the virus stealth and will rewrite it to the physical location of the MBR, on the hard disk.

IV's SeeThru functions only with IDE and E-IDE drives. With other drive types like SCSI and MFM, use the technique described later.

Whenever a boot infector is suspected, then assume that all floppies that were used on that computer might be infected as well. After disinfecting the hard drive, process all floppies with FIXBOOT, the complimentary utility provided with InVircible. FixBoot will install a standard DOS 5 boot sector to the floppies.

Removing Stealth Boot Viruses from SCSI and MFM

The problem with tunneling under DOS and BIOS to find INT 13h's original handler is that there is always a way to defeat it, no matter what method is used. DOS and BIOS interrupts are virus realm. Viruses cannot thwart low level hardware tunneling, yet IV's SeeThru does not work on all drive types. When SeeThru is unavailable, then a message "SeeThru OFF" will pop up when running IVINIT, IVSCAN or IVTEST. This should not concern you unless something else happens.

Boot infectors, except few, disclose their presence by memory stealing. They will also usually relocate the original MBR or boot sector to elsewhere on track zero. Others will patch the MBR without storing a copy anywhere, but will write virus code elsewhere on track zero. Boot infectors that use stealth conceal just the faked MBR itself. These properties can be used for detecting stealth virus doing on the hard disk. It's especially useful on drive where SeeThru is not available.

When installing InVircible to the hard disk, an image of track zero is stored to disk root directory, in a hidden file. Non-stealth boot infectors are detected by IVINIT when executing the autoexec. The presence of a stealth boot infector will usually be indicated by memory stealing, Qemm will not load, or Windows' 32 bit access driver will not load. Each one of the above symptoms should alert the user of the possibility that a stealth boot virus infected the disk.

Run ResQdisk and visually inspect track zero for the traces of unusual code that wasn't there before. You may wish to compare the current content of track zero to the one at the time you installed IV. Run ResQdisk an press **^Z**. When you select compare, then ResQdisk will scan through track zero and stop on every sector that was changed from the time the backup was made.

Non-stealth viruses can be removed the generic way. The best is to find the original but relocated MBR on track zero and to rewrite it to sector 1. Before proceeding verify with the 'analyze' (**^A**) function that the sector contains correct data that fits the hard drive. The number of partitions should fit and the total number of sectors, multiplied by 512, should approximately equal the partition's capacity.

Pick the selected sector with ResQdisk's 'cut and paste' (**^E**) and write it back to sector 0,0,1. When no copy of the original MBR or boot sector can be found, and there is no rescue diskette either, then you may have no other choice except refreshing the bootstrap program (this is the one that gets infected) with a generic one. Both sectors can be refreshed by positioning ResQdisk on the desired sector and pressing F1-F4 in sequence. The DOS boot sector can be refreshed from ResQdisk's **^B** menu and selecting '**SYS**'.

To remove a stealth boot virus, boot clean from the rescue diskette, run ResQdisk (the rescue floppy default) and restore the MBR by pressing "R".

In case there is no rescue diskette and the computer is infected by a stealth boot infector then format a floppy in drive A:, transfer the system files to the floppy with the command SYS A:, next copy the RESQDISK.EXE module to the floppy, log onto the floppy and run RESQDISK/B, to create backup images of the boot and MBR sectors on the floppy. Since the virus is stealth, then the image files are those of the clean sectors. Finally, run FIXBOOT A: and remove the floppy from drive A without running any other command. If you have access to a clean computer then AFTER having prepared the boot images on the floppy, run SYS A: on the clean machine, instead of running FIXBOOT.

Boot from the clean floppy and run RESQDISK /R for automatic recovery, or you may recover the sectors one by one under the visual inspection of ResQdisk.

The following are exceptions when NOT use the generic F1-F4 MBR refresh method.

Do not use it in case access control password software is installed to the hard drive, that use encryption of the MBR. Neither use this method when the partition data was overwritten by a virus. A rule of thumb is this: Boot the disk from the rescue diskette or from clean DOS. If the hard drive is inaccessible, then don't use F1-F4. Note that you should not use FDISK/MBR either in these conditions, as their function is the same.

Removing Boot Viruses from Floppies

Unlike the disinfection of programs, floppies' boot sectors can be disinfected easily without knowing what virus infected them. IVSCAN can be used to inspect floppies and will alert on the presence of both known and generic boot infectors. If found, IVSCAN will announce "boot virus [name] was detected". Not in all cases there will be a name, depending on whether the virus is contained in IVSCAN's database. Since InVircible prefers generic methods, then having a virus name is the exception with IV. If interested in the virus name, then you may want to scan it with your favorite scanner.

Cleaning floppies from boot viruses can be done by IVSCAN, one by one, or in bulk processing by FIXBOOT. When a boot infection of a hard disk is found, then it's most likely that floppies are infected too in the workplace. First disinfect the hard drive as inspecting from an infected computer will infects the floppies processed. IVSCAN /R [drive] will attempt to find the original boot sector and remove known and generic boot viruses. IVSCAN /B will force a boot sector replacement, either by finding the original sector or by installing a standard one. Both switches, /R and /B have a limitation: They are unable to clean a floppy having a closed loop infection. This situation occurs when both the boot sector and the relocated 'original' boot sector are infected. IVSCAN will then announce on an 'indefinite loop infection' and abort.

This is where FIXBOOT can fix the problem. FixBoot installs a standard boot sector to floppies of the following capacity: 360 kb, 720 kb, 1.2 meg, 1.44 meg or 2.88 meg. FixBoot will sense the floppy's capacity and select the right boot sector for it. It will also retain the floppy's bootability on condition that it's a DOS floppy, from version 4 and higher and from either the MS-DOS or PC-DOS flavor. FixBoot is useful to restore readability to floppies that became inaccessible because of a boot infection. A 1.44 floppy that was infected by B1-NYB will seem as unformatted when read in a clean, uninfected machine. FIXBOOT /S will prompt you to select a capacity for the floppy and fix that problem. When replacing an infected boot sector, FixBoot will announce that it replaced an infected boot sector. And last, FixBoot will prompt if to process another floppy, saving a lot of time in bulk processing of large number of floppies.

The files stored on floppies processed by FixBoot will not be altered or affected by FixBoot in any way.

Regaining Access to a Hard Drive

Loosing access to data on a hard drive is a common and daily eventuality with personal computing. If not caused by hardware failure, then InVircible can help in regaining access to the hard drive. Viruses are the least common cause for loosing access to a hard drive. The most common is the wiping out of the CMOS.

First thing to do after installing InVircible is to prepare a ResQdiskette. We assume that you also make regular backups of your important data. In the unfortunate event of losing

access to the hard drive, then first thing find your IV rescue diskette.

Before proceeding, make sure that the rescue disk is up-to-date and reflects the current configuration of your computer. Boot from the rescue diskette (you may need to setup the drive A parameters in case the CMOS was wiped clean) and press Enter. IV will start in the ResQdisk default mode. In case the problem is the CMOS, then ResQdisk will indicate that the hard drive setup has changed. Press **^F3** to restore the CMOS data from backup. Reboot the computer and resume operation.

A damaged MBR or boot sector can be restored from backup either individually, by selecting the sector and pressing (**R**)estore, or automatically with **F1-F3**. Do not use the **^F1** and **^F2** (forced recovery of the MBR and boot sector, respectively) unless you are sure that the rescue floppy is up-to-date and contains valid and current data of the computer you are using it on. Always recover the MBR first, as a messed up MBR may point to the wrong location for the boot sector. You may find after recovering the MBR that the boot sector is just fine, but was inaccessible because of a bad MBR.

Regaining access to a hard drive without having an IV rescue diskette is also possible, yet it requires a licensed copy of InVircible, a bootable DOS floppy and a few DOS programs such as FDISK, CHKDSK and UNFORMAT.

Start by verifying the CMOS setup data and look for hardware problems. Set the hard disk parameters in the setup to its correct type, if you know what it is, or select any type that will let the computer boot. Even if wrong, the disk type can be corrected later, just get the machine started.

The IV rescue diskette procedure creates a text file named HD_DATA.NTZ which is written to the floppy. This file contains the CMOS setup, as well as the IDE parameters of the two first installed hard drives. If such rescue diskette exists then you can read the parameters written to the HD_DATA file and restore them in the CMOS in case of need.

Boot from a floppy, start ResQdisk and watch for its messages. Press F5 to read the setup and the manufacturer's IDE parameters. If the data makes sense in both columns, then the problem is not a hardware one. If on the other hand the IDE data is erratic (provided the hard drive is an IDE or E-IDE) then there might be a hardware problem. Switch then the computer off, disconnect the power cord, open the computer, check the power connection to the hard disk, the flat cables from the controller to the hard disk and the controller card's connector to the motherboard. Too often it's just a matter of a loose contact. Close the computer, connect to power and try again.

The 'setup calculator' panel of ResQdisk (accessed through F5) should show sensible data. The right column should contain non-zero data for IDE or E-IDE drives. If it's empty or shows 'not an IDE', then try reducing speed (switch turbo off, or press **Alt+Ctrl+minus**) and restart ResQdisk. If the IDE data is now visible then you are having hardware mismatch between the hard disk and the IDE controller. The controller is usually the problem, rather than the motherboard or the hard disk. Try changing the hard disk mode in the setup, this might solve the problem. Common setup problems are: LBA is used with small capacity and older drives (don't use LBA if not needed!), mode 3 or 32 bit mode is used with hardware that does not support these modes. It's also possible that the hard drive is simply too slow (and perhaps too old) for the new Pentium/120 MHz just acquired.

Use the ResQdisk setup utility (F5) for finding the original disk's configuration parameters. The parameters in the setup and those of IDE disk manufacturer may differ. However, they should combine to the same disk capacity. An IDE drive that was configured and formatted with setup parameters that differ from the manufacturer's will function improperly if those parameters are not restored to the same as when configured. Fortunately, these parameters can be found with a little detective work and the help of ResQdisk.

Look with ResQdisk for the boot sector. It should normally be accessed with the Left arrow key. If not there then you may find it elsewhere on track zero. Suppose you found the DOS boot record in sector 0,0,52. Press **^A** to check its content and note the number of heads and the hidden sectors. The latter is usually equal to the number of sectors per track. In this case the number of sectors per track should be 51.

Now go to the setup utility (F5), set the number of heads and sectors per track to those found in the boor sector, and change the number of cylinders until the number in the "capacity" field equals that of the manufacturer's. Write down the parameters, exit ResQdisk and these parameters in the setup. Reboot, and the disk should be accessible.

A different situation may exist when the MBR or the DOS boot sector were destroyed or their data was modified. ResQdisk can restore access to such drive when either of the two sectors exists, but not if both were destroyed. In such case you will need the ResQpro version. Another limitation of ResQdisk is that it can recover only the first partition of a physical hard drive. If the disk was partitioned with more than one logical partition, then you'll need ResQpro. InVircible Pro, with ResQpro is available from InVircible agents.

ResQdisk will prompt to press **^F1** or **^F2** to reconstruct the MBR or the DOS boot sector, respectively. After reconstructing either of the above you need to reboot the computer in order of the renewed sector to become effective.

Zeroing the partition data. ResQdisk may 'refuse' to function if the data in the partition makes no sense or if the MBR signature (the 55h AAh bytes at the sector's end) does not exist. It may be necessary to zero the partition data in the MBR and reboot in order to let ResQdisk function. A special key combination is provided for this purpose in ResQdisk, **Alt+0** (Alt+zero). Using FDISK/MBR will do the same in such conditions. Use sparingly and after making a backup of the existing MBR!

To reconstruct both sectors, or to recover a higher partition you will need either a ResQpro kit, or to contact an IV agent, to authorize a single ResQpro session. When authorized, then press **^F4** to start searching for partitions on the hard disk. ResQpro will only handle DOS partition that were prepared with FDISK. It's important to understand the way FDISK prepares more than two partitions on a single physical disk. Each partition sector has provisions for up to four sub partitions. Yet the MBR contains usually only data for two partitions only: The primary DOS partition and an extended DOS partition, occupying the space left after allocating the primary DOS partition.

Higher partitions are the subdivision of the extended DOS partition. The extended DOS partitions are usually written into the second partition's partition sector. ResQpro

reconstructs the MBR only. Therefore, only two partitions need to be rewritten to the MBR. ResQpro will prompt and ask to confirm for every partition found while searching the disk. Only the first one should be confirmed. It is worth writing down the coordinates of all the partitions found, as it may be helpful if manual disk editing will be needed, later. ResQpro will report its finding and prompt before writing the data to the MBR.

IV agents may authorize a single ResQpro session (see the Agents List). For ResQpro's password, run ResQdisk and press **^F10**. Proceed as instructed by InVircible's hotline support.

For the recovery of large capacity IDE drives, using Disk Manager's dynamic boot driver, look in the appropriate section. See also ResQdisk and ResQpro.

Handling Cluster Infectors

There are two known cluster infectors that appeared by the end of 1991, Necropolis (1963) and DIR_II. Both originate from Bulgaria.

DIR_II is almost of no interest, as it does not survive in DOS 5 and higher. When not active in memory, CHKDSK will indicate that all infected files are crosslinked on the same cluster, usually the highest free one in the partition. To clean from DIR_II run IVSCAN as described above. To clean large partitions (more than 32 meg) with IVSCAN you will need to boot from DOS 4.01. For smaller partitions you may boot from DOS 3.30.

CHKDSK will show allocation errors on files infected by Necropolis (1963), when the virus is not active in memory. 1963 is removed with IVSCAN by retro-piggybacking. Do not run retro-piggybacking on a drive infected with Necropolis, unless you are sure the virus is active in memory. 1963's presence in memory is clearly indicated by the response of IV's self integrity check.

In case a cluster infector is suspected, then do not use the CHKDSK /F switch until the disk is completely disinfected! Using the /F switch will ruin the infected files beyond repair.

The Screening of New Software

The purpose of screening is to detect infected software at the earliest possible stage, and to prevent it from infecting other programs or hard drives. Screening consists of two stages:

Preliminary Screening by Scanning

Scan all new software with an up-to-date third party virus scanner and with IVSCAN. Preliminary screening need to be run just once on all new software and distribution floppies.

Active Screening

Active screening is performed under the surveillance of InVircible to capture new viruses that passed the preliminary screening process. Any virus that is unknown to the scanner may be considered as "new", at least to the scanner. As new, or virtually new viruses are caught at the active screening phase, then the updating rate of the scanner may be relaxed, compared to where InVircible is not used.

The IV Quarantine Method

The ideal screening machine is a stand alone PC (not connected to a network) with a hard disk running under DOS, and with functional applications installed on the hard disk. IV should be installed, all programs should be secured by IVB and an IV rescue diskette for that PC should exist.

Install the software to be tested to the screening machine, as instructed by its producer and secure the new files by running IVB.

Exert all the functional and executable modules of the software under test, then exit back to DOS. Run IVINIT, then IVB, on the whole drive content. Reboot and watch IV's tests for exception messages.

Reboot clean from the rescue diskette, run a ResQdisk Compare track zero test to check for changes, run IVB on the hard drive to seek for changes and if nothing is found in any of the tests, then the software may be declared clean from virus.

Software screening is especially important in the corporate and institutional environment where infected software could be distributed throughout the organization. Boot viruses are more prevalent, but their damage is limited to a local hard drive and they cannot propagate through the network. File viruses are far less prevalent, only 30% of the reported incidents, but their damage can be devastating as they may affect many files on hard drives and on file servers and they propagate across the network with the server acting as a virus mailbox between workstations.

Handling Commonly Encountered Problems

Copy
Copy

The following are the most common alerts that IV may provide, and what to do in case they happen.

Memory stealing: Normally attributed to boot infection. Watch for other messages while running IVINIT and IVB. If no other message appears, then run ResQdisk. Inspect track zero (**^Z**) in the Compare mode. Repeat the ResQdisk inspection from the rescue diskette, in case it's a stealth boot infector and the drive isn't an IDE.

Boot infection: A boot infection will usually follow a memory stealing message. In the case of a stealth boot virus the "faked" sector warning will follow. The remedy to these viruses is in IVINIT, IVSCAN, ResQdisk and in the rescue diskette.

Faked partition (or boot sector): Indicates the detection of a stealth boot infector on an IDE hard drive. Partition sector (MBR) infections are also considered as "boot". The remedy is SeeThru recovery with IVINIT, IVSCAN or ResQdisk.

When a boot infection of the hard drive is detected, then assume that there floppies in the workplace could be infected too. After cleaning the hard drive, process floppies with FixBoot to remove possible boot viruses.

Virus active in memory: A memory resident virus may disclose its presence in one or more ways. It may trigger the self baiting of the IV module, it can infect special baits launched by IV, and it the later case, the virus will be sampled into a sample file named VIR-CODE, VIRUSAM.PLE or in both. The sample files are used by the correlator scanner, IVX.

Piggybacking detected: Indicates the activity of a fast infector or a file killer (a virus that erase files on selective criteria). Piggybackers infect host files while they are accessed such as with antivirus scanners for inspection. The detection is based on actual infection occurring on a few files. Known fast infectors can be removed in the screening process, while unknown and new piggybackers are detected by IV, before they caused any significant damage. The few programs that were 'sacrificed' in the detection of the process can almost always be recovered by IVB, or replaced from backup.

Important! The detection of piggybacking is based on sensing spurious background writing to disk, while an IVB, IVX or IVSCAN search is in progress. Multi-tasking and peer-to-peer disk sharing may cause **false piggybacking alarms**. Before running any of the above, make sure that there are no other tasks running in the background, like downloading from the Internet, or the disk being shared as read-write by a remote user

Because of the risk of piggybacking, never run a scanner, not even IV's on a remote drive or a file server without assuring first that there is no piggybacking. Run an IV scanner on a write enabled local hard disk to discard such possibility.

Full stealth fast infectors can be removed by virus cooperative methods such as retro-piggybacking or cooperative integrity recovery.

Virus was sampled to file: IV has a dedicated viral probe and sampler named IVTEST. IVTEST can run on demand to sample an active virus into files. IVTEST will also produce a PARTVIR or BOOTVIR sample when a boot infector is found. The type of sample, their size, and the messages issued by IVTEST are very instructive in assessing the characteristics of the virus and should be reported to support personnel.

File changed: The nature of the changes is always indicated by IVB. It can be an increase, or decrease in file size. An inconsistent change pattern is most probably a new version of a program. IVB will then prompt for renewing the signatures database. A single modified file or several modified ones, with an inconsistent change pattern are not a virus but something else. In such case check for file corruption with CHKDSK. Only a consistent change pattern in several files indicate a virus doing. The remedy is to restore the files to their preinfected state with IVB, after booting clean from the rescue diskette.

More Common Problems

Copy
Copy

Most users are unaware of the potential problems caused by DOS's undelete tracker. UNDELETE has a memory resident mode when loaded with the **/LOAD** or **/S** (sentry) switch. The undelete tracker will then create a hidden directory with the name \SENTRY, often confused with IV's Sentry mode. All the deleted files are moved to the \SENTRY directory, until purged. If not really needed, then turn the undelete tracker OFF.

InVircible uses unique low-level hardware access methods to dig out and remove tough and elusive boot viruses. Though, these very same methods sometimes uncover hidden flaws in hardware and incompatibilities. The most common are problems with an IDE drive, special controllers, and sometimes with special drivers used in the config.sys.

To find whether a problem is caused by a conflict of IV with hardware or a conflict with other software, then start DOS and bypass the autoexec and config by pressing F5 (DOS 6+ users) and run the IV module that malfunctioned in the clean DOS environment. If it works clean, then search for the cause in a drivers or TSR which is loaded before IV. Use the F8 key while starting DOS to step through the drivers, one by one.

Use elimination until the driver, or TSR in conflict is found. You may then change the order in which TSR are loaded not to conflict with IV. In certain cases you may find that a driver or TSR that conflicts with IV, causes problems elsewhere too. In many cases you may give up the driver and improve performance. If the driver is absolutely necessary, then you may have to give up IV.

Anti-Virus **scanners** and **behavior blocker TSR** are a constant source of conflicts with InVircible. Scanner TSR with **background checksumming**, such as **VSAFE** and **Norton AV** (*NAV*) will trigger IV's Piggybacking sensor. There isn't real virus piggybacking, but there is one of the checksummer. Background checksumming slows down disk access and you may wish to turn this option off.

Behavior blockers such as **Thunderbyte's** *TBfile* and *TBcheck* intercept IV's self baiting and the bait launching processes and may crash the computer. If you need a memory resident behavior blocker then you should load it after the IV tests were completed. You should also remember not to run an IV module once the behavior blocker is loaded to memory.

As mentioned elsewhere, IVB, the integrity analyzer will usually be the first to notice file structure problems such as crosslinked files and FAT allocation errors. In case inconsistent changes in files is reported, then check with **CHKDSK** and act accordingly.

Programs Injection and Inoculation

CPAV is now a discontinued anti-virus product, yet there are still many copies around. CPAV had an option that would let a user to 'immunize' or inoculate executable programs with a protective shell. InVircible detects the CPAV inoculation in files and alerts the user of their presence. From other anti-virus products F-Prot does the same and there may be others as well.

This file inoculation is the cause of problems. Many programs do not tolerate being inoculated, especially not DOS files, but not only them.

The inoculation process consists of the encapsulation of executable files (COM and EXE) in a protective shell, like a cocoon. The inoculation adds about 700 or 900 bytes to COM and EXE files, respectively. The inoculation also modifies the programs in other ways.

The inoculation or injection of executables has too many drawbacks and practically little or no merits at all. First, the 700 to 900 bytes waste valuable disk space. Secondly, the CPAV inoculation causes some programs to malfunction. Thirdly, the recovery of an immunized file will in most cases worsen the situation. In order to drop the virus the infected file should run, at the risk of loading the virus into memory and infecting other files. And lastly, inoculation is ineffective against smart viruses like full stealth ones. InVircible's files, when inoculated by CPAV will be ruined.

Except for its malfunctions, there are good reasons why not to inoculate programs. The inoculation may hide a virus that infected the file prior to its inoculation. A dedicated virus may simply peel off the inoculation, infect the file and then re inoculate it as if nothing happened. Since InVircible can and does peel off the CPAV inoculation then there is no reason why a virus couldn't.

InVircible will indicate files immunized by CPAV by either scanning them with IVSCAN or IVB. IVSCAN can remove CPAV's immunization in the "Remove" mode (IVSCAN /R).

Unfortunately there are differences between versions of CPAV, so it is safer to disimmunize files by the same product that inoculated them in the first place, CPAV.

Use the above procedure only if there is no other alternative, and with all precautions. Backup the files to be recovered. IVSCAN will remove the CPAV inoculation from files when used with the /R switch.

Except the CPAV inoculation, there are other antiviral products that inject a protective shell to executable files. One is INJECT.EXE from ARF Software. There aren't any known incompatibilities between the latter product and InVircible and the two coexist without having apparent problems. The IV programs will restore themselves and drop the INJECT shell, if inoculated. INJECT may be used to demonstrate InVircible's self restoration. The latter demo is harmless, to both IV and INJECT.

Memory Resident Anti-Virus (TSR)

Copy
Copy

Two type of TSR are used in various anti-virus packages: TSR scanners - the majority, and virus activity blockers, or behavior blockers.

Anti-Virus TSR affect the computer's performance and its normal operation. Some TSR, especially behavior blockers, interfere with InVircible's operation and with other applications as well.

Among the adverse effects of anti-virus TSR are: They slow down the computer operation, especially disk access, they waste precious computer resources such as memory and CPU time, they tend to conflict with applications and they constantly risk crashing memory and hang the computer's operation.

As for the TSR effectively, it can only handle viruses contained in its database and thus needs frequent updating. An anti-virus TSR scanner database handles usually far less viruses than the on-demand scanner from the same producer. More viruses can be found simply by screening new software with the scanner than with the TSR. Therefore, it's more logical to screen new software once with a good scanner, than to constantly suffer the oppressiveness and inconvenience of the TSR.

TSR scanners and TSR behavior blockers should be loaded after InVircible completed its tests in the autoexec, to avoid them from interfering with IV.

Practicing Antivirus, the AV Practice Lab (AVPL)

NetZ Computing provides a practicing environment named the **Anti Virus Practice Lab (AVPL)**. AVPL is complementary to antivirus software and was developed for the purpose of practicing anti virus techniques in general and InVircible's in particular.

It is best used in concert with InVircible. Yet it can be used with any AV product to learn about viruses and how to protect your PC and network against them.

AVPL is a self contained tool that will let you practice several virus attack scenarios and how to recover from them. Although AVPL is designed to be used with the advanced generic anti-virus tools of InVircible, it can also be used to evaluate AV products in general.

AVPL allows to "infect" selected computer programs in a virus like way, to install master boot sector (MBR) infectors to the hard disk, and to practice InVircible to identify the simulated viruses and recover from them. The "infections" produced by AVPL are ultimately harmless. The viruses created by AVPL will not replicate. They can not spread through a process of spontaneous replication to other programs or disks. They will only affect the program files in a target directory selected by the user and the hard drive of the system on which AVPL is installed. Therefore, the user can be assured that only those programs and the hard disk of their system can be affected by AVPL. AVPL will ONLY affect program files and the hard drive chosen by the user.

Programs that are affected by AVPL can be "cleaned" either through the use of the Uninstall option of AVPL, through InVircible or through the DOS delete command.

MBR infections that were installed by AVPL can be removed as well by AVPL's Uninstall option, or through the use of the ResQdiskette. It is important that you prepare a Rescue Diskette before you begin experimenting with AVPL.

Prior to experimenting with AVPL please read its online hypertext guide. AVPL is available from IV distributors, anonymous ftp sites, AOL and CompuServe. AVPL is freeware, the file name AVPL.ZIP.

For users with knowledge about viruses: Rehearse with real viruses on an isolated (not in a network) computer to get the feel of what a virus attack is like. Experiment only with controllable and 'well behaved' viruses such as Stoned, Jerusalem and Frodo (4096). Avoid Trojans, and deliberately destructive viruses, there are rare and there is little to learn, if anything at all, from sheer destruction.

And last, prepare a recovery plan and simulate it on a small scale, prior to being hit by a real computer virus. When the real thing happens, you will feel much more at ease if you have an idea of what a real virus attack is like and have practiced before.

The NetZ Utilities

From time to time NetZ releases useful utilities to assist handling virus related problems and disaster recovery. A set of such utilities consists of **FixBoot** as described here, **SwapBoot**, used to swap drive A and B and let boot from B without requiring to open the computer cover for swapping the cables, and **XMonkey**, a program that will remove the Monkey MBR infector from up to eight installed hard drives. The three utilities are freeware and are available from BBS and the world networks.

Dedicated Purpose Packages: From time to time NetZ releases dedicated purpose packages containing one or two modules, with ad-hoc documentation. Such were **RESQDISK** - a dedicated disaster recovery package, **IVX** - dedicated signature extractor, signature and correlation scanner, and **XCAIBUA** - a dedicated virus scanner derived from IVSCAN, to constrict the outburst of the new **Big Caibua** virus.

Upgrading and Support

InVircible does not require frequent updates, thanks to its generic features. IV let recover from both known and unknown viral infections. Still, IVSCAN is seldom updated on a need-to basis, to handle just the most common fast spreading viruses.

InVircible is undergoing a continuous improvement process through upgrading. As computer virus technology evolves, its counter techniques should evolve too, to cope with the new techniques. Such are SeeThru, introduced in 1993, the IV Armor and the new hyper-correlator, both introduced in 1994.

In the ongoing process of developing InVircible, NetZ constantly adds new features and replace older ones. Thus, IV is continuously upgraded, while IVSCAN is just updated on an irregular basis.

Thanks to its generic nature, a version of InVircible is usable for at least a year, or two, without really needing to either upgrade or update. At the most, IVSCAN will not identify a virus, but if properly used on regular basis - i.e. routine checks by IVINIT and daily inspection by IVB - then such viruses will be detected and can be removed too, without knowing their name.

Upgrading your Licensed Copy of InVircible: Licensed copies of InVircible are upgraded by overwriting the old files with the new ones, on both the distribution floppy and on the hard disk. You may format the distribution floppy with DOS's **FORMAT** without voiding its license. The same applies to the hard disk. The disk can be formatted (high level FORMAT), or be reconfigured with FDISK, without affecting the license. You may also install/uninstall disk compression software without needing to retract your license. Only Low Level Formatting, or unrecoverable hardware failure will cause loss of the IV license. This is why IV is provided with one spare license on the distribution floppy.

InVircible's upgrades are available through BBS, the Internet, as well as from the major networks. IV's latest upgrades can be downloaded through anonymous ftp from the author's site at **ftp.netzcomp.com**, **CompuServe** (Go *InVircible*), **America On Line** and from the vendors' Websites. A list of vendor's websites are:

http://www.comsecltd.com
http://www.invircible.com
http://www.newcastleintl.com
http://www.nha.com
http://www.second-sight.co.uk
http://www.second-sight.co.nz/ssl

Timed IV Releases: From time to time NetZ releases a fully functional, yet timed version of InVircible. These releases can serve for the full evaluation of InVircible. The expiration date of the timed license is indicated in a file named EXPIRY.TXT. Past that date, the IV copy will revert to *Sentry* mode. Registered users should download the

upgrade package instead.

Online or E-mail Support: Questions can be sent through e-mail to **support@invircible.com**, or addressed to the SysOp on the CompuServe forum (go *InVircible*). Questions to the author should be addressed to **netz@invircible.com**.

Agents List

Copy
Copy

Australia

Richard Macmillan
Second Sight Australia Pte, Melbourne, Australia
Tel: (03) 682 4088
Fax: (03) 682 4089
E-mail: SSL@melbourne.dialix.oz.au

Steven Wilson
Second Sight Sydney Pty Ltd, Australia
Tel: (02) 557 8102
Fax: (02) 519 8774
E-mail: SSAL@sydney.dialix.oz.au

Belgium and France

Dan Oren
Link Systems, Brussels, Belgium
Tel. +32 2 767 6230
Fax. +32 2 767 1569
Compuserve: 100347,2276 Compuserve,

Canada

Chuck Corradi
New Castle International, NH
Tel. 800-803-8700
603-431-6170
Fax: 603-431-6370
email: ncichuck@newcastleintl.com
WWW: http://www.newcastleintl.com

Germany, Austria, Switzerland, Luxemburg

Dan Oren
Link Systems, Brussels, Belgium
Tel. +32 2 767 6230
Fax. +32 2 767 1569
Compuserve: 100347,2276 Compuserve,

New Zealand

Grant Scurrah
Second Sight Limited, Auckland, New Zealand
Tel. +64 9 3661593
Fax. +64 9 3661594
E-mail: SSL@iconz.co.nz
WWW: http://www.second-sight.co.nz/ssl

United Kingdom

Richard Macmillan
Second Sight UK Limited
Tel. +44 1865 821 811
Fax. +44 1865 821 571
E-mail: ssl@easynet.co.uk
WWW: http://www.second-sight.uk

United States

Chuck Corradi
New Castle International, NH
Tel. 800-803-8700
603-431-6170
Fax: 603-431-6370
email: ncichuck@newcastleintl.com
WWW: http://www.newcastleintl.com

Michael Paris
Vine Computer Inc., IL
Sales. 1-800-422-5130
Fax. 1-708-863-1917
Support. 1-708-863-1464 (Software Problems)
email: vine@invircible.com
sales@invircible.com
support@invircible.com
WWW: http://invircible.com

Robert C. Casas, Ph.D.
COMSEC Ltd.
1545 Waukegan Rd. Ste. 2
Glenview, IL, 60025-2166
Sales: 800-754-8214
Support: 847-729-3565
Fax: 847-729-3575
CompuServe: 75162,241
email: rc.casas@ix.netcom.com
rcc@comsecltd.com
WWW: http://www.comsec.com

Norman Hirsch
NH&A
577 Isham St. # 2-B
New York, NY 10034
Phone: 212-304-9660
Fax: 212-304-9759
BBS: 212-304-9759,,,,,,,3
CompuServe: 72115,661
E-mail: nhirsch@nha.com
WWW: http://www.nha.com

Ehud Shany
Winsoft Corp.

Santa Ana, CA 92705
Phone: 714-833-8838
Fax: 714-833-8983
E-mail: winsoft@ix.netcom.com
CompuServe: 74534,2561

Singapore, Indonesia, Malaysia

Alex Neo
Infoware International Pte Ltd. Singapore
Tel: +65 483 3688
Fax: +65 483 0488
E-mail: infoware@singnet.com.sg

Compuserve, The InVircible Forum

SysOp: Robert C. Casas
Support: GO INVIRCIBLE
Registration: GO SWREG, #4999

America On Line

Sysop: Chuck Corradi
IVatNCI@AOL.com

NetZ Computing Ltd, Israel

Zvi Netiv (Author and Producer of InVircible)
Voice: +972 3 532 4563
+972 52 494 017
Fax: +972 3 532 5325
E-mail: netz@actcom.co.il
netz@netzcomp.com
CompuServe: 76702,3423
Author's ftp: ftp.netzcomp.com/private/netz/
Compuserve: go INVIRCIBLE

EIDE Drives and Dynamic Drive Overlay (DDO)

There are a few facts that you should know when using a large capacity EIDE drive with a Dynamic Drive Overlay (DDO).

The enhanced IDE drives are designed to work with extended BIOS definitions that support larger than 1024 cylinder drives. The 80x86 BIOS was limited by its designers to handle geometries of up to 64 sectors, 256 heads and 1024 cylinders, altogether 8.38 gigabytes, in sectors of 512 bytes.

The newer BIOS have LBA translation (logical block access), or a translation which accept higher than 1024 cylinders in the setup.

To overcome the BIOS limitation of older boards, the new EIDE are provided with a dynamic boot driver (DDO), usually Disk Manager 6.x or 7.x from Ontrack. Without the DDO, you will be able to access only 528 megabytes with the older boards and BIOS. The DDO enables accessing the full capacity even with an AT 286 BIOS.

A DDO replaces the standard MBR and writes itself to track zero of the hard drive. When booting, the MBR bootstrap reads the driver from track 0 into memory, and then continues booting the OS. Without the DDO loaded and running the hard drive is not recognized at all. You may configure the hard drive with just FDISK, but then the drive's capacity will shrink to only 528 mbytes.

If the hard drive was configured with a DDO then you won't be able to access the hard drive when booting from a plain DOS floppy. The DDO stealths the content of track zero to protect the special MBR and the DDO from being overwritten accidentally. Yet the stealth self protection is active only when booted with the driver in memory.

There are several good reasons why to avoid the use of the dynamic overlay driver.

The hard drive cannot be accessed when booting from a plain DOS floppy,

The special MBR and the driver are extremely vulnerable. They are overwritten by simple boot infectors, or even by attempting to remove such virus with the **FDISK/MBR** command.

The stealth self protection is disabled by Windows' 32 bit disk access. In result, buggy software, or just the unsuccessful installation of new software may overwrite the special MBR and kill access to the hard drive.

Disaster recovery utilities do not exist (except in InVircible's ResQdisk) to recover the special driver in case of damage. Standard rescue utilities like Symantec's RESCUE from the Norton Utilities misinterpret the stealthed track zero and will cause substantial damage if used on a drive with DDO.

The following are **DO** and **DON'T DO** if you use one of the special boot drivers:

Copy Do NOT install MBR immunization to a drive that uses DDO for booting.

Copy Do NOT install access control or security software that writes anything to the MBR, to the boot sector, or to track 0. You will be locked out of your drive, for good!

Copy Common disaster recovery (rescue) utilities are unaware of the DDO. As track 0 is stealthed then the rescue utility will backup the wrong data. A rescue floppy prepared by such utility is worse than not having one at all, since it will 'restore' the wrong information to the drive, for sure.

Copy Do not forget floppies in your drive A. Such floppy, if infected with quite simple boot/MBR infectors may cause loss of self booting capability, or the total loss of access to the hard drive!

Copy If you have two IDE hard drives, one with DDO and the other without, and the BIOS does not support LBA then install the smaller IDE as master and the EIDE with DDO as slave. You will need to install the DDO drivers to your config.sys (use DMCFIG.EXE for the purpose, on the Disk Manager floppy). This configuration is quite safe against most boot infectors.

Copy Install InVircible and prepare its rescue diskette. IV's rescue is DDO aware and is the only rescue utility that supports the special drivers thanks to its SeeThru capability.

The following is how to recover access to a large capacity IDE, if self booting or access were lost:

Copy **Damage to the system files:** Insert a bootable floppy in drive A, with the same version and OS as installed on the hd. Start the computer and press the **space bar** when prompted to boot from a floppy. When DOS is up, issue the command **SYS C:**

Copy **In case the MBR or one of the driver's sectors was overwritten by a virus.** If you prepared an InVircible rescue diskette, then boot from the floppy and it will automatically enter the ResQdisk program. Press **^Z** and select "**Restore**" of track 0.

If you didn't prepare an IV rescue disk, then first prepare a boot floppy with the FORMAT drive: /S command. Copy the drivers to the floppy (in Disk Manager case copy DMDRVR.BIN and XBIOS.OVL from the Disk Manager distribution diskette and edit A:CONFIG.SYS. Write a line containing **DEVICE = DMDRVR.BIN** in the config.

Boot from the floppy just prepared and see if you can access the drive.

The Disk Manager master boot record (MBR) and dynamic driver overlay can be repaired with the DM.EXE program. DM.EXE can be found on the diskette that you received with the hard drive.

Copy **With Disk Manager version 6.03:** Start the program in the manual mode by either running DM /M or pressing **Alt+M** after DM.EXE loaded. Select the 'Hard Disk Installation' menu, then select the hard drive (if there are more than one) and enter the 'Partition Setup and Configuration'.

> An existing partition will be marked by a preceding asterisk (*). To delete a partition press '**DEL**', to create a new partition press '**N**', to write an existing or new partition to the MBR press '**W**'.

Copy **With Disk Manager version 7.x:** Start the DM program, then select 'Maintenance menu' and 'Write Boot Code in MBR'. When done go to 'Update DDO' to complete the repair.

The repair of a single Disk Manager partition per physical drive is straightforward, just let the DM utility write the default data to the MBR. In case there were more than one partition on the drive, then you will need to manually edit the MBR with the existing configuration data. If you prepared an IV rescue diskette before the DM MBR was damaged then you'll find the parameters in a text file named HD_DATA.NTZ, on the rescue diskette. The parameters can be will be found under "Partitions Layout" in the HD_DATA file. Enter the data manually with DM.EXE, and rewrite the MBR to the hard disk.

> Disk Manager will take care of rewriting the dynamic overlay when running the above procedure.

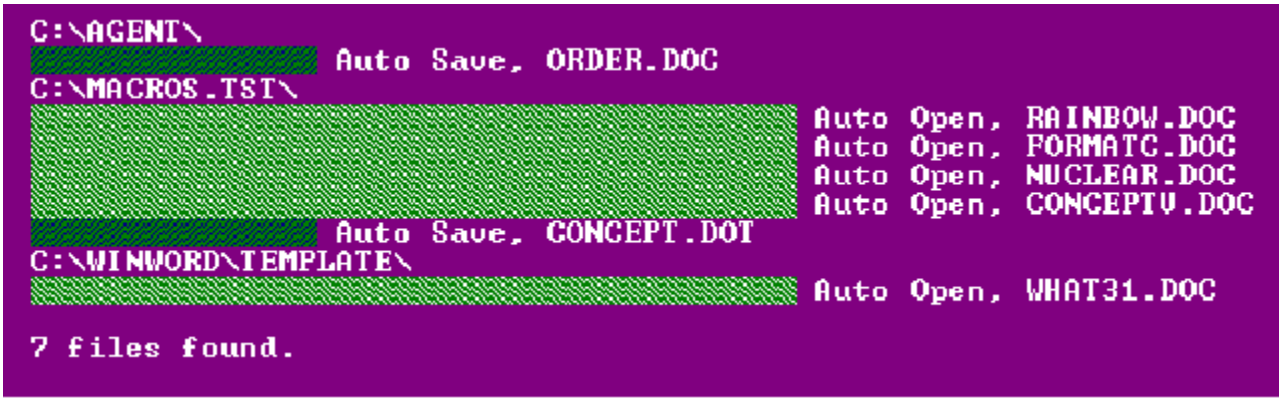
Protecting from Winword Macro Viruses

Copy
Copy

There is more hype than substance in the Word macro virus issue, yet justified or not, they are a serious concern to many, especially in the corporate and network environment.

InVircible uses generic methods for detecting and cleaning forced macros from Word files. **IVX**, the InVircible analyzer and correlator will screen Word files for the presence of forced macros and report them to screen and to a file. IVX is also used for cleaning Word files from forced macros.

Shown below is the IVX report obtained after scanning the author's hard drive for forced macros.



In the above example, seven files are reported to contain forced macros, yet only three contain a genuine macro virus (*Rainbow, ConceptV, and Nuclear*), one contains a macro Trojan Horse (*FormatC*) and the rest are benign, containing an auto executing macro. *ORDER.DOC* is an auto-saving order form, *WHAT31.DOT* is the template used for creating and authoring Windows help files (used by the author for creating the IV manual for Windows) and *CONCEPT.DOT* is a castrated (non replicating) template written to demonstrate the effects of the Concept macro virus. The latter won't replicate.

Not every file containing an auto executable macro is necessarily a virus or Trojan, as demonstrate ORDER.DOC and WHAT31.DOT, above. Yet the presence of an auto executing macro constitutes a higher risk than the presence of a forced save, or close one, as the user may not be given the opportunity to prevent the macro from executing and deliver its payload right on opening the document

IVX, when used with the **/MAC** switch searches for the presence of auto executing and forced saving macros in Word files. IVX does not interpret the macro contents as their presence in documents or in the global template usually isn't benign.

Yet it could be normal to find forced macros in customized and specialized templates, like the two mentioned above. For that purpose there is an exceptions list option in IVX. For details see below.

When finished scanning, IVX will display a report to the screen and save a copy in a file. Files containing an auto executing macro will show as long bright green bars, indicating they are a high risk. Files with auto saving or closing macros are indicated in the report as potentially risky, by a darker shade and a shorter bargraph. Macros of the latter type (auto save or close) are used in macro viruses to induce replication.

A copy of the report can be found in the root of the scanned drive. Where writing is not possible, like with CD-ROM, the report will be rerouted to C:\. The IVX main menu has an option for browsing through the last report.

To enter the macro mode, either start **IVX** from the command line with the **/MAC** switch, or select the Macro mode from IVX main menu. To clean documents from forced macros either use the **/MAC /R** switches from the command line, or select the '**Clean**' option from the IVX dialog when in the Macro mode. IVX will prompt for confirmation before cleaning the file.

The IVX Macro Exceptions List

Word templates may contain forced macros for absolutely benign purposes, like customized standard forms, or macro protection like MicroSoft's *SCANPROT.DOT* etc. Specific files can be excluded from the IVX macro checklist by the IVX exceptions list. Create a file named **IVX.INI** in the same directory as where IVX.EXE resides. For each file to exclude from the IVX checklist, add a line with the following syntax: **SKIP = THISFILE.DOT**

There is provision to exclude up to 20 files from IVX's checklist. Yet one is reserved for **IVMACRO.DOT**, IV's immunization against Concept, thus only 19 are left for the user's definition.

The default checklist of IVX in macro mode contains files with ***.DO?** and ***.XL?** extensions, only. To scan all files, start IVX in the interactive mode and select the '**All files**' option when prompted.

Handling Macro Malware in the Corporate Environment

IV provides several features that can ease controlling macro malware in the corporate and network environment. Users workstations can be checked and even cleaned from a remote drive or through the network login script.

IVX Errorlevel: IVX returns **errorlevel 1** on exit when at least one file containing a forced macro is found, or **errorlevel 0** when none. The errorlevel returned by IVX can be used to test the workstations for forced macro integrity, through the network login script.

The /NoPrompt Command: When run with the **/MAC** switch, the **/NoPrompt** command will induce the unconditional cleaning of files from forced macros without prompting before cleaning. The **/NoPrompt** option can be used in remote ordered cleaning of the workstations.

The anti Concept Immunization Macro: *Concept* is the most common macro virus found in the corporate environment. *Concept* is considered by many as their major virus problem, yet it is surprisingly simple to completely stop *Concept* from further spreading and remove it from already affected files. Like most viruses, *Concept* checks before infecting whether it already exists in the global macro, by checking for the presence of a macro named

'PayLoad'. The creation of a dummy macro with the same name will prevent *Concept* from infecting. The **IVMACRO.DOT** template will install the dummy payload macro to *normal.dot* when opened with Word. The same can be achieved by recording an empty macro and naming it 'PayLoad'.

Switches and Command Line Options.

Copy
Copy

The following are a list of the switches and command line options available in the InVircible modules.

INSTALL

/FAST	Upgrade an existing installation or install with the default parameters. The installation is automatic, no prompting is available and the registration key will not be installed to the hard disk.
/R	Prepare an IV rescue diskette.
DIR=pathname	Install to the specified directory instead of default.

IV

[drive:]	Load directory of the specified drive.
/V	Start IV in virus scanner mode.
/R	Start IV in ResQdisk mode.
/0	Start IV at zero level (all sub-menus closed).

IVINIT

/NOCMOS	Skip CMOS data comparison, recommended for notebooks and laptops. Also use on computers that use the CMOS for power saving ("Green") data.
/SKIPMEM	Disable Memory Stealing test to all programs. To reset the memory stealing test, run INSTALL and quit immediately.

IVTEST

/Q	Quiet mode. Do not display to screen, except when virus activity was found.
/SKIPMEM	Disable Memory Stealing test to all programs. To reset the memory stealing test, run INSTALL and quit immediately.

IVLOGIN

/Q	Check if the login workstation ran IVB DAILY on this date. IVLOGIN returns errorlevel 1 on exit if not checked today.
/RANDOM	Install with random signature file name.
SIG=name	Install with specified signatures' file name.

IVB

ALL	Run IVB on all local drives, starting from root (the default) or from specified directory. The ALL option overrides DAILY or WEEKLY.
AUDIT	Initialize audit log and mode. Available in registered version only.
DAILY	Run complete daily inspection of all local drives and directories.
/ESC	Enables escaping the daily check, in Sentry mode.
/NOSTOP	Do not stop for prompting the user. For unattended systems.
WEEKLY	Same as DAILY, but weekly.
-	Operate on specified directory only, do not continue to sub directories.
/A	Install IV Armor to floppy. Can only be used on floppies.
/D	Restore modified and recoverable files, delete modified and unrecoverable ones.
/R	Restore modified files.
/S	Secure all executable files, overwrites existing signatures.
/U	Uninstall IV Armor from floppy.
/V	Remove current signature files.
/X name	Overrides current signature filename, use "name" as signature filename.

IVSCAN

-	Space + hyphen. Operate only on specified directory, do not continue to sub directories.
/B	Inspect floppy boot sector, force boot sector replacement, if no original sector found.
/D	Remove virus if possible, delete file if no removal algorithm exists in IVSCAN.
/EX	Scan only files with executable extension filenames.
/R	Remove virus from infected file, if algorithm exists in IVSCAN.

IVX

arg1	First argument after program name: Pathname of file to use as sample. Wildcards are allowed.
arg2	Second argument: Path where to start scanning.
arg3	Third argument: Number of bytes past the entry point, where to look for a signature string.
/MAC	Start IVX in batch mode for searching or removing Word forced macros.

/R	Remove forced macros from Word files. The /R switch works only in conjunction with the /MAC swith, in registered IV only.
/NoPrompt	Remove forced macros without prompting.

RESQDISK

/B	Backup the MBR, boot sector and CMOS data.
/R	Restore the MBR, boot sector and CMOS data from image. Prompt before writing data to CMOS.
/Z	Track zero functions. Use with /B or /R, above.
/2	Operate on disk # 2. Use with all switches above.
/DM	Boot sector of Disk Manager partition. Use with /B, /R and /2, above.

FIXBOOT

/IBM	Force a PC-DOS type boot sector instead of the MS-DOS one.
/WIN95	Force Windows 95 boot sector style.
/S	Prompt for a floppy size (360, 720, 1.2, 1.44, 1.68, 2.88) and force a boot sector of the selected size.

NOCMOS

D	Disable the IVINIT CMOS test in the autoexec.
E	Re-enable the IVINIT CMOS test in the autoexec.

InVircible's Main Features

Copy
Copy

Copy It handles both known and unknown computer viruses.
Copy It is extremely fast, professional and easy to operate;
Copy It does not require frequent and costly updates;
Copy Is self-sufficient and provides real time recovery from new or unknown viruses. Does not need outside assistance;
Copy Has Extremely low false-alarm or 'false positives', with the highest probability of 'true positives' detection;
Copy Uses a unique unobtrusive file protection and restoration scheme.
Copy Has an automatically updating distributed database for restoration of files.
Copy Has redundant and user-defined security features.
Copy Uses a unique generic virus behavior probe and sampler.
Copy Is not memory-resident and does not adversely affect computer performance.
Copy Uses unique programs that automatically check themselves for, and restore from virus infection.
Copy Is friendly and uses a menu-driven user interface.
Copy Features an interactive installation utility.
Copy Is compatible with all major network software.
Copy Has an indispensable toolbox for computer security experts and advanced users.
Copy Utilizes sabotage-resistant protection designed to avoid both human- and virus-based deception.
Copy Is Piggybacking resistant to prevent InVircible from becoming a virus carrier.
Copy Has extensive context-sensitive on-line help, and
Copy Is financially affordable.

Glossary

Copy
Copy

Copy **Piggybacking:** Technique used by a fast infecting virus. The virus installs a memory resident handler that monitors DOS for 'open', 'close', 'find-first' and 'find-next' calls and hooks its infection routine to the captured handler.

Copy **Retro-Piggybacking:** A disinfection technique used by InVircible, taking advantage of piggybacking to clean after the virus.

Copy **Tunneling:** A method to bypass 'underneath' a BIOS or DOS service.

Copy **SeeThru:** A tunneling technique used by InVircible to 'tunnel' under interrupt 13h, used for absolute disk read/write, to thwart the stealthing of boot and partition infectors.

Copyright

Copy
Copy

Tunneling: A method to bypass 'underneath' a BIOS or DOS service.

SeeThru: A tunneling technique used by InVircible to 'tunnel' under interrupt 13h, used for absolute disk read/write, to thwart the stealthing of boot and partition infectors.

Piggybacking: Technique used by a fast infecting virus. The virus installs a memory resident handler that monitors DOS for 'open', 'close', 'find-first' and 'find-next' calls and hooks its infection routine to the captured handler.