Viruses and Anti-virus Measures on NetWare

J. Benjamin Sidle and Dr. Jan Hruska, Sophos Plc, Oxford, England

Part # tr00002k/950201

1. Introduction

Computer viruses spread through the interchange of executable code between computers. On Personal Computers (PCs) this interchange is much more frequent and less well regulated than on minicomputers and mainframes. Computer viruses have so far been almost exclusively confined to PCs.

The interchange of executables on non-networked PCs is mostly done by floppy disks and in consequence is relatively slow and physically controllable. PC networks enable high speed interchange and sharing of data and executables. This automated interchange is also comparatively difficult to control.

The danger from a large scale virus attack in a **non-networked** organisation is usually limited to a few PCs before it is spotted and disk interchange is blocked. In such an environment it is relatively easy to contain a virus. The likelihood of a large scale virus attack in a **networked** organisation is much greater and the chances of a successful early containment much smaller.

This report concentrates on Novell NetWare, and is a result of a theoretical and practical study of virus behaviour under NetWare 3.12 and NetWare 4.01. Although practical anti-virus measures described are specific to NetWare, much of it also applies to other network operating systems, such as LAN Manager. It is assumed that the network file server will be a dedicated machine.

2. Virus types and replication mechanisms

A virus is a deliberately written computer program which consists of two parts: Self-replicating code and the 'payload', which produces side-effects. In a typical PC virus, the replicating code may be between 400 and 2000 bytes long, while the size of the payload depends on the side-effects. Typically, this is a few hundred bytes.

The side-effects of a virus are limited only by the imagination of the virus author and can range from annoyance to serious vandalism.

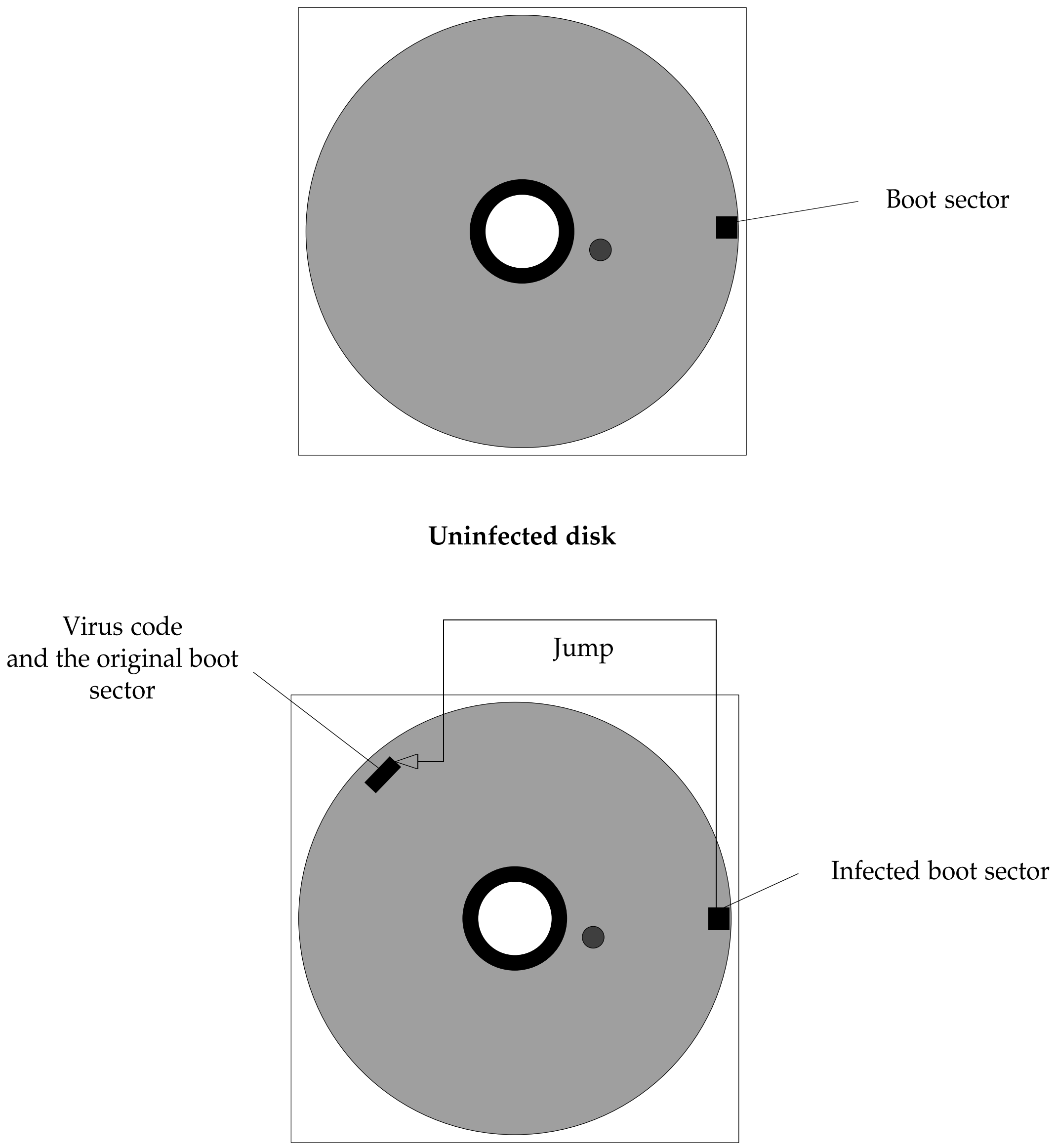
Viruses can be divided into five categories: **Bootstrap sector viruses**, **Parasitic viruses, Multi-partite viruses, Companion viruses** and **Link viruses**. Companion and link viruses could be assumed to be special cases of parasitic viruses.

2.1 Bootstrap sector viruses

Bootstrap sector viruses modify the contents of either the Master bootstrap sector (MBS) or the DOS bootstrap sector (DBS), depending on the virus and type of disk, usually replacing the legitimate contents with their own version. The original version of the modified sector is normally stored somewhere else on the disk, so that on bootstrapping, the virus will be executed first. This normally loads the remainder of the virus code into memory, followed by the execution of the original version of the bootstrap sector. From then on, the virus usually remains memory-resident until the computer is switched off.

Bootstrap sector viruses are spread through physical exchange of any media which can be used for bootstrapping (in most cases this means the physical exchange of floppy disks). In consequence, they spread comparatively slowly and **cannot spread over networks**. Nevertheless, one often finds Trojan horse programs whose only function is to infect the boot sector of a PC. Known as 'droppers', they allow the spread of boot sector viruses via bulletin boards, thereby vastly increasing the spreading potential and the speed with which the virus can travel.

A PC becomes infected with a boot sector virus only if the user (accidentally) bootstraps from an infected disk. It is completely safe to insert an infected disk into the drive and copy data from it (using the COPY command). The PC will not become infected unless it is booted while an infected disk is in drive A.



Uninfected disk

Fig. 1 - Disk infected with a bootstrap sector virus

Examples of bootstrap sector viruses include *Form* (DOS bootstrap sector), *Italian* (DOS bootstrap sector) and *New Zealand* (Master bootstrap sector).

2.2 Parasitic viruses

Parasitic viruses modify the contents of COM and/or EXE files. They append themselves to the file, leaving the bulk of the program intact. The execution flow is thus diverted in such a way that virus code executes first. Once the virus code has executed, control passes to the original program which, in most cases, executes normally. The extra execution time due to the virus is usually not perceptible by the user. Some viruses append themselves to the end of the original file, some prepend themselves at the beginning of the file, some do both and some insert themselves in the middle of the file.

Parasitic viruses spread through any medium which can be used for storage or transmission of executable code, such as floppy disks, tapes, **networks** etc.

The virus code is executed before the infected host program. The virus runs at the same privilege level as the original program and once running, can do anything: replicate, install itself into memory, release its side effects etc.

Most parasitic viruses such as *Cascade* spread when another (uninfected) program is loaded and executed. Such a virus, being memory-resident, first checks if the program is already infected. If it is not, the virus will infect it. If it is already infected, further infection is not necessary (although some parasitic viruses like *Jerusalem* do re-infect ad infinitum).

Parasitic viruses which are not memory-resident normally spread by finding the first uninfected program on disk and infecting it. One such example is the *Vienna* virus.

2.3 Multi-partite viruses

Multi-partite viruses exhibit the characteristics of both bootstrap sector and parasitic viruses. Viruses such as *Flip* infect COM and EXE files (like parasitic viruses) as well as the Master boot sector (like boot sector viruses). By exploiting 'the best of both worlds', their chances of replication are much higher than if they were to use only one method. It is not surprising that the comparatively few multi-partite viruses in existence today account for a disproportionately large number of infections.

Multi-partite viruses are spread through physical exchange of any media which can be used for bootstrapping (in most cases floppy disks), as well as through any medium which can be used for storage or transmission of executable code such as disks, tapes and **networks**. The virus will become active if the PC is bootstrapped from an infected disk or if an infected program is executed.

The network can act as a carrier of programs infected with a multi-partite virus. A workstation boot sector will become infected when an infected file is executed on it.

Most multi-partite viruses are fully multi-partite, which means that a PC infected by bootstrapping from an infected disk will infect other disks as well as executables, while a PC infected by executing an infected file will infect other executables as well as disks. Some multi-partite viruses are only partially multi-partite; for example, *Spanish Telecom* will infect other EXE and COM files as well as boot sectors, while the same virus in a boot sector will only infect other boot sectors.

The speed of propagation of multi-partite viruses is similar to that of parasitic viruses as they can be uploaded easily onto bulletin boards and thus spread over great distances very quickly.

2.4 Companion viruses

If two programs with the same name (before the dot, e.g. WS.COM and WS.EXE) exist in an MS-DOS directory, the command line interpreter will execute a COM file in preference to an EXE file. This property of the operating system is exploited by the companion viruses.

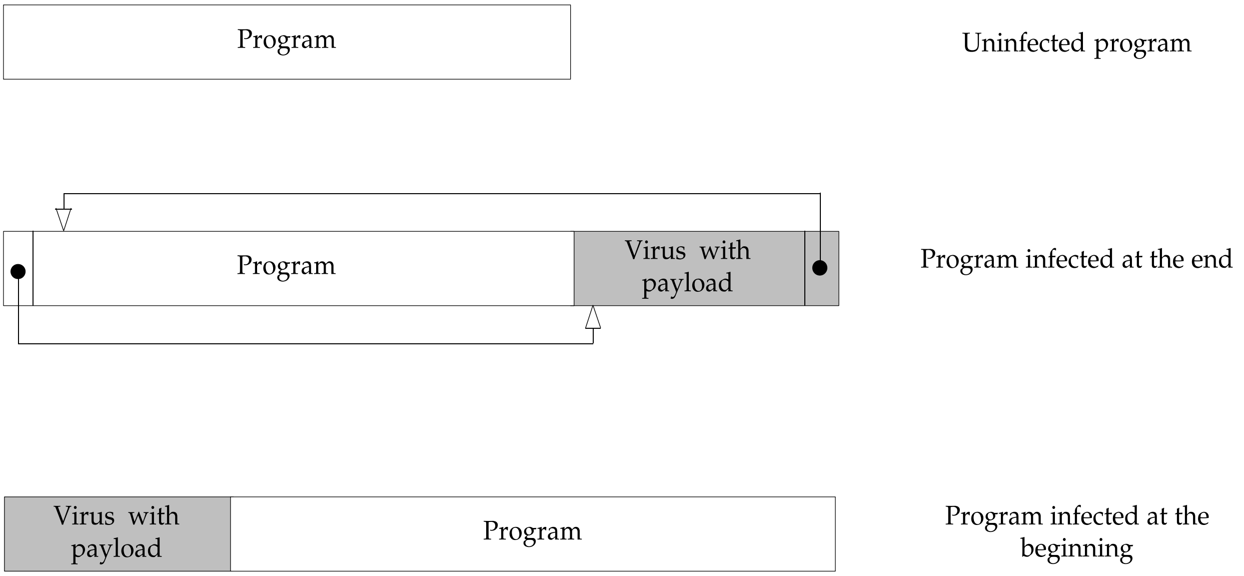


Fig. 2 - Program infection with a parasitic virus

A companion virus creates a COM file with the same name as the EXE file it ‘infects’, storing its own virus code in the COM file. When a user types in the program name, the operating system executes the COM file, which contains the virus code. This, in turn, loads and executes the EXE file. The virus makes **no change at all** to the contents of the ‘infected’ EXE file.

The directory listing below shows an unsophisticated companion virus which has infected WS.EXE by creating WS.COM. More sophisticated companion viruses label the companion COM file with the DOS ‘hidden’ attribute, which means that they will not be shown in directory listings. This, however, also hinders the spread of such viruses, since the DOS COPY command does not copy hidden files.

```
  Volume in drive C has no label
  Directory of C:\COMPANIO

 .   <DIR> 7-07-92 4:45p
 ..   <DIR> 7-07-92 4:45p
 WS EXE 30464 20-02-86 5:43p
 WS COM 4936 20-02-86 5:43p
  4 File(s) 51335168 bytes free
```

Companion viruses are spread through any medium which can be used for storage or transmission of executable code, including **networks** (but see above comment on hidden files). The virus will become active if one of its companion COM programs is executed.

It is unlikely that companion viruses will pose a major threat in the future.

2.5 Link viruses

Link viruses work by linking the first cluster pointer of the directory entry of one or more executable files to a single cluster containing the virus code. The original number of the first cluster is saved in the unused part of the directory entry.

Link viruses are spread through any medium which can be used for storage or transmission of executable code. A PC will become infected if an infected program is executed. Most network drives (including NetWare) **cannot be infected by link viruses,** since their physical directory structure is different from DOS, but they **can act as carriers of the link virus code,** which will spread as soon as it is executed on a workstation.

3. Virus infection under NetWare

Due to excellent emulation of physical DOS disks under NetWare, a large proportion of viruses in existence today are able to infect files on NetWare drives.

The main difference between NetWare and local workstation drives is that NetWare does not allow individual sector addressing either through DOS interrupts 25H and 26H or the BIOS interrupt 13H. This excludes the possibility of bootstrap sector viruses infecting the network, but does not exclude parasitic, multi-partite and companion viruses, all of which can spread freely on a badly protected network. Link viruses do not spread on network drives.

3.1 Virus entry into the network

The point of entry of a virus into a network is invariably the user workstation. In a typical scenario, the user infects his workstation by executing an application infected with a parasitic or a multi-partite virus, or bootstrapping from a disk infected with a multi-partite virus. The virus becomes memory-resident and will typically try to infect any application which is run, or any drive which is accessed.

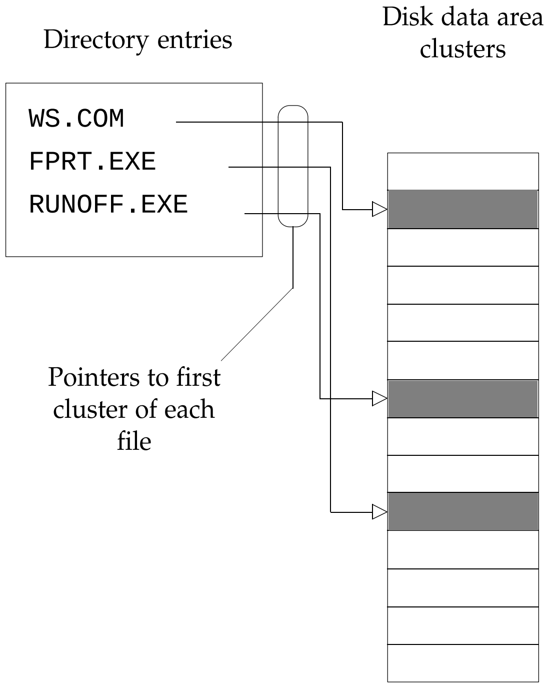
In order to access the network, the user has to execute LOGIN.EXE stored on the file server. If LOGIN.EXE itself (or any other executables) are unprotected, (see ‘Anti-virus measures on NetWare’), they will become infected. Any user executing an infected application will infect his workstation, which, in turn, will spread the infection further (Fig. 3).

On a typical network, an infection can spread onto most workstations within minutes. An infected LOGIN.EXE, or any program executed by the system login script, would infect the workstation whenever a user logs into the network.

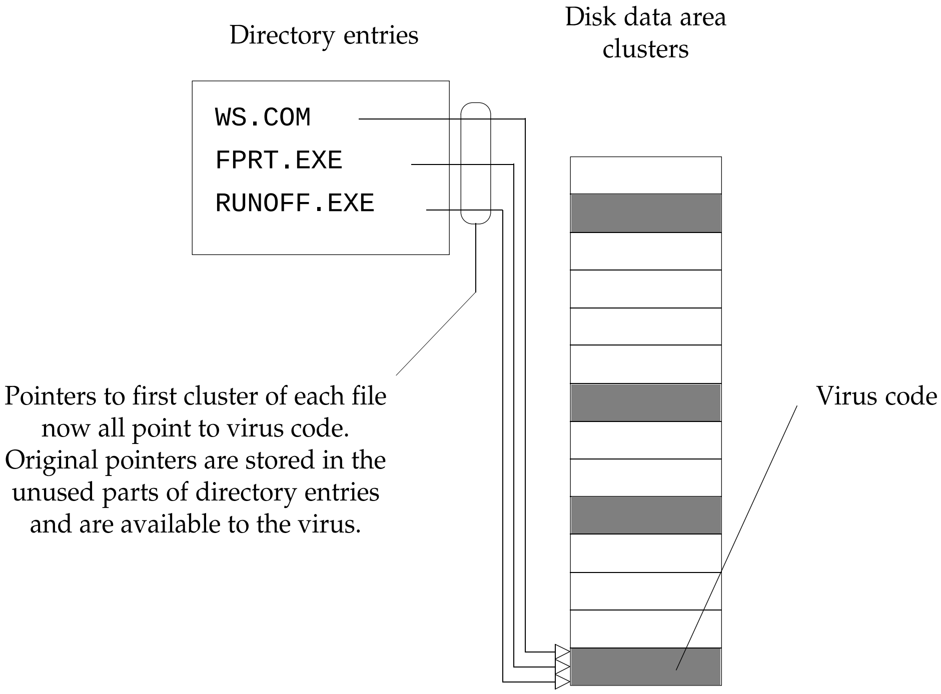
The above scenario has been demonstrated in practice by infecting a workstation with the *Jerusalem* virus and then executing LOGIN on the file server running NetWare 4.01. LOGIN.EXE was left protected only with read-only (R/O) attributes. *Jerusalem* (like most parasitic viruses) sets the R/O attribute to Read/Write (R/W), infects the file and resets the attribute to R/O. After LOGIN.EXE has been infected, any user logging into the network will infect his workstation. Any unprotected EXE or COM file residing on the file server will likewise be infected whenever executed.

3.2 Server infection by boot sector viruses

A server can be accidentally infected by a boot sector virus by bootstrapping it from an infected floppy disk. This is a rare occurrence which should



Directory entries in an uninfected system


Directory entries
Disk data area clusters
WS.COM
FPRT.EXE
RUNOFF.EXE
Pointers to first cluster of each file now all point to virus code. Original pointers are stored in the unused parts of directory entries and are available to the virus.
Virus code

Directory entries in a system infected with a link virus

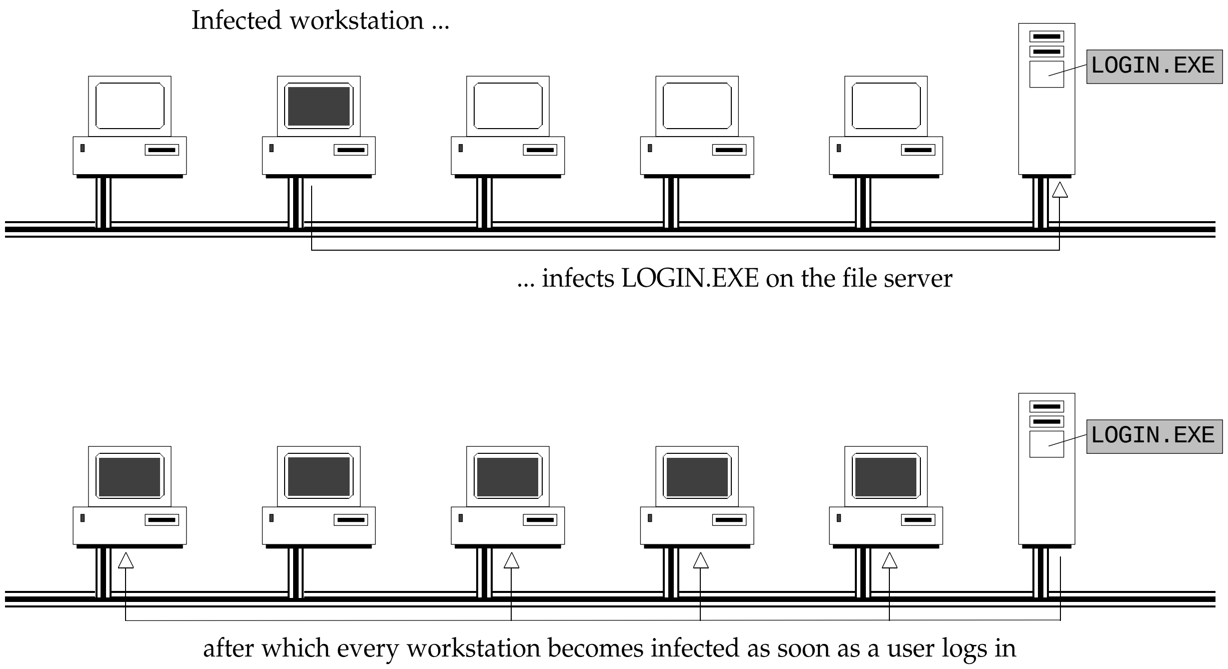

Infected workstation ...
LOGIN.EXE
... infects LOGIN.EXE on the file server
LOGIN.EXE
after which every workstation becomes infected as soon as a user logs in

Fig. 3 - Large scale network infection through LOGIN.EXE

happen in practice only in badly protected environments. NetWare servers have a Master boot sector and a DOS partition, and are susceptible to Master boot sector and DOS boot sector viruses.

It should be noted that even if a server becomes infected with a boot sector virus, **the virus will not spread to workstations connected to the server**.

4. NetWare-specific viruses

There are three cases of viruses reported to have been written specifically to circumvent NetWare security.

4.1 First Novell 'virus'

In February 1990 there appeared an (unconfirmed) report of a 'Novell' virus which supposedly destroyed the Novell-specific file allocation table. The virus was said to be capable of penetrating a file server from a workstation even if it was not logged on to the network. It was suggested that this might be possible by altering the NET$DOS.SYS program using C libraries released by Novell.

Novell has not encountered this virus.

4.2 Jon David's false alarm

In July 1990, New York consultant Dr. Jon David released a report about a virus which he claimed was capable of propagating on a Novell LAN. Dr. David said that the virus, a *Jerusalem* mutation, bypassed NetWare file server write-protection and also deleted write-protected files on the server.

Novell confirmed that the virus was *Jerusalem*, that it did propagate on unprotected networks, but denied the allegation that it bypassed NetWare security in any way.

4.3 NetWare virus from the Netherlands

In April 1991 Sophos received a virus (*GP1*) from the Netherlands which contained program sequences designed to subvert NetWare security. Interestingly enough, the virus was received in source-code form. It is believed to have been developed in Leiden (the Netherlands) as a result of an unofficial challenge by a state organisation employee to a student.

The virus is based on the *Jerusalem* virus, with NetWare-specific instructions added to the disassembled version of *Jerusalem*. It is memory-resident but contains no stealth characteristics.

The virus is not infective unless it is run on a NetWare workstation. It intercepts the user LOGIN procedure and executes LOGIN.EXE under its own control. If the LOGIN is successful, the virus sends a copy of the original login request block to the socket number 2A9FH.

The virus replicates in the same way as *Jerusalem* (when NetWare is present), but no other effects could be observed. The virus seems to be an unfinished creation.

5. Stealth viruses and NetWare

Virus stealth (or 'hiding') has special implications on any network, due to the difficulty in establishing a 'clean', virus-free work environment.

Interrupt interception is the most common technique used by stealth viruses. The virus redirects the interrupt vectors in such a way that operating system service calls are redirected to the virus code first (Fig. 4). For example, the virus can examine every request made to the operating system for reading disk information. If the sectors requested are those used by the virus, their contents are falsified before further processing of the request. This is the tactic used by the *Joshi* virus, which intercepts any call to read the disk bootstrap sector and substitutes the original contents in place of the virus-infected actual contents.

The main problem of dealing with stealth viruses on any network is the difficulty in establishing a positively 'clean' work environment from which the cleanup can be attempted (see 'Secure accessing of NetWare').

Interrupt interception represents a particular problem when dealing with an infected network. Viruses such as *4K* will hide their presence by intercepting different interrupt services, including file-open and file-close. The virus disinfects each file as it is opened it and re-infects it on closing, thus effectively hiding its presence from anti-virus software.

6. NetWare 3.12 practical trials

A set of tests were run on Novell NetWare 3.12 using a Compaq 386 as a server and a portable Amstrad 386 as a workstation. A set of different viruses were copied onto the workstation and attempts were made to infect files on the server. An infected file was then copied onto the server and run from there to see if the workstations could be infected from the server.

6.1 Jerusalem-a

Jerusalem is a memory-resident virus which infects both COM and EXE files. It exists in many variants; the version used in these experiments was *Jerusalem-a.*

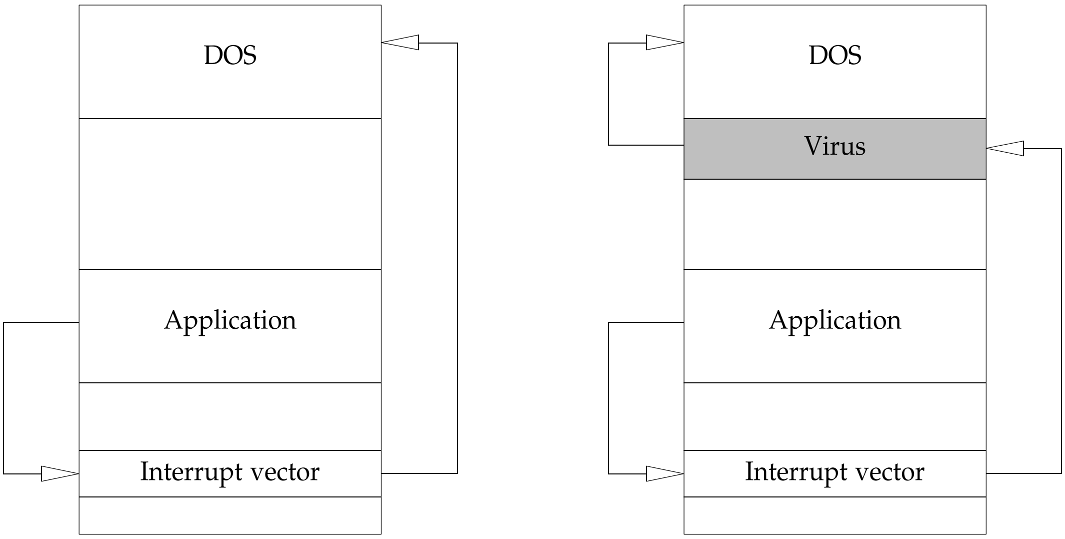


Fig. 4 - Interrupt routing before and after infection

If the network files IPX and NETX are loaded and then a file infected with *Jerusalem* is run, the workstation hangs giving the message:

```
SHELL-331-76 NETWORK ERROR ON
SERVER:Error receiving from network
Abort, Retry?
```

If NETX is infected, it will not run. The file NETX is also corrupted.

If a file infected with *Jerusalem* is copied onto the server and run, the workstation attempting to run the file hangs, giving the same error as before.

6.2 4K (alias Frodo)

4K is a sophisticated memory-resident virus which makes strenuous efforts to conceal its presence by intercepting a large number of DOS interrupts when active (see 'Stealth viruses and NetWare').

The virus will successfully load both before and after NETX is run on the workstation, without causing the workstation to crash. However, it will not infect files on the server. The virus will also infect NETX and IPX without causing the workstation to crash.

If an infected file is copied onto the server and run, files on the workstations are infected, but files on the server are not (regardless of access rights).

6.3 Cascade (variant 1701)

Cascade is a memory-resident parasitic virus which only infects COM files (including COMMAND.COM). When the virus triggers (if the date is October-December 1988) the characters on the screen 'fall' to the bottom of the display.

The virus will run before and after NETX without preventing the workstation from connecting to the server. The virus is unable to infect files on the server (regardless of access rights) and when the virus is copied onto the server and run, it does not infect files on the workstation or the server.

6.4 Vienna-627

This virus is non memory-resident and only infects COM files. The source code for this virus has been widely disseminated thanks to Ralf Burger's book *Computer Viruses, a high-tech disease*. In consequence, there are more variants of this virus than any other virus.

If *Vienna* is run on the workstation, it is able to infect COM files on the server if NetWare has been installed with the default attributes for files and directories. (The defaults are: files - archive, directories - able to create and write to files.)

If the directory rights are set to NO CREATE, *Vienna* will still infect files. However, if the directory rights are set to NO WRITE, the files are not infected.

If the file attributes are set to READ ONLY, *Vienna* is able to infect files. With the file attributes set to EXECUTE ONLY (which requires *SUPERVISOR* privilege), the files are not infected.

If an infected file is run on the server, the files on both the server and the workstation are infected.

Vienna will infect the NetWare file IPX without interfering with the ability of the workstation to connect to the network.

6.5 Eddie-651

This is a small non-destructive virus from Bulgaria which (when memory-resident) will conceal the increase in file size. It places a time stamp of 62 in the seconds field in order to recognise infected files.

If a file infected with *Eddie* is run before NETX, the workstation will hang. If the infected file is run after NETX has run, the files on the server with the default attributes for files and directories are infected.

If the directory rights are set to NO CREATE, *Eddie* will still infect files. However, if the directory rights are set to NO WRITE, the files are not infected.

If the file attributes are set to READ ONLY, *Eddie* is able to infect files. With the file attributes set to EXECUTE ONLY, the files are not infected.

Both IPX and NETX are infected by *Eddie* but NETX fails to run properly when infected.

6.6 Little Brother-321

Little Brother is a memory-resident companion virus which will create companion files on the server which uses the default file and directory attributes.

If the directory rights are set to NO CREATE, no companion is created; if the rights are set to NO WRITE, a companion is created. **If the file attributes are set to READ ONLY or EXECUTE ONLY, a companion is still created.**

If an infected file is run on the server, the files on the workstation and the server are infected.

If a NETX companion is created, no companions are created for other files.

6.7 TPWorm-12969

TPWorm-12969 is a non memory-resident companion virus which will create companion files on the server which uses the default file and directory attributes.

If the directory attributes are set to NO CREATE or NO WRITE, no companion is created. With file attributes set to READ ONLY, no companion is created; with the file attribute set to EXECUTE ONLY, a companion is created.

If an infected file is run on the server, companions are created on the server but not the workstation.

6.8 Tequila

Tequila is a self-modifying, encrypted multi-partite virus which can infect both EXE files and the Master boot sector of the hard disk.

The virus will run before or after NETX, without affecting the ability of the workstation to connect to the network. The virus will not infect files on the server.

If an infected file is run from the server, the Master Boot sector of the workstation is infected.

6.9 Form

Form is the most common virus in the wild. It is a DOS boot sector virus and does not spread across the network.

7. NetWare 4.01 practical trials

A similar set of tests were run on Novell NetWare 4.01 using a Compaq 386 as a server and another Compaq 386 as a workstation. The viruses were copied onto the workstation and attempts were made to infect files on the server. An infected file was copied onto the server and run in order to see if the workstations could become infected. Using the Novell supplied drivers, the file NETX.EXE in NetWare 3.11 is replaced by VLM.EXE and NET.VLM.

7.1 Jerusalem-a

If *Jerusalem-a* is run before the NetWare files, the NetWare fails to load displaying the message:

```
IPXODI -211-0:Unable to locate
message file IPXODI.MSG
Program load aborted.
```

However, if *Jerusalem* is run after the NetWare files are loaded, the workstation continues to run and the virus becomes active in memory. Files on the server with default attribute settings become infected.

If the directory rights are set to NO CREATE, *Jerusalem* will still infect files. However, if the directory rights are set to NO WRITE, the files are not infected.

If the file attributes are set to READ ONLY, *Jerusalem* is able to infect files. With the file attributes set to EXECUTE ONLY (this requires *SUPERVISOR* privilege), the files are not infected.

If a file infected with *Jerusalem* is copied onto the server and run, the files on both the workstation and the server can become infected.

7.2 4K (alias Frodo)

4K will successfully load both before and after the NetWare connection files are run on the workstation, without causing the workstation to crash. Unlike NetWare 3.11 & 3.12, the files on the server which uses default file attributes, are infected.

If the directory rights are set to NO CREATE, *4K* will still infect files. However, if the directory rights are set to NO WRITE, the files are not infected.

If the file attributes are set to READ ONLY, *4K* is able to infect files. With the file attributes set to EXECUTE ONLY, the files are not infected.

If an infected file is run from the server, the files on the workstation and the server are infected.

7.3 Cascade (variant 1701)

Cascade will run before and after the NetWare connection files, without preventing the workstation from connecting to the server. The virus is able to infect files on the server which uses default file attributes.

If the directory rights are set to NO CREATE, *Cascade* will still infect files. However, if the directory rights are set to NO WRITE, the files are not infected.

If the file attributes are set to READ ONLY, *Cascade* is able to infect files. With the file attributes set to EXECUTE ONLY, the files are not infected.

If an infected file is run on the server, files on the workstation and the server are infected.

7.4 Vienna-627

If default NetWare attributes for files and directories are used, *Vienna* is able to infect COM files on the server. The defaults are: files - archive, directories - able to create and write to files.)

If the directory rights are set to NO CREATE, *Vienna* will still infect files. However, if the directory rights are set to NO WRITE, the files are not infected.

If the file attributes are set to READ ONLY, *Vienna* is able to infect files. With the file attributes set to EXECUTE ONLY, the files are not infected.

If an infected file is run on the server, the files on both the server and the workstation are infected.

7.5 Eddie-651

If a file infected with *Eddie* is run before the NetWare connection files, the workstation will hang. If the infected file is run after the NetWare files have run, the files on the server with the default attributes are infected.

If the directory rights are set to NO CREATE, *Eddie* will still infect files. However, if the directory rights are set to NO WRITE, the files are not infected.

If the file attributes are set to READ ONLY, *Eddie* is able to infect files. With the file attributes set to EXECUTE ONLY, the files are not infected.

7.6 Little Brother-321

If the directory rights are set to NO CREATE, *Little Brother* will not create companion files but if the rights are set to NO WRITE, a companion is created. If the file attributes are set to READ ONLY or EXECUTE ONLY, a companion is still created.

If an infected file is run on the server, the files on the workstation and server are infected.

7.7 TPWorm-12969

If the directory rights are set to NO CREATE, *TPWorm* will not create companion files, but if the directory rights are set to NO WRITE, a companion is created. With file attributes set to READ ONLY no companion is created; with the file attribute set to EXECUTE ONLY, a companion is created.

If an infected file is run on the server, companions are created on the server and the workstation.

7.8 Tequila

Tequila can infect before or after the NetWare connection files are run, without affecting the ability of the workstation to connect to the network. The virus will not infect files on the server.

If an infected file is run from the server, the Master Boot sector of the workstation is infected.

7.9 Form

Form does not spread across the network.

8. NetWare security mechanisms

NetWare provides four different aspects of network security: the login procedure, trustee rights, directory rights and file attributes.

1. The login procedure requires all users to identify themselves by a username and a password.
2. Trustee rights are granted to each user by the network supervisor and allow each user various actions such as reading from files, writing to files, creating files etc.
3. Directory rights (read, write, open, close, delete, search) are set separately and can be used to limit the access to certain directories such as those containing executables.
4. File Attributes (read-only, read-write, share) can be set separately.

If the security features are properly implemented, the infection cannot spread to the file server even if a user's workstation becomes infected.

This security does break down if a workstation becomes infected while the user is logged in as a network supervisor, since the *SUPERVISOR* account has write rights to all areas of the file server. Care should be taken whenever logging in as *SUPERVISOR*.

9. Anti-virus measures on NetWare

9.1 Diskless workstations

Diskless workstations are PCs in their own right, sometimes equipped with hard disks, but without floppy disks. The reasoning is that if the user does not have the means of introducing floppy disks into the PC, neither will he have the opportunity of introducing a virus.

This 'no-floppy no-virus' argument only holds up to a certain extent. It is quite true that diskless workstations will prevent accidental introductions of viruses onto the network. However, the malicious introduction of viruses is not prevented, as the virus code can be input through the keyboard using the DOS COPY command or using

DEBUG. The techniques are described in Burger's *Computer Viruses - A High Tech Disease*.

Likewise, diskless workstations can still have modem connections over which software can be downloaded from BBSs. Another disadvantage of diskless workstations is that the transfer of legitimate data by users is made much more difficult. The decision on whether to use diskless workstations in an organisation is a major one. Associated costs and the impact on the efficiency of the organisation should be carefully considered.

9.2 Remote bootstrap ROMs

Most network cards can be fitted with a special Read Only Memory (ROM) chip which reads the operating system and other associated files from the file server, instead of from the local disk during boot-up. Note that the PC will still try to bootstrap from the floppy disk first. If none is present in the drive, the bootstrapping will be performed remotely.

There are several advantages in using remote bootstrap ROMs. Firstly, from the virus protection point of view, the technique diminishes the danger of bootstrap sector virus infection. Secondly, any updates to the operating system used are made much easier, since they can be done on the file server.

The use of remote bootstrap ROMs is recommended for bootstrapping diskless workstations.

9.3 Enhanced access control

NetWare provides very good access control features and utilities for the administration of users.

9.4 Two IDs for network supervisors

One of the weak points in any multi-user computer system is that one or more users must be given high privileges necessary for system administration. Unfortunately, these privileges are also assigned to a virus whenever it infects a workstation with the user logged in as a network supervisor. In fact, the *GP1* NetWare-specific virus seems to exploit exactly that feature by trying to capture the network supervisor password.

One way of reducing the danger from virus penetration via this route is to reduce the time that network supervisors are logged in as *SUPERVISOR*. They should have two user IDs, one with *SUPERVISOR* privileges and the other with restricted privileges (for example, read access to all directories, which is especially useful when virus-checking a network). The use of the former should be limited to system administration functions.

9.5 DOS-based anti-virus software

DOS-based anti-virus software can be used to detect viruses on networks. Two types of software can be used: virus-specific and virus-non-specific.

Regardless of which type of software is used, proper procedures must be followed to ensure that the machine running anti-virus software does not have any viruses resident in memory. If this is not the case, stealth viruses can use hiding techniques to prevent the software from discovering them (see 'Secure accessing of NetWare'). Stealth techniques are not effective if the anti-virus software is run as a process on the file server itself (see 'Server-based anti-virus software').

9.5.1 Virus-specific software

A virus-scanning program relies on the knowledge of known virus characteristics. When a new virus appears in the wild, it is analysed, and its characteristics recorded. The virus-scanning program examines all executables on a disk, including the operating system and the bootstrap sector(s), and compares their contents with the characteristics of known viruses.

This type of software can only discover viruses that it 'knows' about and as such has to be continually updated with new information, as new viruses appear.

9.5.2 Virus-non-specific software

Virus-non-specific or checksumming software relies on the calculation of a checksum of every critical executable on the system, followed by periodic recalculations in order to verify that the checksum has not changed. If a virus infects an executable, it causes a change which will result in a completely different checksum (providing a strong checksumming algorithm is used).

This type of software is reactive rather than proactive, in that a virus attack will be detected after it happens. Checksumming software also relies on the fact that the executables are 'clean' before the initial checksumming is applied. This can be ensured by using virus-specific scanning software to check the system for the presence of known viruses.

The checksumming approach is the only known method which will detect all viruses, present and future, with absolute certainty. This makes it inherently desirable as a long-term anti-virus strategy in any organisation.

It is recommended that virus-non-specific software is used on NetWare in a fashion similar to the virus-specific software. The main problem is deciding which areas of the file server should be fingerprinted and checked regularly. On NetWare it is recommend that all executables in the \PUBLIC, \SYSTEM and \LOGIN subdirectories are fingerprinted. In addition, each system will have subdirectories containing applications software; these should be fingerprinted as well. Checking of the fingerprints is best done from a separate, securely booted workstation.

9.6 Server-based anti-virus software

The use of server-based anti-virus software is recommended, since it has several advantages over DOS-based software: virus stealth is not effective, the network traffic is minimised and the control over the software is centralised.

Server-based virus scanners are commonly used, since the problems associated with updating the master copy are minimal: one copy is held on the file server and can be easily updated. The scanning process can be performed either overnight, minimising the network workload, or as a continuous low priority task.

Server-based virus checking also ensures that the process is not subverted by the presence of stealth viruses, as they are unable to run on the server. Were the server to be checked from an infected workstation with the stealth virus memory-resident, the process would be ineffective. Furthermore, in the case of the virus *4K*, if the server is scanned from an infected workstation, any files on the server with WRITE access would be infected during the checking process and reported as clean!

Workstations can also be checked by the server-based scanner by using the proprietary InterCheck technology from Sophos, which enables true client-server virus detection.

9.7 Secure accessing of NetWare

With the advent of stealth viruses, it is most important to guarantee a clean, virus-free environment on a workstation, before running anti-virus software or investigating a virus-infected network from a DOS workstation.

9.7.1 NetWare 3.11 and 3.12

Prepare a DOS system disk which, in addition to a correct version of DOS system files and COMMAND.COM, also contains the following NetWare files: IPX.COM, NETX.EXE, LOGIN.EXE and MAP.EXE.

Write-protect the floppy disk.

To access the network, switch the workstation PC off, and boot from the floppy disk.

Run IPX first, followed by NETX. Run LOGIN from the floppy disk including the '/S NUL' command line qualifier. This will prevent the execution of both system and user scripts:

```
LOGIN /S NUL <USERNAME>
```

9.7.2 NetWare 4.00 and above

Prepare a DOS system disk. Configure the NetWare workstation client files for the Ethernet card you are using, including the STARTNET.BAT file.

Write-protect the floppy disk.

To access the network, switch the workstation PC off, and boot from the floppy disk.

Run STARTNET.BAT followed by LOGIN including the '/S NUL' command line qualifier.:

```
LOGIN /S NUL <USERNAME>
```

9.8 Tightening NetWare security

NetWare allows the setting of file attributes to execute-only. This prevents their modification or reading by any user, including the system supervisor - the only thing that the *SUPERVISOR* can do (apart from executing them) is to delete them.

Setting the execute-only attributes has mixed blessings. On the one hand it prevents the modification of executables, but on the other, it makes them unreadable (and unverifiable) by anti-virus software (either DOS-based or server-based).

It is recommend that this attribute is not used and that instead WRITE rights and CREATE rights are removed from directories containing executable files.

10. Conclusions

10.1 NetWare Administration

1. Set NetWare directory and user rights correctly.

2. Do not rely on default NetWare attribute settings.

3. Do not use NetWare execute-only attributes unless absolutely necessary.

4. Use secure bootstrap procedure before logging in as a *SUPERVISOR*.

10.2 NetWare 3.11 & 3.12 Virus Infections

1. NetWare 3.11 & 3.12 seems to cause more memory-resident viruses to malfunction than NetWare 2.12.

2. Some memory-resident parasitic viruses interact with IPX and NET3 losing the ability to infect. Some memory-resident parasitic viruses crash the workstation if IPX and NET3 are already loaded when the virus-infected application is run.

3. Most parasitic viruses will infect NetWare 3.11 & 3.12 files protected with a read-only attribute.

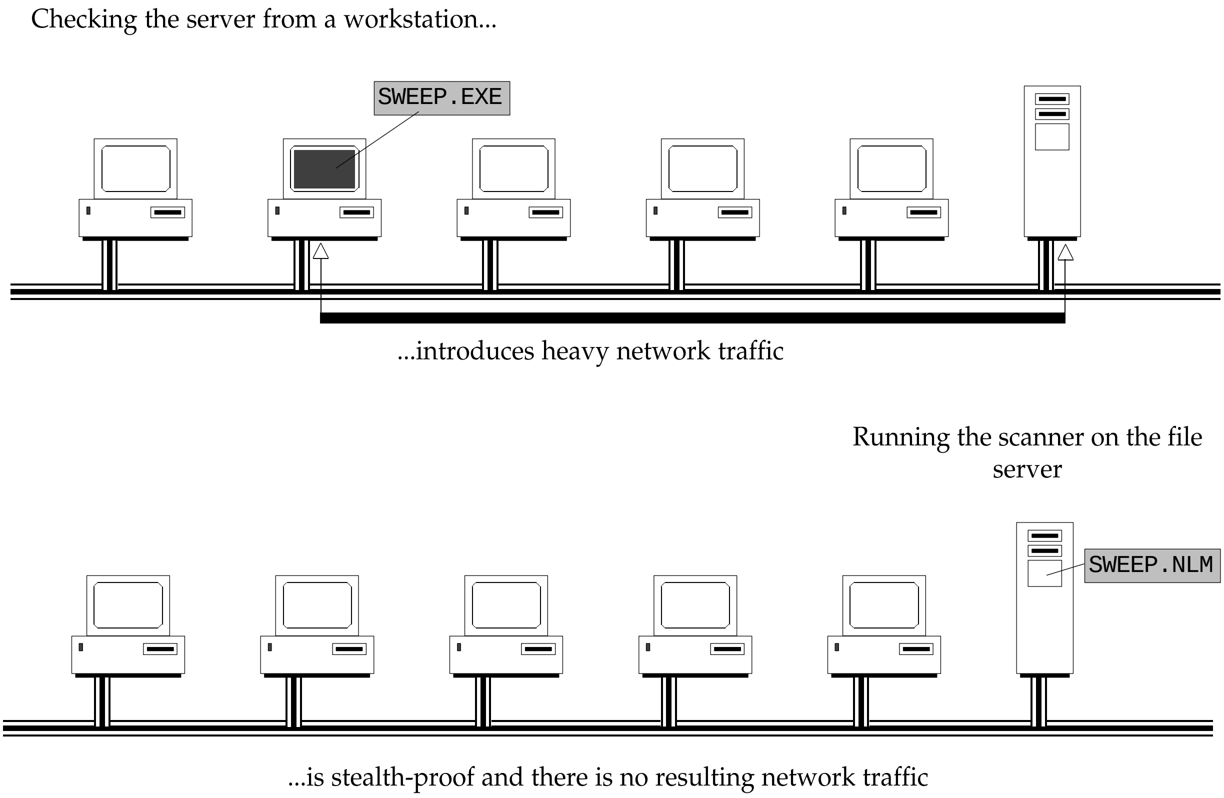
4. Parasitic viruses will not infect NetWare 3.11 & 3.12 files when the user's effective rights do not include WRITE rights. The supervisor has WRITE rights to all directories.

5. Parasitic viruses will not infect NetWare 3.11 & 3.12 files with execute-only attributes set, regardless of the user. This, however, is not a foolproof protection against future viruses.

6. Bootstrap sector viruses and link viruses will not infect NetWare 3.11 & 3.12 drives.

7. Multi-partite viruses will infect unprotected NetWare 3.11 & 3.12 executables.

8. Parasitic and Multi-partite viruses will infect executables regardless of protection levels (execute-only files excepted) if the user is logged in as a supervisor.



Checking the server from a workstation...

...introduces heavy network traffic

Running the scanner on the file server

...is stealth-proof and there is no resulting network traffic

Fig. 5 - DOS based and server-based scanning

9. Companion viruses will ‘infect’ any directory in which file creation is allowed.

10.3 NetWare 4.01 Virus Infections

1. NetWare 4.01 allows more memory-resident viruses to run correctly than previous versions of NetWare. This is not a particular weakness of NetWare 4.01; it is due to the chance interactions of NetWare 4.01 and viruses. Taking the security precautions already outlined, the risks can be reduced to the level similar to NetWare 3.XX.

10.4 Other Points

1. Consider using diskless workstations.
2. Use remote bootstrap ROMs in the workstations.
3. Use server-based anti-virus software.

11. Viruses mentioned in this report

11.1 4K

4K has an infective length of 4096 bytes. It may occasionally cause damage to files, as it manipulates the number of available clusters, which results in crosslinked files. If the virus is resident in memory, it disguises itself from detection by scanning or checksumming programs. Infected systems hang on 22nd September.

11.2 Cascade

This encrypted virus attaches itself to the end of COM files, increasing their length by 1701 bytes. The encryption key includes the length of the infected program, so infected files of different lengths will look different. After infection it becomes memory-resident and infects every COM file executed, including COMMAND.COM. The original version will produce a ‘falling characters’ display if the system date is between 1st October and 31st December 1988. This occurs at a random time after infection (maximum of 5 minutes).

11.3 Eddie-651

This non-destructive virus from Bulgaria marks infected files with a value of 62 in the seconds field of the timestamp. Infected files grow by 651 bytes, but this will not be seen if a DIR command is used - the virus intercepts the find-first and find-next functions, returning the correct (uninfected) length.

11.4 Form

Form is a boot sector virus from Switzerland which infects the DOS boot sector of hard disks and floppy disks. On the 18th day of every month it will produce a noise when keys are pressed. The original boot sector is stored in the last physical sector of the hard disk.

11.5 Jerusalem-a

The virus attaches itself to the beginning of COM files or at the end of EXE files. When an infected file is executed, the virus becomes memory-resident and will infect any COM or EXE program run, except COMMAND.COM. COM files are infected only once, while EXE files are re-infected every time that they are run. Infective length is 1813 bytes (COM) and 1808 bytes (EXE). After the system has been infected for 30 minutes, row 5 column 5 to row 16 column 16 on the screen are scrolled up two lines, creating a ‘black window’ and the system slows down. If the system is infected when the date is set to the 13th of any month which is also a Friday, every program run will be deleted.

11.6 Little Brother-321

This virus is a member of the *Little Brother* family. A companion file (size 321 bytes) is created for all EXE files executed on an infected system. On 3rd November, executing an infected file will delete all the files in the current directory.

11.7 Tequila

An encrypted, multi-partite, self-modifying virus from Switzerland. Contains encrypted text ‘Welcome to T.TEQUILA’s latest production’, ‘Contact T.TEQUILA/P.o.Box 543/6312 St’hausen/Switzerland’. The original Master boot sector is stored in the first sector after the end of the first partition, which is decreased by 6 sectors after infection. *Tequila* displays a crude Mandelbrot set pattern on screen if a complex set of conditions is fulfilled.

11.8 TPWorm-12969

This Bulgarian companion virus was first made available in ‘C’ source form. It is 12969 bytes long.

11.9 Vienna-627

This virus infects the end of COM files. Infective length is 627 bytes. It looks through the current directory and the directories in the PATH for an uninfected COM file. One file in eight becomes overwritten. The seconds stamp of an infected file is set to 62.

Bibliography and references

1. 4K, A New Level Of Sophistication, *Virus Bulletin*, May 1990
2. J. Bates, A Novell-Specific Virus, *Virus Bulletin*, June 1991
3. R. Burger, *Computer Viruses, a high-tech disease*, Abacus, 1988
4. F. Cohen, *A Short Course on Computer Viruses*, ASP Press, 1991
5. Editorial, *Virus Bulletin*, February 1990
6. Editorial, *Virus Bulletin*, December 1990
7. R. Glath, *Virus Propagation on Novell*, Virus Bulletin, December 1990
8. H. J. Highland, *Computer Virus Handbook*, Elsevier, 1990
9. L. J. Hoffman, *Rogue Programs: Viruses, Worms and Trojan Horses*, Van Nostrand, 1990
10. J. Hruska, *Computer Viruses and Anti-Virus Warfare*, Ellis Horwood, 1992

Glossary

Bad Sectors: During formatting of MS-DOS disks, all sectors are checked for usability. Unusable sectors are labelled as bad and are not used by DOS. The remaining areas can then still be used. Viruses sometimes label good sectors as bad to store their code outside the reach of the users and the operating system.

.BAT: The extension given to 'batch' file names in MS-DOS. A batch file contains a series of MS-DOS commands, which can be executed by using the name of the file as a command. AUTOEXEC.BAT is a special batch file which is executed whenever a PC is switched on, and can be used to configure the PC to a user's requirements.

BIOS: The Basic Input/Output System of MS-DOS which constitutes the lowest level of software which interfaces directly with the hardware of the microcomputer. The BIOS is usually stored in a ROM chip.

Boot Sector Virus: A type of computer virus which subverts the initial stages of the bootstrapping process. A boot sector virus attacks either the Master bootstrap sector or the DOS bootstrap sector.

Booting-up: A process carried out when a computer is first switched on or reset, where the operating system software is loaded from disk (either hard disk or floppy disk).

Bootstrap Sector: Part of the operating system which is first read into memory from disk when a PC is switched on (booted). The program stored in the bootstrap sector is then executed, which in turn loads the rest of the operating system into memory from the system files on disk.

Checksum: A value calculated from item(s) of data which can be used by a recipient of the data to verify that the received data has not been altered. Usually 32 or 64 bits long.

CMOS: Complementary Metal-Oxide Semiconductor is a technology used to manufacture chips which have very low power consumption. CMOS chips are used in battery-backed applications such as the time-of-day clock and for the non-volatile storage of parameters in IBM-ATs.

.COM: The extension given to a type of executable files in MS-DOS. They are similar to EXE files, but can only contain up to 64K of code and data. In operating systems other than DOS, the extension .COM can have a different significance.

Companion Virus: A virus which 'infects' EXE files by creating a COM file with the same name and containing the virus code. They exploit the PC-DOS property that if two programs with the same name exist, the operating system will execute a COM file in preference to an EXE file.

Dropper An EXE or a COM file which infects a PC with a virus, but which itself does not replicate. Commonly used for spreading boot sector viruses via bulletin boards.

Encryption: A process of disguising information so that it cannot be understood by an unauthorised person.

.EXE: The extension given to executable files in MS-DOS. These are similar to .COM files, but can contain more than 64K of code and data.

False Negative: An existent event reported as non-existent, e.g. the absence of a virus when the virus is present.

False Positive: A non-existent event reported as existent, e.g. the presence of a virus when no virus is present.

FAT: File Allocation Table, a mnemonic term used by the MS-DOS operating system (and others) to describe the part of a disk which contains information describing the physical location on the disk of the chains of clusters forming the files stored on that disk.

File Server: A central data repository for a computer network, which may provide other centralised services such as shared printer control.

Interrupt: A mechanism by which a process can attract the immediate attention of the CPU, usually in order to serve an urgent request from an external device. Interrupt table on 8086 microprocessors occupies the bottom 1K of RAM.

Link Virus: A virus which subverts directory entries to point to the virus code.

Master Bootstrap Sector: The first physical sector on the hard disk (sector 1, head 0, track 0) which is loaded and executed when the PC is bootstrapped. It contains the partition table as well as the code to load and execute the bootstrap sector of the 'active' partition. Common point of attack by boot sector viruses.

Multi-partite Virus: A virus which infects both boot sectors and executable files, thus exhibiting the characteristics of both boot sector viruses and parasitic viruses.

.OVL: The extension commonly given to overlay files in MS-DOS. Overlay files are used with large programs which cannot fit into RAM: parts of the program are loaded as and when needed. Overlay files can have any extension, not just .OVL.

Parasitic Virus: A computer virus which attaches itself to another computer program, and is activated when that program is executed. A parasitic virus can append itself to either the beginning or the end of a program, or it can overwrite part of the program.

Partition Table: A 64-bit table found inside the Master bootstrap sector on hard disks which contains information about the starting and ending of up to four partitions on the hard disk. The partition table also contains information on the type of the partition, e.g. DOS partition, UNIX partition etc.

Polymorphic Virus: Self-modifying encrypting virus.

Stealth Virus: A virus which hides its presence from the PC user and anti-virus programs, usually by trapping interrupt services.

.SYS: The extension given to system file names in MS-DOS. An example is the file CONFIG.SYS which sets up various configuration parameters for the operating system on power-up.

Trojan Horse: A computer program whose execution would result in undesired side effects, generally unanticipated by the user. The Trojan horse program may otherwise give the appearance of providing normal functionality.

TSR: Terminate and Stay Resident, a term used to describe an MS-DOS programs which remains in memory after being executed. A TSR can be re-activated either by a specific sequence of keystrokes, or at some specific time, or by some specific signal from an I/O port.

VDL: Virus Description Language is a Sophos proprietary language used to describe viruses. It has extensive facilities to cope with polymorphic viruses.

Virus: Sometimes explicitly referred to as a computer virus, a program which makes copies of itself in such a way as to 'infect' parts of the operating system and/or application programs. See boot-sector virus and parasitic virus.

Virus Signature: An identifier recognised by the virus as meaning 'this item is already infected, do not reinfect'. It can take different forms such as the text 'sURIV' at the beginning of the file, the size of the file divisible by a number or the number of seconds in the date stamp set to 62. Some viruses do not recognise their signatures correctly.

Worm: A program that distributes multiple copies of itself within a system or across a distributed system.