December 1990 ISSN 0956-9979

VIRUS BULLETIN

THE AUTHORITATIVE INTERNATIONAL PUBLICATION
ON COMPUTER VIRUS PREVENTION,
RECOGNITION AND REMOVAL

CONTENTS

EDITORIAL

s*LAN*ging Match

The war of words between the main protaganists in the current '*Novell* virus' debate has intensified in the last few weeks, with some fairly venomous exchanges being issued in the public domain.

The controversy began in July of this year when New York consultant Dr. Jon David released a report about a computer virus which he and associates claimed to have observed propagating on a *Novell* LAN. Dr. David said that the virus, a Jerusalem 'variant', bypassed NetWare server write-protection and even deleted protected server files. These effects, he says, were "conclusively demonstrated" to *Novell* representatives at *Novell's* New Jersey facility on July 11th.

Dr. David was subsequently quoted in stories in two U.S. magazines, *Network World* and *LAN Magazine*. The full story was offered to *VB* for publication, but without 'in house' verification of Dr. David's findings and having been denied a specimen copy of the virus for analysis, it was decided that the story should not be published. His account eventually appeared in the *Elsevier* journal *Computers & Security*.

On October 29th, Dr. David posted a message to *VIRUS-L* saying that he had received a note from the *Novell Corporate Counsel*. To quote Dr. David:

"*Seems that they've read some stuff they don't care for, stuff with my name (as a source, not author). Using phrases such as breach of contract and false and defamatory statements, it seems that, if I don't shut up, they're going to sue my butt off.*"

In an astonishing move which followed, Dr. David then solicited reports of *Novell* insecurity from *VIRUS-L* readers. Quoting Dr. David again: *"All potentially damaging (to Novell) reports will be verified, and once an appropriately nasty set of stuff (that can be reproduced at will) is assembled, I will pass it on to a highly respected security/virus expert for release to the public*".

In a statement released on *Internet* on November 6th, Mr. James Brown, senior software engineer at *Novell's* NetWare product division at Provo, Utah, denied the existence of both the virus and the alleged lawsuit.

Mr. Brown believes that Dr. David committed 'pilot error' with regard to NetWare access rights and directory handles. In his statement, there are numerous criticisms of Dr. David's credentials: explicitly, he is described as knowing ''absolutely nothing about NetWare'' while implicitly, his computer skills and virus knowledge are questioned. Mr. Brown claims that despite concerted efforts, *Novell* failed to reproduce Dr. David's results. The virus was found simply to hang infected workstations or cause unrecoverable network errors (symptoms indicative of the Jerusalem virus). Mr. Brown stated categorically that the virus could **not** write to files with Read, Open and Search (ROS) rights as Dr. David had claimed.

According to Mr. Brown "*Novell provided everything that Jon asked for and Jon spent his time complaining about what he said was "different" between the setup in Provo and the setup he originally used in Paramus* [New Jersey] *, even though we provided him with an identical workstation (down to the BIOS version and hard drive type and size)"*.

He continued: "*Novell spent quite a bit of money trying to reproduce Jon's results, and when we couldn't we flew Jon out at Novell's expense plus paid his consulting fees to have him reproduce his results. He was unable to reproduce any of his test results while here. Jon could only give us the name of one of the "many sites reporting this problem" (Jon's words) and that site agreed that the problem was operator error (they could not reproduce the test results that they had originally reported to Jon)*". The statement became quite acrimonious in places, referring to Dr. David as a "crackpot" and saying that *Novell* had no intention of sueing Dr. David because they believed he would eventually "self-destruct".

With *Novell* officials vehemently denying the existence of the virus and Dr. David equally persistent in his claims, it is difficult to ascertain the truth. Of the virus researchers and network specialists so far consulted by *VB*, the overwhelming majority incline towards *Novell's* version of events.

The issue is not whether a virus can propagate on a LAN, which is possible and happens occasionally, but whether a virus can circumvent correctly implemented network security.

VB has received no physical evidence whatsoever of a NetWare-specific virus and has no evidence (other than hearsay) of any virus circumventing NetWare security. Virus researchers who have actively tested the Jerusalem virus on a *Novell* LAN report results which concur broadly with those reported by *Novell*, not Dr. David.

In the interests of not perpetuating a myth, and with an eye to conducting matters scientifically, the virus currently held at Provo should be disassembled (if it has not already been) and its identity made public. Dr. David's apparent refusal to run this code through DEBUG (or *any* disassembler) is bewildering as is the reticence of everyone involved in testing this virus to reveal its exact identity. A proper comparison between the '*Novell* virus' and the Jerusalem virus would reveal any code sections (should such differences exist) included to subvert NetWare - and much more besides.

If, as is strongly suspected, this specimen turns out to be the simple Jerusalem virus, another myth can be laid to rest. However, in the unlikely event that it should turn out to be something more sinister, NetWare users have a right to know.

CASE STUDY

Ray Glath

Virus Propagation on Novell

At 7.00 am on October 12th, 1990, employees at an engineering company in a major mid-western U.S. city began their day by logging on to the firm's LAN. By 7.30 am, several workstations connected to their thirty terminal *Novell* network were experiencing the following symptoms:

- workstation processing slowed to a crawl.
- the infamous 'black hole' side-effect of the Jerusalem virus appeared on the left hand side of colour displays.

The system was immediately brought down by the supervisor and the investigation commenced. The priority was to restore the system to full operation as quickly as possible.

Vi-Spy (*VB, May 1990*) was used for diagnosis and it determined that the suspect program was the Jerusalem virus and that it was resident in the memory of several workstations. **The virus had spread to many program files on the local drives as well as on the server**. The decision was made to undertake a full restore from the most recent tape backup and then perform a further examination of all diskettes.

Normally this would hasten recovery. However, in this case further gremlins were encountered: while attempting to restore 8,000 files into a directory with a stated capacity of 10,000 files, there appeared suddenly a message that the number of allocated files had been exceeded. The restore aborted after only 5,000 files had been successfully loaded.

A new side-effect of the Jerusalem virus? No... simply an upgrade problem that hadn't come to light before. It appears that *Novell* updates the NetWare and the Shell separately. This client had upgraded both components to version 2.15c and the tape backup software (from another company) was not compatible with the new *Novell* shell. This caused the Restore program to create two entries for each file restored.

This problem was eventually resolved, the system was cleansed and normal operations resumed.

How Could It Happen?

A copy of the offending virus was sent to *RG Software Systems* for analysis. It turned out to be the same Jerusalem virus that we've encountered time and time again. **This was certainly not a virus specific to *Novell* networks.** Despite some recent reports of a '*Novell*-specific' virus, it should be pointed out that neither the Jerusalem virus or its many variants contain code which is intended to circumvent NetWare security.

How, then, did the server files become infected? Remember that Jerusalem becomes memory-resident when an infected program is executed and proceeds to infect every program run on the PC until the system is re-booted. (*Some virus listings, including the U.S. listing compiled by Patricia Hoffman, erroneously report that the Jerusalem virus is capable of surviving a warm re-boot (Ctrl-Alt-Del). Ed.*) **Consequently, the NET3.COM and IPX.COM programs, stored on local drives and used to log onto the server can become infected very quickly if they have 'write' access** (Under NetWare all programs have 'read/write' access by default).Then, as programs from the server are run on individual nodes, they become infected as well.

The following were some of the programs found to be infected on the server: LOGOUT, MAP, SYSCON, PCONSOLE, POSTMASTER (for e-mail) and SABER Menu.

What of security privileges on the server? This is the crux of the matter. In this instance, individual users had been granted *Open*, *Read*, *Create*, *Search*, *Modify Flags*, and *Write* privileges for **all** files on their assigned logical drives. This was done so that each user should have complete control within his or her 'world' and the privileges applied equally to both the NET3 and IPX programs.

The server infection resulted from a network configuration designed for flexibility rather than security.

While vendors who only market network security systems can offer more thorough security, *Novell's* NetWare does offer the most extensive security system available as part of a network package; file server security is provided for the login procedure, the allocation of trustee rights, directory rights and file attributes. The latter measure provides a substantial defence against virus propagation.

The original source of the virus appears to have been a demo disk for MultiMate containing the *Ashton-Tate* label. However, being a 'notched' diskette, the demo disk could have acquired the infection at any time. Since I have seen numerous virus infected demo disks, I should like to take the opportunity to encourage all software publishers to issue both demos and full product software on notchless diskettes.

Predictably, no-one could remember where the infected disk came from, nor where or when it was brought on site.

Networks present their own special set of headaches when a virus infection arrives. Not only does the infection have the potential to spread faster and farther, but the complexity inherent in network software and its associated 'add-ons' leaves the system prone to unkown effects should it be tampered with in any fashion, whether deliberately or unknowingly. LAN administrators should ensure that critical programs are properly protected from modification, either by the user or by a computer virus, by limiting file attributes.

Editor's Comment on the '*Novell*' Virus'

There has been recent speculation about the existence of a computer virus which contains code designed to circumvent NetWare security. In February 1990, *VB* published as 'reported only' a virus aimed at *Novell* LANs which was said to be capable of detroying the NetWare FAT on a server. No further evidence of this virus was forthcoming and there is some doubt that it ever existed.

In July of this year, *VB* received an extensive report from Dr. Jon David, a consultant based in Tappan, New York, which described a virus said to be capable of violating NetWare security. It was not possible to assess the accuracy of the report as no disassembly of the virus was undertaken concurrent to 'live' testing on a *Novell* LAN and no example was made available to *VB* for analysis.

Dr. David claims to have witnessed a virus similar to the ubiquitous Jerusalem virus propogating on a *Novell* LAN (2.15c) under test conditions at *Novell's* Paramus, New Jersey, facility on July 12th of this year. The most sensational aspects of the report included claims that the virus was observed to bypass standard NetWare security controls. Dr. David, with witnesses Greg Drusdow, president of *NetWare Users International* and Jay Nickson of *On-Disk Software*, New York, reported that the virus could:

- alter date-time stamps locally or on the server, *even if rights to do so had not been granted.*
- increase file lengths locally or on the server, *even if rights to do so had not been granted.*
- delete, on being triggered, any EXE or COM file invoked for execution before execution locally or on the server, *even if rights to do so had not been granted.*

Dr. David's report on this incident was subsequently published in *Computers & Security Journal, Volume 9, Number 7.*

A *CERT* advisory warning of the imminent impact of this virus was posted as a result of testing on July 12th 1990. This message, entitled 'LAN virus!!!', reiterated the virus' observed ability to subvert NetWare security and recommended that network supervisors advance the system date to 7/14/90. *Novell's* response appeared on *VIRUS-L* and *VALERT-L* on July 12th and it is repeated here.

NetWare Message
July 12, 1990

NetWare Users International (NUI) with the support of Novell has conducted tests that have concluded that a variant of a Jerusalem B computer virus has been discovered in at least one NetWare user site. The virus infects DOS executable files. In order to be exposed to the virus one must import an infected DOS program from the outside. NetWare files are not infected as they ship in the red box from Novell.

The virus operates as a TSR. Once an infected program is run at a DOS PC, the virus takes residence in the PC memory as a TSR. Subsequently, the virus, upon executing any other DOS program on the PC, will attempt to infect the disk resident copy of the program. The infection can occur on local drives as well as network drives provided the workstation has appropriate access rights to write and modify the file. [Editor's emphasis.] *Files on network operating systems other than NetWare could also be infected by this virus. Files infected with the virus will show an increase in size of about 1800 bytes.*

The real negative effects of the virus will not show until certain programmed dates. One confirmed date is July 13, 1990. There is a risk that an infected workstation will delete program images on disk that are requested for execution on that date. A "Bad command or file name" message will result. Under NetWare, the SALVAGE command will restore the file if executed properly. If infection is suspected, it is recommended that you advance the server system date at the close of the working day of July 12, 1990 to July 14, 1990. It is also recommended that professional assistance be consulted.

These facts and report were prompted by a report to NUI. Novell and NUI in their concern for responsible leadership felt it necessary to verify these conditions and notify its users so they could act accordingly.

Richard King
Vice President
Novell, Inc.

It is interesting that Novell did **not** report the virus as being capable of circumventing implemented security controls but merely as being capable of propagating in the *absense* of such controls; a conclusion entirely consistent with *VB's* own research. The case study (*on page 3*) is archetypal.

On September 20th, 1990, *VB* received a letter from Dr. Harold Highland, Editor-in-Chief of *Computers & Security* stating that the this apparently *Novell*-specific virus was simply the standard Jerusalem virus. In subsequent testing at *Novell's* headquarters at Provo, Utah, the '*Novell*' virus' (presumably the Jerusalem virus) reported ability to violate NetWare security was **not** observed nor could it be recreated.

With the evidence available to date, it thus appears that:

- the virus tested at Paramus, New Jersey, did **not** contain code specifically designed to subvert NetWare security.
- the common Jerusalem virus **can** infect the file server of a *Novell* LAN from a workstation and delete server files but only in the *absence* of NetWare security controls. Not a startling revelation, but a reminder to network administrators to implement and maintain available controls. (*Ed.*)

TECHNICAL NOTES

Anti-Anti-Virus Programs

Several programs have emerged recently which aim to attack well-known anti-virus programs in some way or another. The attacks are directed by different means, including the development of "Trojanised" versions of popular anti-virus packages. The changes made to the programs may involve a simple virus infection, or more elaborate destructive code insertion.

In order to avoid malicious modifications, many producers of anti-virus software take some precautions. These include adding a "self-test" to their programs, in order to verify that they have not been changed. A checksum or information about the length of the programs may also be included in the documentation.

Those precautions are of little use against a determined attacker, as the "self-test" routines can be patched and the documentation modified accordingly.

In addition to direct attacks where the anti-virus software is modified, several indirect forms of attack are possible. An interesting example surfaced recently in Bulgaria, in the form of a program which examines recent versions of the SCAN program from *McAfee Associates* and decrypts the hexadecimal search patterns therein to produce a complete listing. This listing is obviously of benefit to hackers who intend to modify an existing virus, perhaps by a minor re-ordering of instructions. With access to the search patterns incorporated, it is easy to produce a virus variant which cannot be detected by any scanning program singled out for attack.

The Long-Lost Agiplan

A virus which corresponds exactly to the description of the "lost" Agiplan virus has been reported in South-Africa, causing considerable surprise to many virus researchers.

A search string for this virus has been available for 18 months, although it was not published in the *Virus Bulletin* until February 1990. Only a single infection had been reported anywhere in the world until the sample arrived from South Africa. This is particularly interesting, as the sole previous report was in a German company more than two years ago.

It is remarkable if the virus has managed to spread from Germany to South Africa without being reported during the last two years. However, there may be another explanation.

The virus might not be identical to the "original" Agiplan virus, but instead written by someone with access to the *VB* Agiplan search pattern and a description of the virus.

It is not known whether this is the case, but one finding supports this theory - a section of "garbage" bytes at the end of the virus, as if it had been padded to match the length of the reported Agiplan virus.

The Search for the UVD

A Univeral Virus Detector (UVD) is a program which can perform static or dynamic analysis of programs and determine with 100 percent certainty whether the programs contain a virus or not. A truly interesting program, but could it actually be developed?

It is relatively easy to show that this type of program is theoretically impossible in the case of *Turing* machines with infinite memory, but what is the case with finite real-world computers?

In fact, such a development appears to be an impracticable, if not impossible, proposition. The UVD would have to simulate the execution of each program tested, going through every possible execution path. The simulated program would then either terminate or get stuck in an infinite loop. By far the most difficult task would be for the program to determine when a virus infection had *actually* ocurred.

This is partially because of the lack of a precise definition of the term "virus", which can be illustrated by studying the original definition by Dr. Fred Cohen, from his dissertation *Computer Viruses: Theory and Experiments*:

> "*We define a computer virus as a program that can infect other programs by modifying them to include a slightly altered copy of itself.*"

It must be noted that "programs" also include boot sectors, INITs and all other forms of executable code. Also, the term "include" must also cover cases like the 405 virus, which overwrite the victim file, and may destroy it completely.

(Note: Dr. Cohen's definition of a virus includes the AIDS Trojan (*VB*, January 1990) which requested that the *user* actively copy the program to another machine and the DOS DISKCOPY function. His reasoning was published in *VB*, May 1990, p.7.)

The precise definition of the term 'computer virus' is proving elusive: Dr. Cohen's definition, for example, does not appear to encompass the so-called 'companion' viruses at all. In fact, there is no consensus that these programs are computer viruses! They do not appear to conform with any published definitions.

Nor does his definition say that the program has to perform like a virus every time it is run, only that it must be able to demonstrate the essential 'viral criteria' of modifying programs to include itself in some form.

Although Dr. Cohen's definition appears, initially, to be helpful, it has proven to be of little value in developing generic virus defence. Consider the following pseudo-code:

Program P1

```
Display "This is a copy utility"
Display "Name of input file?"
Input In-File
Display "Name of output file?"
Input Out-File
Copy In-File to Out-File
End
```

Is P1 a virus? According to Dr.Cohen's definition, yes. If it is given the name of itself as In-File and the name of some other existing program as Out-File, it will behave just like any other destructive overwriting virus. Since P1 is capable of placing a copy of itself within another program, it is (according to this definition) a virus.

In fact, Dr. Cohen's definition is so extensive that it will classify most operating systems as viruses, since they can easily "infect" other programs in the same way - provided that the right sequence of commands is entered. Any compiler used to compile the source to itself would also qualify.

Clockwise

David Chess at the *IBM Thomas J. Watson Research Center*, Yorktown Heights, New York, has pointed out a minor error in the dissection of the Flip virus in the September edition of *VB*. The virus will activate between 10:00 and 10:59 on the second day of the month, not between 16:00 and 16:59.

MAC THREATS

ZUC 2

A new variant of the ZUC virus has been detected in Italy on 26th November 1990. This strain appears to be functionally similar to ZUC 1. The original ZUC virus carries signatures of well known anti-virus products and bypasses protection INITs by utilising stored ROM addresses for key functions (*see VB, October 1990, p. 6*).

Public domain and commercial anti-virus software is being upgraded to detect and disinfect this new strain. Available search strings and product updates are published here.

SAM 2.0

SAM 2.0 will detect the modification of the application code resources by the ZUC 2 virus in *standard*, *advanced* and *custom* modes. The following virus definition can be added using the SAM virus clinic function:

Virus Name: ZUC 2
Resource Type: CODE
Resource ID: 1
Resource Size: Any
Search String: 7002A2604E752014A0552240
Search Offset: Any

Virus Detective

A ZUC search string for Virus Detective has been released by Jeff Schulman. This string detects both versions of ZUC.

Filetype=APPL & Resource CODE & ID=1 & WData A746*A038#31E*A033 ; for finding ZUC.Virus 1 & 2

A New Anti Strain

A new variant of the Anti virus has also appeared. The strain is similar in functionality to the original Anti virus. Anti was the first Macintosh virus which does not add new resources on infection, the virus instead appends its code to the CODE 1 resource of the application being infected. When an infected application is run, the virus installs istself in the system heap and thereafter infects any application which is launched or has its resource fork opened.Unlike other Macintosh viruses, Anti does not infect the system file and only becomes active in memory when an infected application is run. Anti does not spread under Multifinder. Detection details appear below.

SAM 2.0

SAM 2.0 will detect the virus in the *standard*, *advanced* and *custom* modes. The following specific recognition string has been made available:

Virus Name: ANTI
Resource Type: CODE
Resource ID: 1
Resource Size: Any
Search String: 000A317CFFFF000CA033303C0997A146
Search Offset: -886 (For early SAM versions use 'Any')

Disinfectant 2.3

A new release of Disinfectant (release 2.4) is now available which detects both the ZUC 2 virus and new Anti strain.

(*A list of public domain and shareware Macintosh anti-viral products appears on page 7.*)

IBM PC VIRUSES

Amendments and updates to the *Virus Bulletin Table of Known IBM PC Viruses* as of December 4th 1990. The full table will be published in *VB*, January 1991.

Hexadecimal patterns can be used to detect the presence of the virus by the "search" routine of a disk utility program or, preferably, a dedicated scanning program. (For further information see *VB*, August 1990, pp.7-16).

Diskjeb - CER: A disk-corrupting virus with an infective length of 1435 bytes (COM) and 1419 bytes (EXE). Only infects COM files longer than 1000 bytes and EXE files longer than 1024 bytes. In October, November and December disk writes will be intercepted and corrupted. A possible mutation of the Tenbyte virus.

```
Diskjeb 5351 061E 9C8C C88E D8E8 5D00 803E 4903 ;
Offset 4E8
```

Freeze - CR: A 1024 byte virus which makes the computer "hang" at random intervals.

```
Freeze 4545 5A45 B8EF EFCD 213D FEFE B800 0074 ;
Offset 002
```

Saddam - CR: This topical virus was uploaded to numerous bulletin boards in Israel in September. It is related to the Stupid/Do Nothing virus from Israel.Virus extends the file length by 917 to 924 bytes. Displays the following string (which is stored encrypted):

```
HEY SADAM
LEAVE QUEIT BEFORE I COME
```

after 8 requests for INT 21H. Resides in the area of memory not labelled as used, so large programs will overwrite it.

```
Saddam BB00 0153 5052 1E1E B800 008E D8A1 1304 ;
Offset 010
```

Spanish Telecom - MCER: This encrypted multi-partite virus contains a message by "Grupo Holokausto" demanding "lower telephone tariffs, more services". It proclaims to be an "Anti-CTNE" virus where CTNE is *Compania Telefonica Nacional Espana*. A message in English states that the virus was programmed in Barcelona, Spain.

```
Spanish Telecom 8BDC 33FF FA57 B2
```

Turku - CER: A Finnish virus, 1332 bytes long, but adds 1472 bytes to EXE files. Reported to intefere with keyboard operations. Awaiting disassembly.

```
Turku 0175 118C C0BB 0001 B910 00BE C005 BF00 ;
Offset 053
```

REPORTED ONLY

Guppy - CR: A small, 152 byte virus. Many infected programs will fail to execute.

Park ESS - CER: A new variant of Jerusalem.

WORLDWIDE

Dutch Disaster

16,000 Cascade (1704) infected cover disks were distributed with the the November edition of the Dutch publication *PC-WORLD Benelux* on November 9th, 1990. The magazine circulates in Holland, Belgium and Luxembourg and is part of the international *IDG* stable of computer titles.

Following the lodging of an official complaint by *IDG*, the Dutch police's fraud squad are investigating the matter. Pivotal to the investigation is the need to establish whether the master diskette or the duplication machine were infected by the virus. It is understood that no formal anti-virus procedures were adopted by either the publisher or the duplication facility prior to this incident. Unchecked, the disks were distributed to subscribers and news-stands throughout Holland. However, the distribution company contracted to ship the magazine within Belgium said that it was still in the process of packing and subscriber labelling when the infection was confirmed and the magazine and disk were withdrawn.

Managing editor, Koos Delange, distributed formal warnings by letter to subscribers and had 4,000 copies of the magazine and cover disk withdrawn from Dutch news-stands. Mr. Delange also appeared on Dutch national television to warn PC users. According to a report in UK's *The Independent*, *IDG Communications* subsequently spent approximately £40,000 on television and newspaper advertisements alerting Benelux readers to the virus-contaminated disk. The company also distributed 11,000 copies of virus disinfection software.

Cascade is common in Western Europe. After infection the virus becomes memory-resident and infects every .COM file including COMMAND.COM. The 1704 byte version is highly infectious but relatively innocuous. Most disinfection programs will remove it. Alternatively, infected .COM files can be deleted and replaced with clean copies of the master software. Given the longevity and frequency of this virus, all the major virus-scanners should detect it.

The incident highlights yet again the absolute necessity for publishers to verify the integrity of cover disk software. In the interests of both the publisher and readers, this validation must take presidence over shipping 'deadlines'. Master disks should be scanned in a known 'clean' environment, and samples from the production run checked before despatch.

The incident also demonstrates the continuing need for end-users to check **all** incoming software (and data diskettes), irrespective of source, for viruses using at least one virus scanning program. The establishment of a *Software Quality Assurance Section* (*VB, May 1990, p.5*) will assist in this process.

PUBLIC DOMAIN ANTI-VIRUS SOFTWARE

The following is a list of current releases of public domain and shareware anti-virus products. In the United Kingdom they are available by electronic mail from JANET <Info-server@cs.hw.ac.uk> or from the national public domain software archive at *Lancaster University* (0524 63414, 0524 67671, 0524 67754, 0524 62423 supporting V21/V23, 0524 381819 supporting V22).

These programs are distributed internationally and all are actively supported by their authors. Prospective users are reminded of the need for reasonable caution when downloading and running BBS public domain software and shareware. The list is provided for information purposes only as the quality and integrity of the software is not attested by *Virus Bulletin*.

Product	Author	Release	Description
Apple Macintosh			
Antipan	Michael Hamel	1.5	nVIR scanner and disinfectant
Disinfectant	John Norstad	2.4	Scanner, disinfectant and monitor
GateKeeper	Chris Johnson	1.1.1	Resident interrupt monitor
GateKeeper Aid	Chris Johnson	1.1.1	Implied loader virus trap
Repair	Steve Brecher	1.5	nVIR disinfection utility
Virus Detective	Jeff Schulman	4.0.3c	Desk accessory scanner
VirusRX	Apple	1.6	General scanner and disinfectant
IBM PC			
Alert	Ted Emigh	1.4	Checksum utility (window driven)
AVS	Tjark Averbach	2.20	Scanner
Checkup	Rich Levin	3.9	Checksum utility
Cleanup	McAfee Associates	5.1 v67	Disinfectant
Detective	PC Solutions	3.1	Checksum utility
Fprot	Fridrik Skulason	1.13	Scanner, disinfectant and monitor
Flu-Shot +	Ross Greenberg	1.81	Checksum and monitor
Fshield work)	McAfee Associates	1.5	Monitor (permits development
Immune	Yuval Rakavy	9.00	Jerusalem virus monitor
Netscan	McAfee Associates	v67	Network scanner
Secure	Mark Washburn	v2.09	Monitor (general)
Sentry	McAfee Associates	v2.0	System area checksum
Unvir	Yuval Rakavy	9.02	Scanner and disinfectant
Virus central	Alejandro Abello	1.03	Virus scan window interface
Viruscan	McAfee Associates	5.3 v67c	Scanner
Virstop	Tacoma Software	1.05	Monitor (virus specific)
Vshield	McAfee Associates	67-B	Monitor (virus specific)
Vtac	Randolph Beck	47	Monitor (general)
Atari ST			
Vkiller	George Woodside	3.11	Scaner and disinfectant
Commodore Amiga			
VirusX	Steve Tibbett	4.0.1	Scanner and disinfectant

FROM THE FIELD

Bulgaria - An Eye-Witness Account

Early in November, UK consultant Bryan Clough travelled to Bulgaria to investigate the notorious Bulgarian 'virus factory'.

During the course of his four day fact-finding trip, Clough met a number computer virus writers. He also talked to Vesselin Bontchev at the Bulgarian Academy of Sciences at Sofia. Clough describes Bontchev as a 'genuine missionary' trying to educate the computer literate and guide them towards progressive programming. However, Bontchev's efforts appear to be in vain; Clough returned to the United Kingdom with more than a hundred different computer viruses of which he reports more than half as being currently undetected by Western virus-specific software.

Mr. Clough's report makes depressing reading, particularly for the developers of anti-virus software, who must struggle to keep pace with developments. Vesselin Bontchev, who recently attended a major conference on computer viruses in Kiev, reports to VB that viruses are also rife in the Soviet Union and that virus-writing in the USSR is just as prolific as in his native country.

The Dark Country

Bryan Clough

Bulgaria is financially bankrupt and it shows.

On the first anniversary of the November 10th revolution that ended the 35-year rule of President Todor Zhivkov, the country is being torn apart by protest meetings, riots and threatened strikes. It also has an ailing Chernobyl-type nuclear reactor to contend with.

Electricity is supplied for two hours in every four and with no replacement light-bulbs in the shops, the country gets darker by the day. Optimistically, there is a 20-year wait for a house or a new car, but a 200 mile round trip by train, first class, costs less than £1. The average wage equates to £20 per month but there is virtually nothing available to purchase.

Breakfast at my hotel in Sofia comprised undrinkable coffee, hard bread, greasy butter and thin, tasteless jam. I settled for a bottle of water, provided at extra cost.

The telephone system is sixty years old and there are no telephone directories, so even though I had names of people I wanted to meet, it was impossible to make contact unless I already had their numbers.

The people, however, are marvellous. Even with the gauge hovering on 'Empty', they will use their last drop of petrol to provide their visitor with transport. Petrol is rationed to 30 litres per month - if you can find it and if you are also prepared to queue for hours on end.

They use the Cyrillic alphabet and their language is closest to Serbo-Croat, followed by Russian, so communication for the English-speaking traveller is often difficult. I found myself trying to hold conversations in German, French, Italian and also Spanish which is surprisingly popular because of work assignments in Cuba.

Of the 9 million population, approximately 130,000 have emigrated during the past twelve months; many others would like to follow. However, exit visa forms (which are nominally free) can now only be obtained on the black market.

At the *State Printing House*, I opened conversation with Salsa Halacheva, their System Engineer, by saying:

"I believe that you are short of paper".

She replied sadly: "We are short of *everything*".

The Virus Factory

According to Vesselin Bontchev, a leading virus researcher at the *Bulgarian Academy of Sciences* in Sofia, one 'commodity' is certainly **not** in short supply - computer viruses.

Bontchev has identified over thirty distinct types and over 100 strains that have been produced locally. New virus specimens continue to appear at the rate of one every week. He personally receives around 100 telephone calls each week from users suspecting a virus problem and about seven percent of the calls are confirmed as viruses.

During my four-day stay, two new viruses appeared: a fifth version of Number of the Beast that had been found in a computer club - a favoured way of disseminating viruses - and a completely new virus which was presented to me by its author, a 23-year old army lieutenant. He had written it in PASCAL because he had heard that it was possible to use a high-level language to write a virus and he had wanted to prove this for himself.

His two earlier viruses had been written when he was at university to embarrass his professor. The professor never even suspected that the 'jokes' were caused by virus infection and the viruses quickly migrated outside the university.

Bulgaria attempted modernisation in 1984 by setting up a computer manufacturing facility to make Apple II and IBM XT clones. There is no patent or copyright law governing this manufacturing process. Software was pirated in similar manner and no copy-protection scheme has been found which can defeat the Bulgarian pirates. Even dongles can be circum-

vented without the need for a sample of the dongle itself. The version of PKZIP that they use has DES encryption - the version prohibited from export outside the United States. So much for US security regulations.

Dimitri Vavou, who used to work at the computer plant, estimated that there are now around 250,000 computers in Bulgaria, half in education and the remainder in organisations. Very few individuals can afford their own computer.

Without copyright protection, there is no incentive to develop commercial software, so those with programming skills sometimes turn their hands to other things. There were those who chose to modify the Vienna virus which they called VHP-648 (the name used in the Cuban anti-virus program that they had obtained) and, in December 1988, Old Yankee - the first true Bulgarian virus - appeared. The author, Vladimir B, then lost interest in writing viruses: the challenge had been met.

His work, however, served to inspire 'TP' who started writing his virus on December 12th 1988 and by January 6th 1989 was on Version 18 which, according to his source listing:

```
; Don't beeps
; Plays Ynkiedudle on soft reset
```

TP's 'development cycle' involved producing a virus and then developing a vaccine for it. As would be expected, his viruses are far more widely disseminated than his vaccines. As well as this series of viruses (known variously as Vacsina, Yankee Doodle or TP), he has also written the TPWorm and another (unfinished) virus called VBN. He is said to have taken great care not to inflict damage and stopped writing viruses some 12 months ago. Reportedly, he is now hoping to find employment in the West.

"Some of the recent viruses have not been released in Bulgaria but uploaded directly to Bulletin Boards in the UK, Norway, The Netherlands and Greece 'to see how quickly they return'."

PD who lives in Plovdiv, the second largest town, has written several viruses including those called Anti-Pascal, Tiny, Terror, INT13, V1600, Nina and the only Bulgarian produced boot sector virus, XBOOT. The Tiny viruses were, as had been supposed, merely exercises in optimisation.

He has stopped writing viruses because "It is now too easy, there is no more challenge".

Two virus writers in Varna, VP and SK, (authors of the Shake, Dir and MG viruses) reckon that they could produce an even smaller virus than PD's, possibly only 50 bytes, provided that it is kept very simple.

The programmer calling himself 'Dark Avenger' started writing his first virus in September 1988 but it was March 1989 before he had a version to unleash. His viruses are often deliberately malicious, highly infectious and characteristically corrupt the File AllocationTable. He revels in the knowledge that that they have spread to all the countries in Eastern Europe including the Soviet Union as well as the United Kingdom, Germany, The Netherlands, the United States, Taiwan and, reportedly, even Mongolia.

According to Bontchev, the two Dark Avenger viruses known as Evil and Phoenix are not only viruses but they also Trojanise an EXE file on an infected system. The Trojanised EXE keeps checking for the virus and when it is removed, the Trojan destroys the hard disk (*see page 14*). It is also believed that his Anthrax virus (which infects program files and boot sectors) can be resuscitated by another of the Dark Avenger viruses, V2100, if not fully removed from an infected system. (*See Technical Notes, VB, November 1990*).

Some of his recent viruses have not been released in Bulgaria but uploaded directly to Bulletin Boards in the UK, Norway, The Netherlands and Greece "to see how quickly they return". (*This phenomenon was first reported by Dr. Alan Solomon who claims that an individual calling himself 'Dark Avenger' uploaded virus source code to S & S Ltd's BBS in the UK. The source code listings were for Dark Avenger, Yankee Doodle and V1024 (aka Nomenklatura). Ed.*)

The Nomenklatura virus (*see pp*. 18-19) which has recently appeared in the UK was almost certainly transmitted directly by modem. It has not yet been seen within Bulgaria, but the name is Bulgarian and it shares similarities with three other Dark Avenger viruses (Evil, Phoenix and V800).

The Dark Avenger writer has deliberately subverted two anti-virus programs. One of these works satisfactorily for most of the time but under certain conditions, some versions also release one of his viruses. The other, a version of John McAfee's *ViruScan* program, now not only contains a virus but its self-checking feature has been modified. Dark Avenger has also embedded another virus in an AFD (Advanced Full Screen Debug) program.

The so-called 'virus factory' has no formal organisation and the virus writers are everywhere including Sofia, Plovdiv, Varna and Svishtov. They help and liaise with each other, exchanging virus code (Dark Avenger, Number of the Beast and Phoenix are among the viruses that are known to have been exchanged), copies of about thirty anti-virus offerings (including *Dr.*

Solomon's Anti-Virus Toolkit), undocumented features of DOS and the *Norton Guide to DOS Interrupts*.

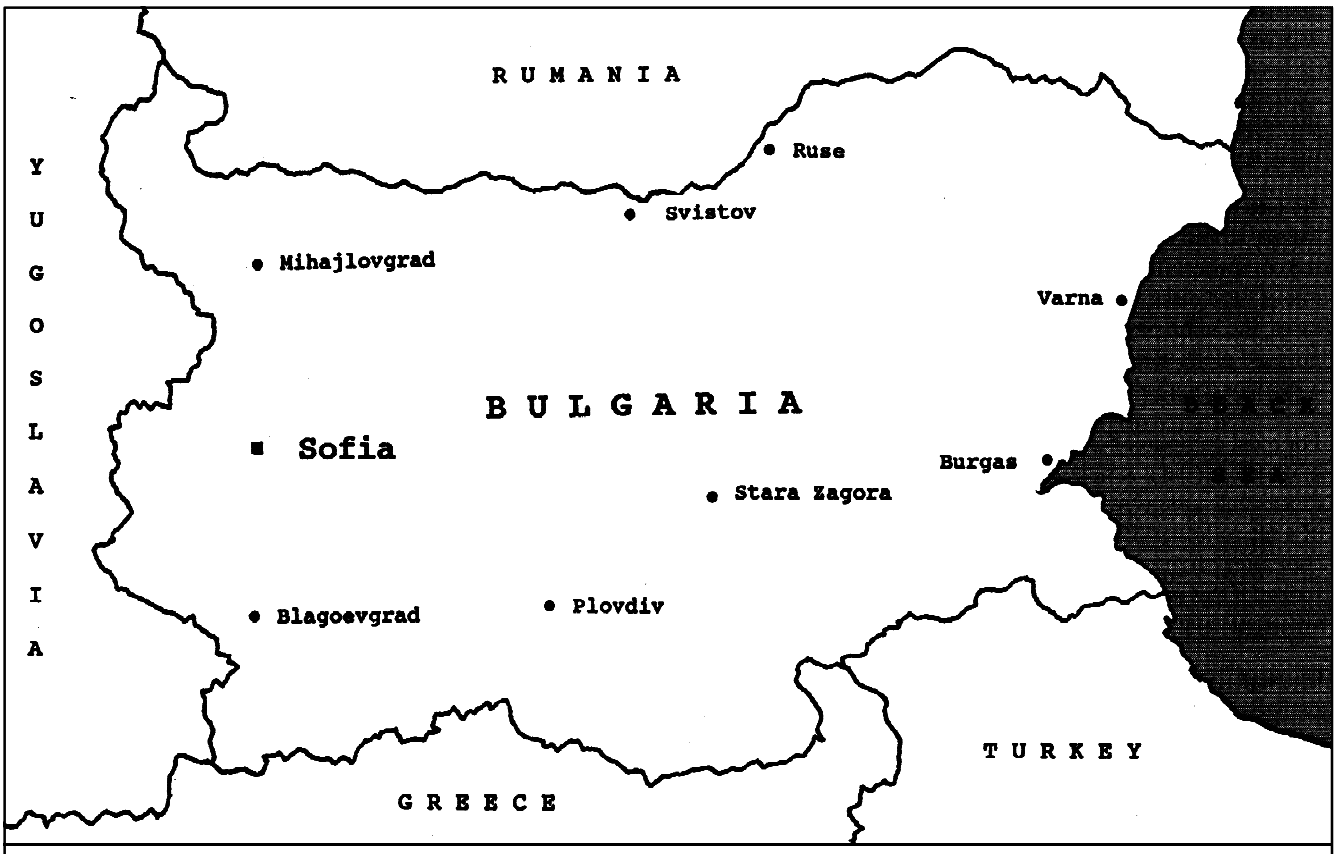
Bulgaria undoubtedly has a valuable and an under-utilised resource in its software development capability but, according to Nikolay Karinkov, President of *Ten Plus Software*, the virus writers have seriously damaged Bulgaria's chances of selling software abroad. He is always being told: "You can only write viruses in Bulgaria".

Within Bulgaria there is considerable concern about this reputation but they ask:"What can we do? There is no law against writing viruses and, even if there were, how could we find proof?"

In addition to everything else, many observers are now worried about the prospect of having to contend with Russian viruses. It is believed that the Russians may be using viruses for software copy-protection and their reportedly 'clever' LoveChild virus may have been produced for this purpose. It has a counter which decrements on certain DOS functions and, at zero, it destroys the hard disk. The text message: "LoveChild in reward for software sealing.." supports this view if "sealing" is a misspelling of "stealing".

The Russians who have just held a week-long virus conference in Kiev are also developing anti-virus measures. Among other things, they have reverse-engineered John McAfee's *ViruScan* program and published his identification strings. While this may possibly help researchers, it will also assist virus writers in circumventing this widely used tool.

Meanwhile, Bulgaria is heading for a long, hard winter and every Bulgarian I met believed that things can only get worse. Tackling the virus-writers is a low priority for the government of a country beset by social disorder and facing imminent economic collapse.



National crisis - the Bulgarian 'virus factory' is a virtual irrelevance compared to the economic and humanitarian disaster that has hit Bulgaria. A national debt of $10 billion and a moritorium on creditor repayments has effectively denied the import of the most basic commodities. Food, energy and particularly medicine is in desperately short supply. The resignation in November of Prime Minister Andrei Lukanov, amidst rioting and looting on the streets of Sofia, was a timely reminder of the political and economic turmoil that besets the country. However, the local computer virus problem will inevitably impact on the long-term prospects of the Bulgarian software industry.

FOR MANAGEMENT

Dr. Keith Jackson

PC Security Part II - Backups

This article explains what backups are, why computer users should take backup copies, and what part backups can play in helping to mitigate any deleterious effects caused by viruses, other malicious programs, hardware failure, malicious or inadvertent human action(s). As backup is such an important topic, I make no apologies for explaining such details in very simple terms.

Computer programs and the data processed by these programs, are usually stored on disk in a file. A file is merely a series of items of information collected together in a form that can be manipulated by the computer as a single unit. A backup of a file is a copy of the file which is identical in every way with the original.

Although this explains what a backup is, it does not address the reasons why taking backups is an essential practice, or explain what is required to create a backup copy.

All files used on a computer (whether programs or data) are usually stored on magnetic media which can be accessed by the computer. In most computers this comprises a judicious mix of hard disks, floppy disks and magnetic tapes. In all these cases, the same physical principle is involved: a special-purpose piece of hardware (the disk drive and its associated controller chips) can both *read* and *write* information.

For most computer systems, accessing information stored on magnetic media is simple for any program to carry out, and in most small computers, programs can manipulate the content of a file without difficulty. As a computer virus is merely a computer program, it too can manipulate files stored on disk. Indeed if this were not the case, viruses would have a hard time replicating and spreading to other computers.

Damage to files can either be *physical* (failure of a disk drive, change of magnetic media characteristics, or physical abuse of any kind), or *logical* (erasing or altering parts of selected files). **Under many circumstances the damage is irreparable, and this is where backup copies are commonly required. It is irrelevant what caused the damage to a file if a backup copy can be accessed to retrieve the original information**.

Having explained what a backup copy is, what methods of taking backups are most useful?

The answer to this question depends in part on what type of computer system is being used. By and large, business users of mainframes and the larger minicomputers have a built-in solution: it is the responsibility of the technical staff who operate the computer to see that regular backups are taken and to make arrangements for these backups to be accessed as necessary. In short backups are somebody else's problem.

The luxury of leaving the responsibility of maintaining backups to a third party is not available to the users of small computers of any type.

Most personal computers use a hard disk as primary storage and I shall consider in detail the situation where the user of one or more small computers has to maintain backups for the hard disk on each computer. Many methods of taking complete backup copies of hard disks are available. I shall consider them in turn.

Floppy Disks

All of the information on a hard disk can be copied to a series of floppy disks. The operating system used by most PCs comes with two built in commands (BACKUP and RESTORE) which facilitate just such an operation. **However, these MS-DOS commands will only permit file restoration to the exact subdirectory from which the backup was originally taken** Using backups to restore files on a different machine is often either impossible, or involves creating a fake directory, reloading the files, then copying to the desired location.

BACKUP and RESTORE are also specific to the version of the MS-DOS operating system in use, so a backup made using MS-DOS version 2.11 cannot easily be restored using version 3.00 of the same operating system. (*This problem has been rectified in MS-DOS v3.20 and later. Ed.*)

In short, if you intend to backup to floppy disks, purchase one of the many software packages which offer more comprehensive backup facilities (*see below*). **Do not use the built-in MS-DOS commands unless you are absolutely forced to**

Floppy disks are very cheap (a 'throw-away' medium), very suitable for taking multiple copies, and a floppy disk drive is provided with almost every PC. Unless backups are enormous, or particularly sensitive, it is difficult to justify major expenditure on any special purpose backup hardware. **Floppy disks storage is, however, prone to corruption in the presence of even weak electromagnetic fields or physical damage in the event of incorrect handling**

Substitute Hard Disks

The phrase 'substitute' covers a multitude of methods of introducing a *second* hard disk to a computer for the purposes of taking a complete copy of the original hard disk.

For many years Bernoulli Boxes have been available which provide a removable disk cartridge of roughly the same

capacity as the hard disk. The cartridges can be changed as desired and can be used to store a complete copy of the primary hard disk. In recent years, many PC manufacturers have expanded this idea to provide completely *removable* hard disks. These are now manufactured as very robust plug-in units and journalists have even been encouraged to review their performance after throwing the removable part down a flight of stairs! It should be borne in mind that to create a backup copy, the computer must be able to access *two* hard disks simultaneously.

Several manufacturers sell computers which offer hard disk 'mirroring', where all data written to the primary drive simultaneously updates another hard disk. **While this technique provides excellent backup coverage in the case of hardware failure, it does not prevent data being actively deleted (or modified) either by a program, or by human intervention**. In recent months, the *Opus* Datasafe, an 80286 machine incorporating two 60 Mb hard disks has broken new ground in offering affordable hard disk mirroring.

Tapes

Backup copies of a hard disk are often made using a device called a tape streamer. This is special purpose hardware which uses magnetic tape cartridges. The cartridges are relatively cheap and can each store many hundreds of Megabytes of data. Tape streamers offer good value for money when backups have to be made for several computers or file servers. To minimise the investment in hardware interface boards, portable tape streamers are available which plug into an I/O port. Tape streamers are not the most reliable of devices. **If tapes are used, then testing that the newly created backup tape can actually be restored is essential**. There have been numerous instances of backup copies that cannot be read on *any* machine other than the one on which it was created. Floppy disks suffer from this problem to a rather lesser degree as the floppy disk drive is used more frequently with disks from several sources.

Optical Disks

Write Once Read Many (WORM) drives use a laser to make tiny holes in the recording substrate on a special type of disk. This method of storing data is immune to magnetic interference and physical shock and can offer capacities in excess of 400 Mbytes on a single WORM cartridge. **However, the action of making irreversible changes to the disk surface means that any backup copy is permanent and the disk cannot be reused**. WORM drives are relatively expensive and suffer from a lack of standardisation between manufacturers. Erasable optical drives are now becoming available, but these are more expensive.

Optical disks are suitable for archiving very large volumes of data, but unless prices fall very steeply in the near future, they will probably remain too expensive for routine backups.

Procedures

It is absolutely essential to evolve working practices which control and enforce backup procedures and which guide restoration when anything goes wrong. To this end, always take backups as part of a pre-determined plan.

One of the best methods is to backup the complete content of a hard disk at regular intervals, and take frequent (ideally daily) backups of any files that have changed since the last complete backup was taken. This presupposes that the backup software is capable of creating various types of backup (*see below*). **The newly created backups must be tested frequently so that they are known to restore correctly**. If this is not done, a failure to restore at the worst possible moment is almost guaranteed. Any halfway decent backup procedures should include regular verification.

Backups should not be kept near to the originating computer. If they are nearby, and a disastrous mistake is made, then it is all too tempting to reach for the backups and repeat the same mistake, while in the case of physical damage to the computer (fire, flood etc.) the backups are also likely to be damaged. The simplest backup of all for a specific project is to use three floppy disk copies. One for immediate use, one for use if the first backup fails, and the third copy stored off-site for use only if all else fails. This method only works for small amounts of data.

Software

What facilities should a backup software package offer?

It is important to be able to write to **all** of the above mentioned backup hardware, i.e. floppy disks, tapes, Bernoulli cartridges, optical hard disks, and substitute hard disks - *anything* with which the operating system can communicate.

Error correction and data compression should be available to provide a choice between minimising the backup storage space required, and minimising the time required to perform the backup. Backups should be capable of being made for *any* specified set of files, from *any* chosen combination of subdirectories and from *any* range of date/time on the file.

Especially important are facilities which allow backups of only those files that have been changed since the last complete backup (known as a *differential* backup), files that have changed since the last backup of any type (known as an *incremental* backup), and a copy facility (repeat the last backup, but don't alter any of the backup markers).

There is no space within this article to provide a review of the software packages that are available for PCs. The list on the following page (which is by no means definitive) gives some idea of the cost of the commonly available packages, and their respective developers. Similar packages are available for *Macintosh* computers.

Package	Supplier	Price
BACK-IT	Gazelle Systems	£85
Copy II PC	Central Point Software	£30
DOSTAR	New Generation	£190
Fastback Plus	Fifth Generation Systems	£125
Intelligent Backup	Sterling Software	£99
Norton Backup	Symantec	£99
PC Fullbak	West Lake Data	£65
Shadow	Knowledge Dynamics	£195
SitBack	SitBack Technologies	£65

Common backup software packages. Prices are approximately UK *Recommended Retail Prices* and range enormously.

Extremely good backup facilities are often provided with general 'utility' programs. Two examples are:

Package	Supplier	Price
PC Tools Deluxe	Central Point Software	£85
Mace Utilities	Fifth Generation Systems	£95

(*Trademarks are acknowledged*)

An Example Procedure

Apart from discussing in general terms the backup methods available, you may wonder what backup methods do I actually use? Bear in mind that my two major computer tasks are software development and the writing of books and technical articles. My backup methods are tailored accordingly, and I cannot really justify the purchase of special purpose hardware to back up the relatively small (40 Mbyte hard disks which are about half full at any time) amount of information involved.

I always take floppy disk copies of all current projects (software source files, binary files and text files) at daily intervals. Two local copies are updated each day, and an off-site copy is updated at suitable intervals during the project lifetime. Each hard disk is backed up to a set of floppy disks using Fastback Plus (version 2). Three complete sets of floppy disks are maintained for each hard disk by updating two of them alternately, and depositing one set off-site.

Using this scheme, the last backup set is always in reserve while a new backup is being generated and an off-site set is always available if things go awry. Fastback Plus is capable of generating a 'differential backup', which includes only the files which have been altered since the last complete hard disk backup. Differential backups are continued on a daily basis until the data will no longer fit on a single floppy disk, at which point the complete hard disk is backed up once more. The daily differential backup process then restarts.

Using data compression, a complete hard disk backup can be taken using only a single box of 1.44 Mbyte floppy disks. I use this as a marker for the hard disk being too full. If a complete backup will not fit within a single box of floppy disks, then either a new backup method is required, or the hard disk should be 'pruned'.

I have used Fastback Plus and the above described backup procedure for some years now. When one of my hard disks fails (this is inevitable), I think that I am prepared for it. While writing reviews of some of the less well developed security products (not, I hasten to add any product reviewed in *VB*) I have sometimes had to low level format a hard disk after the security product had run amok. So far the restoration of a hard disk from a backup copy using the above techniques has always worked.

Although I cannot claim to have surveyed all of the available software, I have used many different backup and restore programs, and in my opinion the best such program available for the PC is Fastback Plus from *Fifth Generation Systems*, which is sold by most software retailers and is well worth the modest outlay.

A Cautionary Tale

If special backup hardware is used, then it is vitally important to use a backup system from a manufacturer who is likely to remain in business I used to manage a small network of PCs primarily used for software development. The manufacturer of the hard disk(s), and of the tape streamer, had just gone out of business in the USA, and soon after this, the system backups began to display frequent errors on many of the backup tapes. The error was almost certainly due to an intermittent hardware fault in the one and only tape streamer that we possessed. Given the lack of a manufacturer for replacement hardware, I often pondered on how hardware could be repaired, and was rather taken aback when the main hard disk containing all of the software for a major development project expired. After much hard work we eventually restored all the files from decidedly flaky tapes, which proved to be a chastening experience.

Only a few years ago, the amount of information which required backing up was such that a few floppy disks had adequate capacity. Indeed, floppy disks were often the main computer storage media. Those readers of more than tender years can probably remember when hard disks were rare (and precious) items on small computers. Nowadays, the increased volume of data that has to be backed up can force users towards an alternative method of taking backups. Such decisions should be taken with due care and attention.

Backup is 'non-trivial'. There is no universally 'best' solution, and this article should be thought of as a series of helpful 'hints' rather than a set of rules.

VIRUS ANALYSIS 1

Fridrik Skulason

Proud and its Relatives

1226, Evil, Phoenix and Proud are related Bulgarian viruses, probably written by the person calling himself 'Dark Avenger'. They were mentioned in the November edition of the *Virus Bulletin* (page 16), with a short description of the decryption method they use. They are also discussed in the eye-witness account from Bulgaria in this month's edition. Although this group of viruses is not a serious threat outside Bulgaria, at least not yet, they are certainly of interest from a technical point of view.

The following article will examine the Proud virus, noting as necessary how the other viruses differ. The most obvious difference is in the length of the virus code shown here in brackets, 1226 (1226), Proud (1302), Evil (1701), Phoenix (1704).

The Proud virus derives its name from an encrypted text string it contains, which says "Proudly made in Sofia". All four viruses were first analysed by Vesselin Bontchev in Bulgaria. He isolated the viruses and sent copies to several virus researchers in the West. Each virus was distributed in three forms, the "standard", "-M" and "-D" form, causing some researchers to believe that each virus existed in three forms in the wild. This is not the case - the "-M" form is a sample multiply infected file. The "-D" form is a memory dump of the virus in decrypted form. Interestingly, this form is also able to replicate.

These viruses only infect COM type files. 1226 will not infect COMMAND.COM, but the other variants will, using the same method as the Lehigh virus - overwriting unused space within the program, so the total length remains unchanged. **In addition to the FAT corruption described later, Evil and Phoenix contain some additional code, which is intended to corrupt EXE files and subsequently attack the hard disk.** According to Vesselin Bontchev these viruses *Trojanise* an EXE file on infected systems which triggers to destroy the hard disk when the virus code is removed (see page 9). I have been unable to verify this effect, but tests are continuing.

When infecting other COM files, the virus code is *inserted into* the file. This is unusual, as most other COM infecting viruses place the virus code either at the beginning or the end of the infected file. The only virus which works in the same way is the 800 virus, also from Bulgaria. It is similar to these four viruses in other ways, suggesting a common author, or possibly just that the author(s) of these viruses had access to the 800 virus. An infected program starts with a 3-byte JMP to the beginning of the decryption code. This code is highly variable, but nevertheless it provides the only possibility for a virus-scanning program to locate the virus.

```
          xchg      ax,bp                   ; save ax register
          mov       r1,100H                 ; beginning of program
          inc       r1
          add       r1,[r1]                 ; initial JMP instruction
          mov       r2,r1                   ; beginning of virus code
          xor       r3,r3                   ; zero key
          mov       r4,length_of_virus      ; length of virus
          push      r4                      ; and store it for later use
again1:   xor       r3,[r1+22h]             ; obtain one word
          inc       r1                      ; and point to the next one
          inc       r1
          dec       r4
          jns/jge   again1                  ; until code has been xor-ed together
          pop       r5                      ; restore length
again2:   xor       [r2+22h],r3             ; decrypt one word
          inc       r2                      ; and point to the next one
          inc       r2
          dec       r5
          jns/jge   again2                  ; until all words have been decrypted
```

Figure 1. Decryption routine - the only code fragment offering any opportunity for a search pattern.

In *Figure 1*, r1, r2, r3, r4 and r5 stand for *ax*, *bx*, *cx*, *dx*, *si* or *di*, the selection of which register is used where, varies from one infection to another. The conditional jumps can also be coded in more than one way, but the first instruction is identical in all infected files.

After decrypting itself the virus will check if it is already present in memory. This is done by checking if the INT 2AH vector points to the virus code. **No other viruses have been found to intercept INT 2AH, but the objective is to gain access to the INT 21H functions, without ever calling INT 21H directly, thus bypassing any program which might be monitoring the function calls.**

If it is not already installed, the virus will allocate a block of 8K at the top of memory, and finally intercept INT 2AH and INT 13H. It uses the same method to intercept INT 13H as the Number of the Beast virus, which involves calling an undocumented INT 2FH function. **This serves a similar purpose as intercepting INT 2AH - that is gaining access to the INT 13H functions, but bypassing any program which might monitor that interrupt**. However, this method will only work on machines running MS-DOS 3.30 or later versions. If the virus discovers another program which has intercepted INT 13H in the same way, it will simply hang the machine.

If the virus is successfully installed, it will restore the original program and transfer control to it.

Interrupt 2AH

The virus intercepts only function 82H, which INT 21H calls for each file-related funtion. **This function is intended to be intercepted by network software, but the virus uses it to bypass TSR programs which monitor INT 21H** The virus only monitors three functions, Open File (3D), Close File (3E) and Execute (4B). When any of those functions is called, the files may be infected. One unusual side-effect of this is that just copying one COM file to another may result in infection of both files.

If the file is opened in read-only mode, the virus will adjust it to read-write mode before infecting the file

Interrupt 13H

The actual damage performed by the virus occurs in the Disk I/O interrupt. Whenever a sector is read from the disk or written back, it is examined to determine if it is a part of the FAT. The algorithm only works on a 16-bit FAT, so most diskettes should be safe from corruption. The virus first counts the number of consecutive entries in the sector, examines each word and checks if it is one larger than the one before it. All words containing 0000 or FFF6-FFFF are ignored. If more than half of the sector contains a list of consecutive words, it is very likely a FAT sector, and will possibly be corrupted.

The corruption is simple, as the virus will just swap two randomly selected numbers in the sector. This will cause serious corruption, but very slowly, as this operation will only be performed occasionally when reading or writing FAT sectors.

This corruption may be detected by a program such as Norton's Disk Doctor or a similar disk repair utility, which might detect a difference between the two copies of the FAT. **However, this corruption, similar to that inflicted by the Nomenklatura virus** (see page 18)**, is potentially extremely pernicious and denotes an obvious sabotage mentality**.

Infection

While the virus is reading or writing files, it installs a new INT 24H handler to prevent any "Abort, Retry, Ignore" messages when writing itself to a write-protected diskette. The virus may infect a file in two ways. One is used to infect COMMAND.COM, and possibly other programs which end in a block of 0-bytes.

The other method is more complex. First the length of the file is checked. Files longer than 64K or shorter than 2K will never be infected. Then an additional check is made to see if the total length of the file after infection will be below 64K. This check will not only exclude files with length in the range 63487-65535, but also files with a length of 14226-16383 or 30720-32767 bytes. The virus then uses the timer to obtain a *random* value, which determines where the virus will be inserted. The first part of the virus, containing the decryption code, modifies its identity and writes itself to the file, followed by the rest of the virus in encrypted form.

Detection

Detecting the virus is somewhat difficult, as no identification string can be provided for it. It would be possible to design a string containing "wildcards", which correspond to the variable registers, as indicated by the decryption source. Ironically, this difficulty in detecting infected files actually results in the virus often repeatedly infecting the same file causing system degradation and increasing the likelihood of the virus' discovery. **The best advice is to use dedicated virus detection software**.

Disinfection

The virus overwrites a part of COMMAND.COM, destroying the previous contents. **This file must be replaced with a clean copy**. The virus also inserts itself into COM files on a *random* basis, which effectively complicates file disinfection. **The safest disinfection method is to delete infected files and restore from clean copies of the master software** Professional disinfection software may also be available to automate the process, but this should be tested prior to general use.

VIRUS ANALYSIS 2

Richard Jacobs

Joshi - Spreading Like A Forest Fire

One of the most common viruses in recent months has been the New Zealand (2) virus. This was, until recently, the only virus to infect the Master Boot Sector of a disk.

Joshi is the second virus of this type to be seen. **Removal of a Joshi infection from hard disks is complicated by the fact that this virus, like New Zealand, is unaffected by a DOS FORMAT**. It is therefore necessary either to perform a low level format of the disk, followed by repartitioning, and then a DOS FORMAT of all DOS partitions on the disk, or to replace the original Master Boot Sector.

The first of these options is painstaking and involves replacing all files on the hard disk from backups. Fortunately in the cases of both Joshi and New Zealand, the 'non-destructive' option is a straightforward procedure involving the restoration of the original Master Boot Sector using utilities such as Norton (*see VB, September 1990, p.9*).

Joshi was first reported in August of this year. The virus originated in the Indian sub-continent and is now widespread in Europe and has recently appeared in the wild in the UK. Unlike many new viruses, Joshi does not employ self-modifying encryption, so every copy is identical. **However, the virus does use 'stealth' which makes it undetectable if it is active in memory.**

The virus consists of a boot sector and then uses a further 8 sectors elsewhere on the disk. One of these sectors contains a copy of the original Master Boot Sector, the next two sectors are not used, and the remaining five contain the virus code.

As with the majority of viruses, Joshi is not deliberately destructive. However, due to an oversight by its author, Joshi is likely to corrupt some data on infected 720 Kb diskettes.

Intentional Side-Effects

The only deliberate side effect of Joshi occurs on January 5th of any year. If an infected disk is used to boot from, the following message will be displayed on a cyan background:

```
Type "Happy Birthday Joshi" !
```

This message remains on screen until the required text is typed in, unless the PC is switched off and booted from a clean disk. Once the text is entered, the boot process continues normally and no further evidence of the virus is seen.

Survival and Deception Features

There are several features of this virus which are of particular interest. **First of all this virus will survive a warm boot (Ctrl-Alt-Del)**. Secondly, on floppy disks the virus formats a new track at the end of the disk, which it then uses to store itself and the original boot sector of the disk. **Also, on floppy disks some or all of the error messages contained within the original boot sector are copied to the virus boot sector, so if an infected disk is inspected on a clean PC, using a utility such as The Norton Utilities, it will look like a clean boot sector. For this reason dedicated virus detection software is essential for reliable diagnosis.**

As with New Zealand, the Joshi virus can only infect a PC if the machine is booted from an infected disk. **Non-system disks can spread infection to a PC; the usual 'non-system disk. Please insert a system disk and retry' will be displayed as the virus goes into memory**. This re-emphasises the danger of negligently leaving diskettes in the floppy drive when the machine is shut down. Once the machine is powered up again, it will automatically boot from the floppy drive, providing the opportunity for a boot sector virus to infect the hard disk. **Note: boot sector viruses will infect any DOS- formatted diskette, regardless of whether it is used to transfer pure data or executable images.**

Operation

When the PC is booted from an infected disk the virus checks as to whether or not it is already in memory. If it is, control is passed straight to the virus, otherwise the amount of available memory is reduced by 6 Kbytes. The virus boot sector plus the 8 sectors assigned to the rest of the virus, including the original disk boot sector, are loaded into this 6 Kb reserved block of memory and control is transferred to the virus in memory.

Next the virus checks the interrupt vectors for INT 8H, INT 9H and INT 13H. If these vectors do not already point to the virus' own sub-routines, they are altered to do so and the previous settings are stored for later use. The virus then sets markers to indicate that it does not know whether the first two floppy drives and the first two fixed drives are infected. It then copies the original disk boot sector stored in the virus' 6 Kb reserved memory block to the address to which it would have been loaded by the computer's start up process. The virus jumps to that address, thus returning control to the normal boot up procedure.

The memory-resident part of the virus is subsequently accessed through INT 8H (Timer Interrupt), INT 9H (Keyboard Interrupt), INT 13H (ROM BIOS disk services) and INT 21H (DOS services).

At this stage INT 21H has not been set. This is because the Master Boot Sector executes before DOS is loaded into

memory and any setting of this vector would be overwritten by DOS. This problem is solved by using INT 8H to set the vector for INT 21H. INT 8H is generated 18.2 times per second to keep the time-of-day clock current. The INT 8H handler monitors the INT 21H vector and does nothing until the vector changes. It then changes the vector to point to its own routine and saves the previous value.

The other function of the INT 8H handler is to monitor the state of the floppy drive motors. If it detects that a motor has stopped, a marker is set so that next time that drive is used the disk is checked for infection. **This means that all uninfected floppy disks used in an infected PC will be infected.**

The INT 9H handler monitors what is typed at the keyboard. If the ''Happy Birthday Joshi!'' message is displayed, this routine supplies the codes of the keys typed to the INT 21H handler rather than to the normal operation of INT 9H. The second function of this routine is to intercept a warm boot request (Ctrl-Alt-Del) and prepare the PC so that the virus remains intact in memory during the boot process.

"Dedicated virus detection software is essential for the reliable detection of the Joshi virus"

The INT 13H (ROM BIOS disk services) handler checks for disk infection and infects all clean disks. Every time INT 13H is called, it checks whether or not a disk is infected. If it is, the disk function is checked and if it is not a request to *read*, *write* or *verify*, the Master Boot Sector control is returned to the INT 13H handler.

Otherwise the first sector on the disk is loaded and 344 bytes of its contents are checked against the copy of the virus boot sector originally loaded during the bootstrapping process. If they match, then the disk is already infected and control is returned to the normal INT 13H handler unless the INT 13H is a call to *read*, *write* or *verify* the Master Boot Sector. If it is, the call is redirected to the original Master Boot Sector rather than the virus boot sector.

Any attempt to read the Master Boot Sector of a disk will show the clean original Master Boot Sector rather than the virus boot sector. **This will cause any virus scanning program to diagnose a PC as uninfected if the virus is memory-resident at the time of checking. This re-emphasises the need to boot the PC from a clean write-protected system diskette prior to using virus scanning software.** Scanning software should not be installed or run from a hard disk.

Infection Routine

Infection is the same on floppy disks and hard disks, except for the location at which the virus is stored on disk.

For hard disks the virus is placed on the first track of the disk, which is unused in almost all cases. For floppy disks an extra track is formatted after the last track and this track is used to store 8 sectors of data. On floppy disks the number of sectors per track is checked and if it is less than 15, the disk is assumed to have 40 tracks, otherwise the disk is taken to have 80 tracks. This assumption is incorrect in the case of 3.5 inch 720 Kb disks, which have 9 sectors/track and 80 tracks, which causes corruption of track 40.

The virus alters the copy of its own boot sector in memory to contain the correct BIOS Parameter Block (BPB) (for hard disks this will be meaningless data). It then copies itself from the reserved 6 Kbyte memory area, which now includes the original Master Boot Sector, into 8 sectors chosen for the type of disk and writes the virus boot sector to the Master Boot Sector location. The marker is set to indicate that the disk has been infected and control is returned to the start of the virus INT 13H handler.

The INT 21H handler checks the date, and if it is the 5th of January of any year, it starts the message routine, which retains control until the correct key sequence is entered.

Disinfection

The PC must be switched off and booted from a clean write-protected system floppy disk before commencing disinfection. A warm boot (Ctrl-Alt-Del) is **not** sufficient to remove Joshi from memory.

For floppy disks, all files can be copied safely to another disk and the disks then reformatted using DOS FORMAT. To copy the files use the DOS COPY command or a file-by-file backup program. **Do not use DISKCOPY or any image copier as this will copy the virus onto the destination diskette.**

For hard disks there are two methods:

1. Backup any data needed and then do a low level format*, followed by FDISK and a DOS format (FORMAT C:/S/V), and then restore all files. (*See VB, July 1990, pp.3-5.*)

2. Use the routine described in *VB*, September 1990, p.9 except that stage 10 of the process should be changed to 'Select: "Side 0",''Cylinder 0'', "Sector 9",''Number sectors 1''' for Joshi. It is advisable to take a full backup before undertaking this procedure, as a mistake could make the disk inaccessible.

* The low level formatting procedure will be described in the manual supplied with the PC. Some system disks are also supplied with a low level formatting utility.

VIRUS ANALYSIS 3

Jim Bates

Nomenklatura - Primitive but Devastating

The recent shift among virus writers towards self-encrypting, self-protecting and anti-disassembly code does not mean that the more primitive types have ceased to appear. One of the latest emanations from the bored and irresponsible Bulgarian virus "factories" is a 1024 byte virus named "NOMENKLATURA" (from the title within the code).

This is a primitive and untidy parasitic virus which infects COM and EXE files. No attempt is made to hide the code either on disk by means of encryption, or in memory by means of interrupt redirection. The code is thus easy to disassemble.

However, Nomenklatura qualifies as one of the nastier viruses because of the nature of its crippling payload and the inconvenience caused by even a slight infection (*Nomenklatura is known to have struck two sites in the United Kingdom. The first report of the virus in the UK was from an individual on the CIX BBS whose machine had become infected as a result of running downloaded BBS software. It is probable that the Nomenklatura virus was deliberately uploaded to Bulletin Boards in the United Kingdom directly from its country of origin, believed to be Bulgaria. This disturbing trend is described in the eye-witness account of the current situation in Bulgaria, pp.8-10. Ed.*)

Installation

The virus code will be appended to the host file in the usual fashion, with appropriate modifications having been made to the beginning of the file to ensure that the virus code is executed first.

The code begins by issuing an "Are you there?" call to the operating system. This consists of placing a value of 4BAAH into the AX register and issuing an INT 21H call. If the interrupt handler returns with the Carry Flag set, then the virus is not memory-resident and the installation routine is invoked. Otherwise processing continues by repairing the header of the host file and then passing control to it.

The installation routine first accesses page zero of memory and collects the INT 13H vector address. This is then swapped into the INT 13H handler by calling the 13H function of the multiplex interrupt at 2FH. This swapping process reveals the vector address of the specific machine disk handler (usually in ROM) which is pushed onto the stack. The INT 2FH is then immediately called again to repair the vectors and the saved address is popped from the stack into memory. The INT 21H vector is then collected by directly accessing the interrupt table in low memory.

Once these vectors have been collected and stored, the code continues by modifying the memory pointers stored in the Program Segment Prefix area to make room for the whole of the virus code to be moved up to high memory. 43 paragraphs (1072 bytes) of memory are made available in this way but before the code is moved, the multiplex interrupt 2FH is called again to insert the virus' own vector into the handler chain. Then the virus code is re-sited with a block move.

The final step of this section of code is to repair the host program header (or original jump in the case of a COM file) and then transfer control to it. Once the virus is installed and "hooked in" to the operating system interrupts, further virus control is established via the newly installed interrupt handler routines.

Interrupt 21H Handler

Nomenklatura intercepts only the Load and Execute (4BH) and ASCIIZ Open (3DH) functions of INT 21H. On receipt of either of these function calls, the relevant file will be checked and infected where possible. The only exception is the virus' own recognition call of 4BAAH which simply clears the carry flag (by reference to the stack) and then returns to the calling routine.

The infection process is similar, regardless of which function has been received. A description follows:

The ASCIIZ filename of the target file is first examined to find either a COM or EXE extension. If neither of these is found then the virus allows the function request to continue unaltered. At this point, no note is taken of which type of extension was discovered and both types are treated similarly until a later check for the presence of the 'MZ' header which typifies EXE files. First the file date and time stamps are collected and stored, then the file is opened for Read/Write access using function 3DH of the original (ie: pre-virus installation).

If the Open request is successful, then a routine is invoked which installs two temporary vectors for use only during the infection cycle. The vectors involved are INT 24H which is modified to prevent DOS from reporting disk errors to the screen, and an INT 13H function dispatcher which is installed to prevent possible FAT corruption during the infection cycle.

Once these temporary vectors have been set, the virus reads the first 24 bytes of the file header into a prepared buffer area. It is at this point (if the read was performed successfully) that a check is made for the 'MZ' header marker and according to its presence or absence, the appropriate infection appending routine is called.

Infection Criteria

Infection of COM and EXE files is substantially similar with one or two small exceptions. With EXE files, any file greater than 1023 bytes in length is a candidate for infection, while with COM files only those with a length between 1025 and 63999 bytes may be infected.

The method by which this virus recognises infected files (for both EXE and COM) is to check the available code after the initial jump destination (or the initial IP setting for EXE files) and if this is exactly 1024 bytes then the file is assumed to be infected. This method is somewhat hit and miss but as someone once said, "*When did the recipient of a virus ever complain that it didn't work properly?*".

"The fact that its payload is delivered on such a random basis and at such a vital section of the disk architecture makes this a particularly nasty specimen."

Once infection is completed, the file date and time stamps are reset to their original values, the file is closed, the temporary interrupt handlers are removed and processing continues with the original INT 21H function call.

Since at no time is a check made of the Read/Write attribute of the target file, infection spread can be prevented by the simple expedient of setting all EXE and COM files to Read Only. However, this will **not** stop an already infected file from installing the virus and the corruption described in the next section may still occur.

INT 13H Handler Routine and 'Payload'

This is essentially a flag controlled routine which monitors sub-functions 0, 1, 2 and 3 of INT 13H. The monitoring processing consists of generating two distinct counts from the contents of each word of the sector buffer. The words are examined sequentially as if they were FAT cluster markers and any word with a value of less than FFF7H (ie: EOF marker) causes Counter 1 to increment. At the same time, any word which has a value of exactly one less than the succeeding word (ie: denoting contiguous clusters) also increments Counter 2. The contents of Counter 1 are then halved and if the original value was an even number, processing continues uninterrupted.

However, if the original value is odd, then a comparison is made between Counter 2 and the halved Counter 1 and if Counter 1 is the lower value, then the "payload" is delivered.

This process is described in detail to give some idea of the random nature of the occurrence of the "payload" routine and in tests on a sacrificial machine, corruption began to appear as file numbers were increased and the disk passed the half-full mark.

It is quite possible that no corruption would occur on a badly fragmented disk, but the nature of this virus is such that it is almost impossible to verify whether corruption has occurred or not.

The corruption introduced by the "payload" consists of swapping a pair of words in random positions within the sector buffer. This is done by using a modified reading from the system clock as a double index into the sector buffer and exchanging the words found at each index point.

The effect on machine operation is totally unpredictable since any two clusters, anywhere on the disk will be transposed. **Thus any file occupying an affected cluster will suddenly contain completely different data at that point, and such data may or may not actually belong to another file**(no attempt is made to check the contents of the transposed cluster words).

Obviously *any* type of file (data, program, system, control) may be affected as will *both* of the File Allocation Tables - but differently. **In the absence of comprehensive backups, recovery from such effects will be totally impracticable for the ordinary user**.

File Recognition

Since this virus is non-encrypting, file recognition is easily accomplished by searching for the signature listed in last month's *Virus Bulletin*:

```
B8AA 4BCD 2173 785E 5606 33C0 8ED8 C41E; Offset 2DD
```

Conclusion

Nomenklatura is poorly written and untidy in its coding. Mention has already been made of its failure to check the attribute of target files, but there are several other indications that the author does not fully understand many of the functions and capabilities of the PC environment. This leads me to conclude that the author is probably someone quite new to computing and who may well have become involved in the "virus fervour" currently reported in Bulgaria.

Nevertheless, the virus works, and the fact that its payload is delivered on such a random basis and at such a vital section of the disk architecture, makes this a particularly nasty specimen. **Nomenklatura infects on executing a program or opening a**

file which means that a virus scanning program will infect all files on the system if the virus is in memory. Should this virus execute on a machine, then all files on that machine will subsequently be suspect. The best course is to delete them and reconfigure the machine from scratch.

All of the above re-emphasise the importance of booting an infected machine from a **clean write-protected system diskette** prior to using diagnostic tools. Note that even a secure CRC checking program which has taken unique fingerprints of all files may itself have suffered corruption along with any associated key files. The only alternative to complete reconfiguration is separate verification of the integrity of each file on a separate machine!

Hidden message? A final section of code exists within the Nomenklatura virus. It is not actually accessed and appears either to be foreign (Cyrillic?) or encrypted text. The message is probably demented babble but may contain information of forensic interest.

Cursory analysis has failed to produce any sense from this code. The code fragment is published here for budding or professional cryptanalists/linguists to conduct their own examination. Decryption and/or translation gratefully received!

Nomenklatura Code Fragment at Offset 384H

```
DB      92H,  0AEH,  0A7H,  0A8H,  20H,  0A4H
DB      0A5H, 0A1H,  0A5H,  0ABH,  20H,  0A8H
DB      0A4H, 0A8H,  0AEH,  0B2H,  20H,  0A2H
DB      0ACH, 0A5H,  0B1H,  0B2H,  0AEH, 20H
DB      0A4H, 0A0H,  20H,   0B6H,  0A5H, 0ABH
DB      0B3H, 0ADH,  0A5H,  20H,   0B1H, 0AEH
DB      0B7H, 0ADH,  0A8H,  0B2H,  0A5H, 20H
DB      0B3H, 0B1H,  0B2H,  0ADH,  0A8H, 20H
DB      0ADH, 0A0H,  20H,   0ACH,  0AEH, 0ACH
DB      0A8H, 0B7H,  0A5H,  0B2H,  0AEH, 2CH
DB      20H,  0A2H,  0AFH,  0A8H,  20H,  0B3H
DB      0B1H, 0B2H,  0ADH,  0A8H,  20H,  0B2H
DB      0A0H, 0ACH,  2CH,   20H,   0AAH, 0BAH
DB      0A4H, 0A5H,  0B2H,  0AEH,  20H,  0B2H
DB      0B0H, 0BFH,  0A1H,  0A2H,  0A0H, 0B8H
DB      0A5H, 20H,   0A4H,  0A0H,  20H,  0B1H
DB      0ABH, 0AEH,  0A6H,  0A8H,  20H,  0B1H
DB      0BAH, 0A2H,  0B1H,  0A5H,  0ACH, 20H
DB      0A4H, 0B0H,  0B3H,  0A3H,  0AEH, 20H
DB      0ADH, 0A5H,  0B9H,  0AEH
```

PRODUCT REVIEW

Mark Hamilton

Sophos' VACCINE

(*NOTE: The VACCINE product and program by that name refer to the package and program produced by Sophos Ltd and not to any other product or program which is offered for sale under the same name.*)

VACCINE is one of a portfolio of security products produced by *Sophos Ltd* based in Abingdon, near Oxford. It claims to provide both specific and non-specific virus detection.

Documentation

The documentation is provided in an A5 three ring, cloth covered binder. At least half the contents consist of a perfect-bound volume entitled *Data Security Reference Guide* which provides general information about data security, viruses, encryption and secure erasure. This book goes on to explain the company's products - carefully omitting price details - in short, an expensive yet impressive product catalogue.

The remainder of the documentation details the various VACCINE constituent programs. Because the software is continuously updated, the authors have chosen to produce the manual "to order" - that is to say, each manual is individually laser-printed. This means that the documentation should, theoretically, always accurately reflect the capabilities of the software it accompanies. This seems to work in practice too, because although there is a READ.ME file on disk, it contained no information, caveats or bug information that was not already in the printed version of the documentation.

There are three principal documents: *Quick Start Manual* (4 pages), *User Manual* (102 pages) and *Using VACCINE in a large organisation* (23 pages) each of which has its own table of contents and the latter two also have indexes.

The documentation is clearly written but the *User Manual* would benefit from some form of chapter dividers for ease of reference. Overall, however, the documentation is excellent.

The Disk

VACCINE is provided on both 5 1/4-inch 360k diskette and 3 1/2-inch 720k diskette. There are six executable programs - including an installation module - and several support files. A file called VIRPATS.LST contains brief descriptions of the viruses which SWEEP, the virus-specific program, can detect. I would like to see this file include a cross-reference of all the names attributed to various viruses. For example, how many people know that what SWEEP refers to as "Cascade (1) 01"

is otherwise known as 1701? As a matter of urgency and to assist bemused users everywhere, computer virus nomenclature needs to be addressed by the research community and a consensus arrived at.

There are two principal software components: the afore-mentioned **SWEEP** and **VACCINE**.

SWEEP searches for viruses by looking for known patterns; the version tested looks for 179 of these. Not all viruses can be found by searching for a pattern and this program is capable of finding a further six viruses by looking for what it calls "identities". To my knowledge this software is, at the time of writing, alone in identifying the WHALE virus in its many guises (*see VB, November 1990, pp.17-20*). Virus hunting is a fast-moving and cut-throat business, so perhaps by the time you read this, Santa can stuff your stocking with other packages with a similar capability.

VACCINE writes a file containing cryptographic checksums (or fingerprints) which are later used by DIAGNOSE which checks to see if there have been changes to files. You are given a choice of three standard "setups" indicating which parts of the disk and memory to fingerprint. A fourth program, FILEMAC, produces the same cryptographic checksums as VACCINE but on a file by file basis and, unlike VACCINE, in a human-readable format. There is also a program to change VACCINE's colour mapping to make it suitable for use on monochrome systems.

SWEEP: Virus-Specific Detection

SWEEP is updated monthly and, according to *Sophos*, has a useful life of three months. The program urges you to obtain an upgrade if it is run after the three month period. This "annoyance" can only be circumvented by tinkering with the system clock. Presumably this message is included to remind customers to renew their subscription to the product.

By default, SWEEP looks for viruses within or attached to COM, EXE, SYS and OVL files as well as the Master Boot Sector and DOS Boot Sector of the selected disk drive. If you want it to include other file types or parts of the disk, you must create an "area" file which includes the details.

When I tested the prior release (4.20) last month as part of a comparative review for *PC Business World*, I criticised the product on three counts. First, it failed to check all overlay files (specifically OVR-type files) for viruses known to infect them without being told to do so, by the expedient of creating an area file. Second, SWEEP produced a number of false positives whereby it incorrectly indicates that a particular file is infected by more than one virus. It also requires a command line parameter to be given to make it check the files' date and time stamps; these are known to be places used by virus writers to indicate infected files.

Well, nothing has changed with release 4.21, these same failings are there.

With virus writers employing tactics to ensure that as many files as possible become infected - as is the case with Fish-6 and 4K which can infect data as well as program files - users need better protection. It is not really feasible to create an area file which includes the extension of every file which could become infected. Anti-virus software houses could easily provide this protection with little additional effort. In fairness, these remarks apply equally to the majority of virus scanners.

Live Virus Detection

SWEEP found all the viruses that it was exposed to including a number which use self-modifying encryption (Casper, 1260, WHALE, Flip, Suomi).

However, I can only give it a "good" rating (rather than excellent) for the reasons outlined above. Speed of operation is acceptable in that it scanned a test hard drive in 3 minutes 39 seconds (*405 files checked, see Test Conditions section on the next page*).

Generic Virus Detection: VACCINE and DIAGNOSE

According to *Sophos*, VACCINE complies with two differing standards for fingerprinting data. It defaults to using ANSI X9.9 in conjunction with a proprietary block cypher algorithm but there is also a command line option to use ISO Standard 8731 Part 2; this method is slower than ANSI X9.9.

In addition to allowing for a password, the DIAGNOSE program itself can be cryptographically protected - using a user-defined "response phrase". This is designed to protect the DIAGNOSE program from tampering.The password and response phrase (which are provided as options) are set and can be changed from within the VACCINE program.

VACCINE provides three levels of protection which, in ascending order of security (and processing time), are known as the *short*, *medium* and *long lists*.

The *short list* is restricted to fingerprints of:

- the Partition Record, the Master Boot Sector, AUTOEXEC.BAT, COMMAND.COM and all files with SYS extensions.

The *medium list* adds the headers of:

- all COM, EXE and OVL files, and the full file fingerprints of all BAT and SYS files.

Finally, the *long list*:

- replaces the fingerprints of the headers of COM, EXE and OVL files with full file fingerprints of these files and adds fingerprints of interrupts 21H, 25H and 26H.

It should be noted, however, that you do have the opportunity of editing the various lists and saving the results, so that you can develop your own particular setup.

Installation and Execution Speeds

The test hard drive contained 423 files that had extensions of COM, EXE, OVL, BAT or SYS. The times taken to create a fingerprint file with VACCINE and then check it with DIAGNOSE are shown in the table.

	VACCINE	DIAGNOSE
Short List	0m 02s	0m 02s
Medium List	2m 40s	1m 24s
Long List	7m 40s	6m 29s

Simulated Attacks

After each run, small and insignificant changes were made to several files that were fingerprinted during that run. These included updating the date and time stamp without changing the file itself, altering file sizes, one byte changes and simulated viral attacks. In each case, DIAGNOSE, trapped and reported each of the changes accurately.

However, in the case of a simple date/time change, DIAGNOSE reported "Bad Attributes" for that file - not really very explanatory. Where contents have changed, it reports "Bad Contents". Interestingly, I added one byte to a text file: DIAGNOSE reported "Bad Attributes" (date, time and file size had changed) and "Bad Contents (one extra byte). In the summary, it erroneously said that two files had altered.

Limitations

VACCINE/DIAGNOSE offer realistic protection on PCs where the fingerprinted files are going to remain static but are of little real benefit in a development environment. DIAGNOSE can be set to run periodically according to the wishes of the security officer by installing it in AUTOEXEC.BAT. It is, of course, essential to ensure that a machine is virus-free before installing the program by using SWEEP and/or another scanning program.

There are inherent risks in such a strategy. Computer viruses such as 4K will subvert VACCINE (and any integrity checking program) if it is run from the hard disk. For this reason, the developers recommend that fingerprints of the required hard disk files are maintained on diskette and that DIAGNOSE is periodically run from the floppy drive after booting from a clean, write-protected DOS diskette.

I noticed one bug in VACCINE whereby if the program is aborted following a fatal error (eg: attempting to write to a write-protected floppy), the cursor is switched off when VACCINE returns to DOS.

The sort of protection afforded by VACCINE/DIAGNOSE is better than by strictly virus-specific programs as it is "future proof". If you are attacked by an unknown nasty - be it hacker or virus - and you've protected your programs and data with VACCINE, you'll know about it.

Conclusions

This is an impressive package which carries the weight of a *CLEF* UKL1 certification for security. To my mind, it is no bad thing that the software does not contain disinfection or inoculation procedures; it is to the authors' credit that these fringe 'benefits' have been omitted. Both are inexact sciences and in the case of disinfection, this is often best left to a case-by-case appraisal should files become infected by viruses.

Would I recommend it? On the plus side, SWEEP is updated monthly; it is written by experienced software security specialists actively engaged in virus research; and it is reasonably fast. VACCINE/DIAGNOSE is secure and detected all the test attacks on files; it has passed Government vetting and its security blanket is extremely flexible.

On the other hand, SWEEP requires specialist settings to ensure maximum coverage; produces false-positive results which could be confusing and DIAGNOSE's status reports are less than clear.

On balance, I can recommend both VACCINE and SWEEP.

Technical Details

Product: VACCINE

Developer and Vendor: Sophos Ltd, 21 The Quadrant, Abingdon Science Park, Abingdon, Oxon OX14 3YS, UK. Tel 0235 559933, Fax 0235 559935.

Availability: IBM PC/XT/AT/PS2, Networks supported.

Version Evaluated: 4.21

Serial Number: None Visible.

Price: VACCINE (includes one issue of SWEEP) £195.00. SWEEP is available separately (year's subscription) £295.00. Site licence and file server prices available.

Hardware Used: Compaq Deskpro 386 running at 16 MHz. Test hard drive with a 1-1 interleave containing 859 files in 22 sub-directories occupying 20 megabytes.

Virus Test Suite: Standard *Virus Bulletin* test suite contained in 168 files (*see Technical Details, VB, November 1990, p.23*). A further battery of 111 virus-infected programs was used for virus-specific tests. In total, 80 separate computer viruses (discounting variants) were used in 278 infected samples.

END-NOTES & NEWS

IBM PC VIRUS (UPDATE)

Bloody! - MR: Isolated in London, UK (2/12/90). A primitive Master Boot Sector virus occupying 1 sector (512 bytes). On the 129th boot and every sixth boot thereafter the virus displays the message ''Bloody! Jun. 4, 1989''. (*Virus Information Service*, UK. Tel 0533 883490*)*.

```
Bloody! 80FC 0272 0D80 FC04 7308 80FA 8073 03E8 ; Offset 1F in MBS
```

The *Virus Bulletin* Conference on Combating Computer Viruses, September 12-13th 1991. The venue will be the Hotel de France, St. Helier, Jersey. The conference will be chaired by Edward Wilding (UK) and Fridrik Skulason (Iceland) and speakers include Jim Bates (UK), Vesselin Bontchev (Bulgaria), David Ferbrache (UK), Ross Greenberg (USA), Jan Hruska (UK), John Norstad (USA), Yisrael Radai (Israel), Ken van Wyk (USA) and Gene Spafford (USA). Several additional speakers, including security specialists from *DEC* and *IBM*, will be announced in the final programme which will be available in February 1991. Information from Petra Duffield, *Virus Bulletin Conference*, UK. Tel 0235 531889.

A **Trojanised version of *McAfee Associates*' SCAN anti-virus software called SCANV70.ZIP** has been discovered on BBS systems in the USA. The most recent release of the bona fide program is SCAN67C.ZIP. The Trojan is understood to erase data and programs.

Network Security, a one day seminar, takes place in London, January 22nd 1990. Information from Amanda Stuart, *IBC Technical Services*, UK. Tel 071 236 4080.

Sophos, UK, continue a series of **Computer Virus Workshops** (London, January 29th-30th 1991). Introductory and Advanced streams available. Information from Karen Richardson, *Sophos*, UK. Tel 0235 559933.

Seminars on Data Recovery, (January 23-24th 1991) and **The Virus Threat**, (February 13-14th 1991). *S & S Consulting Group*, UK. Tel Ann Creamer or Janet Rudkin. Tel 0494 791900.

3rd Scandinavian Conference on EDP Audit Control & Security, Geilo, Norway, February 26-28 1991. Information from Terje Bjornstad, *EDPAA Norway Chapter*. Tel +47 (0)2 52 83 05.

A call for papers for the **4th Annual Computer Virus Conference** organised with *ACM/IEEE* sponsorship has been announced. The conference takes place at the World Trade Center, New York, March 14-15th. Conference information from Judy S. Brand, *Nationwide Computing*, USA. Tel 800 835 2246 X190. Programme information from Richard G. Lefkon, *Data Processing Management Association*, USA. Tel 212 663 2315.

Worming season - reports on BitNet suggest that the CHRISTMA.EXEC IBM Bitnet worm (see *VB*, April 1990, p. 8) has been modified and 're-released'. *VB* wishes Wide Area Network supervisors everywhere a happy and *peaceful* Christmas.

VIRUS BULLETIN

Subscription price for 1 year (12 issues) including delivery:

USA (first class airmail) US$350, Rest of the World (first class airmail) £195

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Abingdon, OX14 3YS, England

Tel (0235) 555139, International Tel (+44) 235 555139
Fax (0235) 559935, International Fax (+44) 235 559935

US subscriptions only:

June Jordan, Virus Bulletin, 590 Danbury Road, Ridgefield, CT 06877, USA
Tel 203 431 8720, Fax 203 431 8165