March 1993 ISSN 0956-9979

VIRUS BULLETIN

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

Editor: **Richard Ford** Technical Editor: **Fridrik Skulason**

Consulting Editor: **Edward Wilding,** Network Security Management, UK

CONTENTS

EDITORIAL

NEWS

INSIGHT

FEATURE

VIRUS ANALYSES

TUTORIAL

PRODUCT REVIEWS

CONFERENCE REPORT

EDITORIAL

A Testing Time

The life of a product reviewer is not a happy one. Not only are their days spent slaving (or slavering, depnding on disposition) over a hot PC, locked in their virus infested workshop, but when they emerge after all their hard work, they will face nothing but criticism. It is true that many reviews are pitifully bad, but even the few carefully thought out reviews get more than their fair share of complaints.

A useful rule of thumb is that nobody is *ever* happy with a review. Excuses for bad performance range from ‘It only costs x dollars, what do you expect?’ to ‘It’s a Beta test version. The bugs will be fixed for the real thing’. However, the most useful excuse of all concerns The Test-Set. Notice the capitals - this is a rather special beast.

The question of how to test anti-virus software has never been satisfactorily dealt with, as so many different factors need to be taken into account. Even *if* the best way to check its efficacy is to run various scanners against an ever increasing battery of infected files, there is the question of which viruses they should be run against. Following a less-than-perfect review, a great deal of sniping about who owns/has seen/has been in the same room as the test-set almost inevitably occurs. But what *should* be in a test-set?

There are two different sources of viruses for the would-be product reviewer. Firstly, there are virus collections available from some of the more anarchic Bulletin Board systems. These collections tend to range in size from 100 to 1000 samples, of which, in most cases, only 80% are functioning viruses. The remaining samples are badly corrupted viruses which do not and cannot work, non-functioning binary images of boot sector viruses, joke programs, text files renamed COM or EXE or all manner of computer ‘odds and sods’. Any scanner test against such a collection would, of course, produce highly misleading results, implying that many scanners do not recognise certain ‘viruses’.

The main problem is distinguishing junk from real samples - that is, transforming this dirty collection into a clean one. Ideally, all the viruses should also be stored on some sort of standard ‘goat’ executable. This operation is completely beyond the capabilites of nearly all reviewers, and therefore Bulletin Board based virus collections should not be used for reviewing scanners.

The only reasonably complete ‘clean’ virus collections today are maintained by companies with a commercial interest in the anti-virus field. There are good reasons for such commercial organisations both to release and not to release the collection.

From a marketing point of view, it is eminently sensible to release the collection to any reviewer who asks for it. This is simply because in any comparative review you are more likely to score well against your own test-set than one from your competitor. Furthermore, a reviewer’s view of a company which refuses (in his eyes) to cooperate with him is unlikely to be favourable.

The main reason for not releasing a virus collection is ethical: the potential damage that would be done if a large collection were to become generally available is incalculable. Releasing the collection implies absolute trust in the recipient - absolute trust that they will not accidently or (God forbid) intentionally release it into the wild.

A constant pitfall which reviewers fall into is in their testing of boot sector viruses. The only meaningful way to test a product’s ability to detect a boot sector virus is to insert a floppy disk with a live copy of the virus into the floppy drive and scan it. Anything less is, frankly, useless. Of course, with some 150 different boot sector viruses this can become just a little tedious. Imagine doing a comparative review of, say, twenty products. This means that working at 1 minute per insertion and scan, it would take approximately fifty hours to test the software. How many reviewers can put their hands on their hearts (yes, most product reviewers *do* have one) and say that they have done this?

Even so called ‘clean’ test-sets have their own problems. Virus samples are frequently the original copy of the virus downloaded from a Bulletin Board, and while that sample *does* replicate, it may be different from all its offspring. Of course, there is still the knotty question of how big a test-set should be. As the number of viruses known spirals upward running tests against ‘all known viruses’ becomes meaningless. Yes - it is important that scanners are kept up to date, but it is even more important that common viruses (those doing damage on user’s PCs) are detected reliably. It is all very well to detect esoteric viruses such as Uruguay 3, but while certain scanners still miss Tequila, this result pales into insignificance. How many viruses the scanner identified is revealing, but *which* viruses it missed is paramount.

A number of anti-virus researchers may feel that the short term gains obtained by letting large virus collections out of their control may get them ‘brownie points’ amongst the reviewing fraternity. What they may be doing is digging themselves a large hole in which to fall, by accidentally increasing the number of viruses in the wild. Frequently heard accusations that the anti-virus industry is behind virus distribution may well become true - and surely no-one wants that to happen.

VIRUS BULLETIN **©1993 Virus Bulletin Ltd, 21 The Quadrant, Abingdon Science Park, Oxon, OX14 3YS, England. Tel (+44) 235 555139.**
/90/$0.00+2.50 This bulletin is available only to qualified subscribers. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means, electronic, magnetic, optical or photocopying, without the prior written permission of the publishers.

NEWS

Michelangelo Day...

By the time this arrives on your desk, PC Support teams worldwide will have survived ‘Michelangelo Day’. Oddly, however, this day looks set to arrive without any of the furore which accompanied March 6th 1992.

Much of the media attention which focused on the trigger date proved to be intensely embarrassing for those who made the wildly excessive claims heard at the time. After this whopping ‘false positive’ by the anti-virus industry, it is rather difficult to get the popular press to take the problem seriously once again.

Interestingly enough, some good did come from the panic - although the frantic searching did not reveal anything like as many Michelangelo infections as predicted, a great deal of other more common viruses were found. Indeed, statistics presented by *IBM* at the 1992 *Virus Bulletin* Conference showed a dramatic ‘glitch’ around March 6th, as users embarked on a scanning frenzy. Readers may like to take note of the deafening silence from last year’s pundits in the Michelangelo sweepstake ❑

Virus Prevalence Table - December 1992

Incidents reported to *VB* during December 1992

Virus	Incidents	(%) Reports
Form	15	34.1%
Tequila	7	15.9%
Joshi	5	11.4%
New Zealand 2	4	9.1%
NoInt 3	3	6.1%
Spahish Trojan	3	6.8%
Cascade	2	4.5%
BFD-451	1	2.3%
Datalock	1	2.3%
Flip	1	2.3%
Helloween	1	2.3%
Necros	1	2.3%
Keypress	1	2.3%
V-Sign	1	2.3%
Yankee Doodle	1	2.3%
Total	44	100.0%

Virus Prevalence Table - January 1993

Incidents reported to *VB* in January 1993.

Virus	Incidents	(%) Reports
Form	28	41.8%
Tequila	7	10.4%
Cascade	5	7.5%
New Zealand	5	7.5%
NoInt	4	6.0%
Joshi	3	4.5%
Michelangelo	3	4.5%
Flip	2	3.0%
Italian	2	3.0%
1575	1	1.5%
AntiCad	1	1.5%
CMOS 1	1	1.5%
Eddie 2	1	1.5%
Keypress	1	1.5%
Spanish Telecom	1	1.5%
Vacsina	1	1.5%
V-Sign	1	1.5%
Total	67	100.0%

Is Nothing Sacred?

The list of objects targeted by viruses contiues to grow. 40-Hex issue 9 contained an article explaining how to write viruses which infect SYS files. Not content with this ‘innovation’, the next object to be infected is the batch file.

Batman (see page 12) is a virus which uses a simple DOS ‘trick’ to convert a batch file into executable binary code. This development adds yet another item to the ever-increasing list of objects which can be infected - and therefore have to be scanned.

However, even if you have anti-virus software installed, it may not be protecting you. The Tremor virus (see page 10) is designed to specifically attack *Central Point’s* TSR (and several other anti-virus packages in addition). The virus is reported to be relatively widespread thoughout Germany. Those speculating about the effectiveness of the rumoured anti-virus component in MS-DOS 6 may find this interesting food for thought.

The important point to note is that no matter how well thought out a TSR detector is, it will always be susceptible to techniques aimed specifically at it ❑

IBM PC VIRUSES (UPDATE)

Updates and amendments to the *Virus Bulletin Table of Known IBM PC Viruses* as of 24th March 1993. Each entry consists of the virus' name, its aliases (if any) and the virus type. This is followed by a short description (if available) and a 24-byte hexadecimal search pattern to detect the presence of the virus with a disk utility or preferably a dedicated scanner which contains a user-updatable pattern library.

Type Codes

C = Infects COM files **E** = Infects EXE files **D** = Infects DOS Boot Sector (logical sector 0 on disk)

M = Infects Master Boot Sector (Track 0, Head 0, Sector 1) **N** = Not memory-resident

R = Memory-resident after infection **P** = Companion virus **L** = Link virus

Known Viruses

A&A - CER: A 506 byte virus, probably of Russian origin, which uses tunneling techniques to bypass monitoring programs.

```
A&A           3D00 4B74 03E9 AB00 8BDA 817F 084E 4474 F4B8 0043 CD78 72ED
```

ARCV - ER: Even though the ARCV group has been 'put out of business', we still see some new viruses from them. Two related variants are Benoit, (1183 bytes) and X-2 (795 bytes), both of which are semi-stealth. In addition the '330' (CN, 330 bytes) variant has been made available, but it is most closely related to the 'Friends' variant.

```
ARCV.Benoit   E800 005E 81EE 0600 8D84 1F00 508D BC1F 00B9 4C04 2E80
ARCV.X-2      8DBC 0E01 B9C9 022E 80?? ??47 E2F9 C3
ARCV.330      E800 00B9 1301 5E81 EE21 028D BC0B 0180 3551 47E2 FAC3
```

Atas.3321 - CR: A much improved version of the earlier Atas viruses - with encryption and stealth features added.

```
Atas.3321     8B3E 0201 B0?? B9E6 0CBE 1300 01FE 3004 46E2 FB
```

Beer - CER: Two variants are known of this virus - 2794 and 2850 bytes long. Not fully analysed.

```
Beer          FA90 3D00 3D74 0F3D 023D 740A 80FC 5674 053D 004B 7523 1E06
```

Bubbles - CEN: Yet another 'mass-produced' encrypted virus, 684 bytes long.

```
Bubbles       8D9E 1701 B97F 022E 8AB6 AE03 2E8A 2732 E62E 8827 43E2 F5C3
```

CCCP - CN: This 510 byte Russian virus contains the texts 'DoomsDay' and '*** CCCP - 75! ***'.

```
CCCP          CD21 B43D B002 061F 8BD3 83C2 1ECD 2173 158D 3ED2 0403 FEFF
```

Deicide_II.Breeze - CN: Similar to the Brotherhood variant.

```
Breeze        B440 BA00 01B9 4B01 CD21 B457 B001 5A59 CD21 B43E CD21 8B1E
```

Diamond - CER: Some virus author recently released 19 new variants of the Diamond virus. Those which are detected with the Diamond pattern have the following sizes: 602, 606, 607, 608, 609, 614, 621, 624, 626, 891, 978 and 1013 bytes. Many of the variants are badly written and may crash the system or cause the 'DIR' command to malfunction. 568, 584 and 594 byte variants:

```
Diamond2      00B4 40CD 213B C174 01F9 C39C 0EE8 ???? 5306 B451 CD21 8EC3
```

444, 465 and 485 byte variants:

```
Diamond3      C307 5B58 80FC 4B74 083D AAD5 75?? F7D0 CF3C 0273 ??60 B824
```

And finally, the 620 byte variant:

```
Diamond.620   00B4 40CD 213B C174 0390 90F9 C39C 0EE8 C200 5306 B451 CD21
```

Dudley - CER: A variable-length, polymorphic virus, which cannot be detected with a simple search string. It seems to be based on the 'No Frills' virus.

Experiment - EN: A 755 byte virus, which is awaiting analysis. It contains the text ‘Small experiments path 2.1’.

```
Experiment    D3E6 83C6 092E 8B86 BC02 33C9 B204 D1E0 D1D1 FECA 75F8 2E8B
```

Kiwi - ER: A 550 byte virus, that contains the slightly unusual text ‘I’m KIWI-586.(C) Vegetable-Soft,1992.’

```
Kiwi          3D00 4B75 1F8B F246 803C 2E75 FA46 803C 4575 114E 4EFD 8CC8
```

LPToff - CR: A 256 byte virus, which interferes with the operation of the printer, by disabling the INT 17H function.

```
LPToff        9C50 5351 0657 561E 52F7 D03D FFB4 7564 B802 3DCD 218B D8BA
```

Lythyum - CN: This 512 byte virus is very similar to another one known as Radyum, and they are probably written by the same author. It contains the text ‘lythyum, the attitude adjuster, ViRuLeNT GRaFFiTi’.

```
Lythyum       E800 005D 81ED 0801 E804 00EB 21?? ??8D B633 018B FEB9 E300
```

Malign - CR: A 630 byte Russian virus, which searches for files to infect whenever the user switches to a different directory. One 575 byte variant is also known.

```
Malign.630    2EA3 1C01 9C2E FF1E 2A01 7212 9C2E 803E 1D01 3B74 1A2E 803E
Malign.575    2EA3 1301 9C2E FF1E 2101 7212 9C2E 803E 1401 3B74 1A2E 803E
```

Minimax - CN: This virus is somewhat unusual structurally. It will make all infected programs at least 31125 bytes long, but the actual virus code is only a small part of that.

```
Minimax       CD21 725F C38B CDFC FAF3 A4FB C3BA 9E00 B802 3DE8 EAFF 8BD8
```

Not-586 - CR: Awaiting analysis.

```
Not-586       EBE7 FAFE C480 FC4C 743A 80FC 2674 0D80 FC36 741C FECC FBEA
```

Oxana - ER: A 1670 byte virus that contains a long message in Russian.

```
Oxana         B890 35CD 218C C88E D82B 06F7 03A3 F703 8CC0 3D00 0075 5FB8
```

Pitch - CN: A 593 byte virus. Awaiting analysis.

```
Pitch         F3A4 BA22 0083 C202 061F B847 25CD 21C7 0600 0000 00B8 1C25
```

Protect.2235 - CER: A new variant, detected with the Protect.1157 search pattern.

PS-MPC.Chuang - CEN: A non-remarkable, 970 byte variant. Detected just like other encrypted PS-MPC-created viruses. Several other encrypted PS-MPC viruses are among those ‘published’ this month, including: Bamestra.1 (EN, 530 bytes), Bamestra.2 (EN,535 bytes), Bamestra.3 (EN, 531 bytes), Bamestra.4 (EN, 536 bytes), Bamestra.5 (EN, 535 bytes), Bamestra.6 (EN, 530 bytes), Bamestra.7 (EN, 529 bytes), Bamestra.8 (EN, 534 bytes), Bamestra.9 (EN, 530 bytes), Bamestra.10 (EN, 530 bytes).

PS-MPC.DemoEXE - EN: A small, 381 byte variant which is not encrypted. It does nothing but replicate.

```
DemoEXE       33C9 99CD 21B4 408D 9600 01B9 7D01 CD21 B801 578B 8E93 028B
```

Skew - ER: A 469 byte virus. Awaiting analysis.

```
Skew          5052 060E 0726 833E 9201 FF74 0726 FF06 9201 EB03 E804 0007
```

Stasi - ER: This 728 byte virus contains a text message claiming it is by the same author as the Witcode virus, and examination of the two viruses did indeed reveal significant similarities, so the viruses should probably be classified as members of the same family. The virus may display the message ‘Stasi is watching you’, but it also appears to contain a destructive code sequence, which is awaiting full analysis.

```
Stasi         1E?? 062E D106 2100 B802 002E D106 2A00 B8?? ??0E ??07 2ED1
```

TU - CR: A 482 byte Russian virus. Awaiting analysis.

```
TU            3DFF FF75 04B8 0100 CF3D 004B 7405 2EFF 2E8C 009C E80F 0152
```

V-388 - CR: A 388 byte virus. Awaiting analysis.

```
V-388         3DCD AB75 04F9 CA02 00FE C43D 004C 7403 E9D7 0050 5351 5256
```

V-550 - ER: This virus might be written by the same author as the previous one, but just like it, this 550 byte one has not been fully analysed yet.

```
V-550         9C3D 004B 7412 80FC 3D74 0D3D CDAB 7403 EB44 90B8 EFCD 9DCF
```

Voodoo - CEN: This 4745 byte virus replcicates in PKLITE-compressed form, with the file header modified so that PKLITE does not recognize it as compressed, and will not attempt disinfection. The virus seems to activate on the birthday of a well-know virus researcher, urging people to send him an E-mail message. Due to the high risk of false positives, it is recommended that this virus is not detected with a search string.

INSIGHT

Meet John McAfee

Many of the big names in the anti-virus industry seem to have become involved by pure chance. John McAfee, founder of *McAfee and Associates*, got involved in the anti-virus industry in a way which is very similar to many others. 'I got involved by accident.' he explained, smiling. 'It was a period of my life where I had little to do. I was shown a copy of Pakistani Brain, and found the whole concept intriguing.'

It was this fortunate accident which led McAfee into the anti-virus world. By the time he saw his first virus, *McAfee and Associates* already existed as a Bulletin Board system and software distribution house, and before long McAfee was distributing his own anti-virus program.

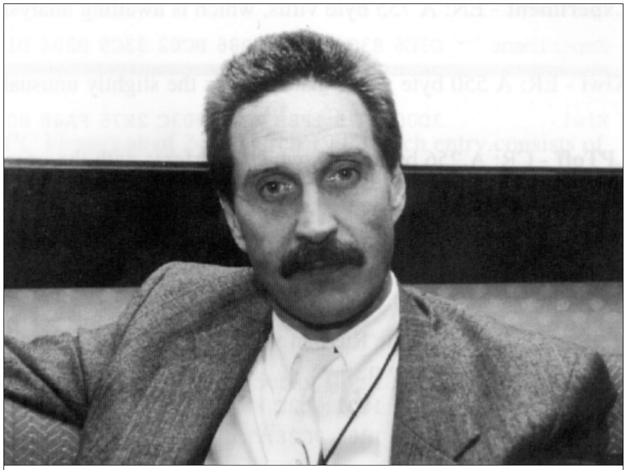
The product began life as a shareware package distributed electronically, and although the product now has many thousands of registered users it is still sent out in this way. This distribution method has undoubtedly been one of this reasons for the popularity of SCAN, although the number of unregistered users is thought to be many times the number of registered users.

Does McAfee have any regrets about deciding to market his product this way? 'No, not at all. I'm a firm believer in electronic distribution.' Regardless of the number of unregistered users? 'We're no different from anybody else. If you shrink-wrap a product you get the same thing. Shrink-wrapping does not prevent software piracy.'

Big Business

The anti-virus industry is now firmly established as big business. I asked McAfee if he was surprised by the way that the company had taken off. 'No, not at all. One of the advantages of shareware is that your product becomes the *de facto* standard if it is a fairly good package, simply because for every single shrink-wrapped package which is shipped, there are a thousand copies of our product which are, quote, ''shipped'' '.

McAfee has recently gone public with his company, selling off 59% of the stock. This has lead to widespread rumours within the industry that he is leaving the anti-virus business and, as one version of the story had it, 'going off to sail his yacht around the world.' I asked him if the floatation of the company marked his retirement from the anti-virus world. 'No, not at all. I am the CEO of the company. We focus exclusively on government agencies and businesses as our clients, and it is more difficult to service some of those sectors as a small unknown company, but as a large public company it is a lot easier. There are a lot of rumours about everyone in this industry, and I neither pay much attention to them or respond to them.'



McAfee: ' If I could see a month ahead, I would feel lucky these days. The virus world is changing daily...'

No Promises

The hardest question for any anti-virus researcher to answer is possibly one of the most common: what does the future hold in store for computer users? 'I wish I knew - I wish that I had that crystal ball. I do not, however. If I could see a month ahead, I would feel lucky these days. The virus world is changing daily, new techniques and technology is being invented, new laws are being considered or passed, vendors come and go - it is a very turbulent industry. If someone says that they know the future beyond a few months, they are either kidding themselves or they have something which I don't have. Every time we try to predict what is going to happen tommorrow we get knocked on our butts - it's happened over and over again.'

McAfee does have some predictions however. He believes that the industry is set to face tougher times: 'I think at this time, the biggest threats come from the new polymorphic viruses, simply because of the man-time required to deal with them. In the long term, the biggest threats come from virus construction sets because once these sets can produce polymorphic viruses the virus problem is going to escalate geometrically. As it stands now, you have to be a fairly competent programmer to write a threatening virus. If you can do it with pull down menus then...'

The increasing complexity of viruses and sheer weight of numbers is putting scanner developers under increasing pressure. With *XTree's* recent departure from anti-virus software manufacture, will other companies soon be following, and will new companies still be eager to join the anti-virus ‘bonanza’? ‘I think the industry is pretty stable now. If you consider the monumental amount of work which is required to start from the ground up and build an anti-virus product, it is very hard for any new companies. We have five years of experience in keeping up with viruses and developing anti-virus software. To do that all at once and to come up with a product in a reasonable time-frame is not realistic.’

Not everybody will find things so easy though: ‘We are certainly going to see a shake-up of the companies which are out there. You are either committed to it or you’re not. If you’re committed to it, you have to work pretty darn hard, and not everybody is willing to do that.’

> *“We have an overriding passion to change the world. We may not - but its our passion nevertheless.”*

‘There are scanners which simply use public domain strings and call themselves a virus product. This will work fine for viruses which can be caught that way, but we’re in a whole new world here - strings are a thing of the past. Even those viruses which can be caught by strings, in many cases we’re converting to the algorithmic approach to make things more efficient. Rather than fifty strings for fifty different viruses, we can have one algorithm for a class of fifty viruses.’

Generic Detection And Cooperation

As the number of viruses continues to go, I asked if he had ever thought about marketing a generic virus detector. ‘There is no such thing in my mind as a generic virus detector.’ he replied. ‘Once you make a generic technique, all future viruses will bypass that. I don’t care what technique is used - put a hard card in, and we’ll still find viruses which get around it. The DIR-II viruses is a prime example. The hardcard sits and watches all changes to executables, but DIR-II doesn’t touch the executables, it just alters the file allocation table, and you can’t restrict changes to the FAT. If you do, you can’t add files, you can’t copy files... you can’t do anything.’

Organisation and cooperation within the virus industry has always been better than that between anti-virus vendors. Things have begun to change in recent months are more and more vendors join *CARO* (*Computer Anti-virus Research Organisation*) which aims to improve communication between vendors. *McAfee Associates* is not a *CARO* member. Why? ‘There’s little we can add to *CARO* - we look at it as sort of a reasearch venture. We don’t look at the industry like that, we think of it as a business. We develop anti-virus software and sell it. I’m not certain what we could give them, or they to us.’

Whether the virus problem ever be controlled is a difficult question. McAfee believes that the problem is here to stay, and users have to live with it. ‘It may not be controllable - we may be in a reactive stage for ever. How do you control car theft and burglary? You are at the whims of whoever wants to steal your car.’ Carrying the analogy further I asked if it was the responsibility of the user to protect themselves. ‘No question. If you leave your car keys in your car and the window down, you can’t complain to the police when it gets stolen that they did not protect your car. The world cannot work that way.’

Loss Of Discrimination

A common criticism within the industry is that McAfee’s scanner does not identify viruses uniquely. However, McAfee firmly believes that ‘precise identification’ will prove to be a luxury the industry cannot afford: ‘Ultimately we have to give up a certain level of disgression. We have to give up a level of separation. Right now, we can identify all viruses uniquely and give names and so on, but ultimately we can’t do that. What we can do is say ‘‘You have got a virus, and it has infected these files, and it is this type of virus. To get rid of it, you do this.’’ *CARO* is not going to like that, I know, because they are really focused on the idea that we have got to name things.’

‘If you go to a doctor, and you’ve got a cold, there are over three thousand discrete cold viruses, and they are discovering more every day. The doctor doesn’t say ‘‘You’ve got virus 1713, subvariety B’’ - he doesn’t care, you don’t care. You’ve got a cold. Do this, this, this and this, and come and see me in a week. We’re going to have to do the same thing with computer viruses. If the user *really* wants to know a name then he’s going to be sadly disappointed two years from now. You can criticise me now, but two years from now, you’re either going to be out of business, or doing the same thing. When we get fifteen thousand viruses, you’re not going to be able to do it anymore - and even if you could do it, it would be meaningless.’

McAfee believes that the future of the software industry is shareware. ‘We are more interested in electronic distribution than anything. We are dedicated to solving the virus problem, and we always will be, but our aim is to try and change the way software is percieved, used and distributed. We have an overriding passion to change the world. We may not - but its our passion nevertheless, it’s our goal.’

FEATURE

Tim Twaits

The G^2 Virus Code Generator

During the past year, a number of utilities have appeared to aid the generation of new viruses. Among the more well known examples are the *Virus Construction Set* (*VCS*), the *Virus Creation Laboratory* (*VCL*) and the *Phalcon/Skism Mass Produced Code generator* (*PS-MPC*). Although the concept of a virus-creation program initially generated some concern, the imagined threat never materialised. So far the programs have only been capable of producing simplistic viruses which have been detected easily by commercial anti-virus software.

The most recent addition to this anti-social collection is G^2 (‘the Second Generation in Virus Creation’). It was produced by the same person who was responsible for the Mass Produced Code generator. He claims that it represents a new generation of virus creation technology. Could this be the realisation of the threat of thousands of complex viruses unleashed by malicious young maniacs with little or no technical knowledge? I suspect not, as the author claims that G^2 took only three days to write.

A Professional Package?

The presentation of the software imitates that of a commercial package. It is supplied with 9 pages of documentation, a quick reference guide and an annotated configuration file. The documentation begins with a copyright notice and a ‘please don’t take me to court’ disclaimer:

> ‘G^2 and the source code generated by said program are not to be used in a malicious or otherwise irresponsible manner. The author is not responsible for any damages, incidental or otherwise, resulting from running either the core program or any programs generated by it.’

One wonders whether this is a plea for respectability or an attempt by the author to distance himself from the legal implications of distributing this program.

Anyone who has ever written an assembly code program will find G^2 easy to use. I produced my first virus in about 15 minutes. The program does not create viruses that are ready to run, but generates assembler source code which must then be assembled and linked by the user. Although instructions are provided, this will restrict the use of the package to those who have some programming experience and access to the relevant assembler and linker.

The generator is capable of producing a number of different types of virus. The characteristics of the generated viruses are selected by editing a text-based configuration file. The configuration file includes comments and is easy to understand. Below is an example of a typical entry, showing how the virus name is specified.

```
; VirusName = <string>
; The name of the virus should be placed here. This
; string will appear directly in the virus code.
; The only limitation  to the string is that you may
; not use both the single and  double quotes in the
; string, ie the string B’li”p is not legal
VirusName = [ G2 Virus]
```

The only function of the generated viruses is replication. There is no trigger, nor are there any deliberate side effects; it is left to the user to add these as required. On the subject of writing destructive routines, the author has the following comment: ‘heck, any programmer worth his salt can write one in his sleep’.

Configuration Options

The following list is a summary of the configuration options available in G^2.

Infect	The file types to be infected (EXE files, COM files or both).
FileName	The name of the assembler file produced.
VirusName	The name of the virus.
AuthorName	The name of the author.
Resident	Should virus go memory resident?
ResID	The identifier used when checking if the virus is already memory resident.
Encrypted	Should virus be encrypted?
IDWord	The identifier used to recognise infected EXE files.
MinSize	The minimum size of files to infect.
MaxSize	The maximum size of files to infect.
Infections	The maximum number of files to be infected at one time
ErrorHandler	Should a critical error handler be installed to suppress error messages?
CommandCom	Should command.com be infected?
AllowZero	Should zero (unencrypted) be used as a valid encryption key?
AntiDebugger	Use anti-debugger techniques?

A number of the options appear to be available purely so that the amount of code generated can be reduced. The size of the virus code varies between roughly 400 and 1400 bytes, though users may add to the code before assembly.

Encryption and Detection

A virus created with the encryption option selected uses a simple exclusive OR or addition with a random value to disguise the main part of the virus code. As with all encrypted viruses, the important part of the virus in terms of detection becomes the decryption routine (which obviously cannot be encrypted). *G*^{2} uses only a small number of variations on a similar theme for the decryption code. They are only slightly polymorphic and will all be recognised easily by algorithmic scanners.

Several further techniques are employed to avoid detection when the use of encryption has not been selected. Each routine can be implemented in a number of different ways and then the individual lines of code within the routines can be rearranged where this does not affect the functionality. This effectively makes it impossible to create a single simple search pattern to detect all the viruses produced by the *G*^{2}. However, once again, I believe that the commercially available algorithmic scanners will be able to detect all of *G*^{2}*'s* creations.

This is not the whole story with regard to the detection of viruses produced by *G*^{2}. The user is encouraged to modify the generated sources and part of these modifictions will most likely be directed toward attempting to defeat any virus detection software. This will only be partially successful, since the virus writer will not know how individual scanners identify the viruses, and even when it is successful, the result will be much less of a threat than a virus produced by simply modifying one of the more sophisticated viruses currently at large.

Non Memory-Resident Viruses

The operation of the non memory-resident viruses produced by *G*^{2} is very simple. When an infected file is run, the virus code attempts to infect other files in the current directory or its parent. The number and type of files infected are determined by the configuration parameters selected. The virus code is written to the end of the infected file.

The virus will not re-infect programs. The method used to determine whether a file is already infected differs between COM and EXE type programs. In COM files the virus checks that the file starts with a jump to the virus code. In EXE files the Initial SP field (offset 0x10) in the EXE header is set to a user defined value (the IDWord field in the configuration file) when the file is infected.

Memory-Resident Viruses

The memory resident viruses produced by *G*^{2} are also very simple. There are no stealth features. A routine is installed at the top of DOS memory, which intercepts all DOS (interrupt 21h) function calls. It infects programs when they are executed. The presence of the virus in memory can be detected using a special DOS interrupt 21h function call. The function number is selected in the configuration file (ResID). This is about the only technical information required in the configuration file. If the user should select a function used by DOS or another application, the operation of the system will be impaired. Those in doubt will likely leave this value set to the default.

Anti-Debugger Code

The option to introduce anti-debugger code causes the virus to redirect the interrupt 3 vector to reference the DOS interrupt 21 handler and perform all DOS accesses using interrupt 3. Since interrupt 3 is used as default by most debuggers to handle breakpoints, it becomes slightly more difficult to debug the code. However, there is no real advantage gained from including this code within the virus, as most researchers would not need to disassemble programs created by *G*$^{2.}$. Any information required about the operation of the virus can easily be obtained by disassembling the program.

The Second Generation?

There is little evidence of *G*^{2} being the start of a second generation of virus creation programs, as claimed by the author. The only feature that has not been seen in previous virus generation programs is the variation of instruction order within individual sections of the virus code. This is hardly a sophisticated or a new idea. In fact, in some ways the viruses generated are less sophisticated than those produced by previous programs, for instance the complexity of the encryption code.

The author claims that the program should be considered as 'second generation' because of the structure and design. There are two parts to the generator: the program and an associated data file. The program itself is described as a generic code generator with no specific knowledge of virus operation. The nature of the code produced is determined by the data file. I have been looking at version 0.7 Beta. We will have to wait for the next version to discover whether there is any truth in this claim.

Implications

The conclusion must be that there is nothing to cause any great concern within the *G*^{2} program except that it is part of an increasingly widespread distribution of information about viruses. *G*^{2} merely provides yet another example for somebody wishing to start writing a virus. The author could have saved himself considerable effort by simply distributing some example virus source code!

VIRUS ANALYSIS 1

Jim Bates

Tremor - A Shaky Start For DOS 6?

It is unfortunately true that no matter how well written an anti-virus program is, it can be targeted by appropriate code. Part of the skill in writing anti-virus software is to appreciate this problem and install comprehensive precautions that make targeting as difficult as possible. The author of the Tremor virus, which is at large in Germany, has obviously disassembled several anti-virus packages and built an awareness of their methods into the virus. Some are avoided and others are subverted as the virus tries to wriggle its way through the defences. Disassembly and analysis of this virus reveals routines which target specific packages, but there are other routines which I was unable to identify. All of the testing and analysis of this virus was done on machines using DOS 5.00 and below.

There is a strong possibility that most of this virus is directed at DOS 6, which I understand contains some built-in anti-virus precautions. If this *is* the case, it simply confirms that any sort of global attempt to include virus protection within a system will provide an easily accessible target that virus writers will be unable to ignore. Even with the highest internal security checks, such protection is bound to be extremely vulnerable.

General Information

Tremor is an encrypting, resident, parasitic virus which appends 4000 bytes of code to COM and EXE files. It marks infected files by adding 100 to the year value in the Date/Time field of the file directory entry. This marker has been used by other viruses (notably the 4K or Frodo virus) and many packages will detect it quite easily. However, when the virus is resident, a stealth routine removes the marker if any attempt at monitoring is detected. This virus is capable of using the upper memory blocks or extended memory when it becomes resident.

Tremor infects files by simply appending its code to the end of the file and altering the file header to ensure that the virus code is executed first. It is therefore possible to recover the original file image by removing the virus code and repairing the file header. Once infected by this virus it is essential that a machine be booted from a clean system disk because the virus will invariably infect COMMAND.COM or any other file named in CONFIG.SYS as the command interpreter.

Virus Installation

When initially executed, the virus first decrypts itself and enters an installation routine. This collects the current system date and compares it against a date inserted when the file was infected. If the current date is less than three months after the file was first infected, the code is modified to prevent the shake effect trigger from being processed. The virus then checks the DOS version and aborts if it is 3.29 or below.

Next, a check is made which depends upon previous infection conditions. If when the file was originally infected an INT 01h routine was installed and a function 30h request (get DOS version) was detected which had the current date in the CX:DX registers, the code which processes the installation check is modified so that the virus will not function. This INT 01h check is presumably a test for the existence of an anti-virus monitor. The design is such that if the check fails, installation continues - otherwise installation is aborted and processing eventually returns to the host program. In this way, the check also functions as an ‘Are you there?’ call.

Installation continues by testing for the presence of extra memory. This virus first attempts to install into extended memory and if that fails it tries the upper memory blocks. If this too is unsuccessful, the virus installs itself into the top of conventional memory.

Once the code has been copied into the chosen memory block, the INT 21h and INT 15h vectors are collected and stored within the virus code. Then an INT 01h routine is invoked which uses the techniques known as tunnelling to determine the true system entry point of the INT 21h service routine. The virus then checks if an additional TSR has been hooked into this interrupt and if so, this entry point is also stored and the MCB marker is checked for the unusual value of 44h. This too is probably a check for the presence of anti-virus software!

Processing continues by creating a temporary disk transfer area and searching for the first file in the root directory with the archive bit set. The date stamp of this file is checked and compared to a previous date collected in similar fashion when the file was first infected. If these dates do not match then the shake effect trigger routine is disabled. Thus a newly infected file introduced to a machine will not cause the trigger to activate.

The system environment is then examined for the file specified in the ‘COMSPEC=’ variable. This file is infected by using a special call to the newly resident virus code. This means that the command interpreter file (usually COMMAND.COM) will become infected during the first execution of the virus code. This special attention to the command interpreter ensures that on subsequent reboots, the virus code becomes resident before most anti-virus software. After this infection, the installation routine concludes by repairing the host file image in memory and erasing all traces of the initial virus code from memory (leaving just the resident code active).

Resident Operation

Most DOS function requests are intercepted, but under a varying range of conditions. A large proportion of the resident code is involved with avoiding detection by different checking programs. For example, special code is included to ensure that the DOS CHKDSK program does not show anything amiss. These tests are too convoluted to list here and include several self-modifying options which are applied under highly specific circumstances.

However, one check worthy of special mention concerns *Central Point Anti-Virus*. Some anti-virus detection software needs to disable its activities under certain circumstances and such potential security loopholes are usually well protected within the code. During the DOS interception routine, the Tremor virus issues a special INT 13h call which appears to turn off the *Central Point* vector checking routines and thus allows unimpeded changes to be made to the system vectors.

Infection of files seems to take place during most of the intercepted functions but conditions vary according to the prevailing system condition.

Within the resident code there is a check to see if the target file name begins with CH, ME, MI, F2, F-, SY, SI and PM. If so, temporary changes are made to the allocation of system memory to avoid detection. Similarly, if the second and third letters of the file name are ‘RJ’ then part of the interception routines are disabled.

Triggers

This virus has two trigger routines. The first is called very rarely (on a random basis) and produces a slight vertical movement in the screen display before causing the machine to hang. The second routine is hooked into the INT 15h intercept routine and as this interrupt service is used by many different packages (*MultiDos*, *DesqView*, etc) it is impossible to forecast when this will execute. The routine displays the message

```
     -=> T.R.E.M.O.R was done by NEUROBASHER
          / May-June '92, Germany <=-
  -MOMENT-OF-TERROR-IS-THE-BEGINNING-OF-LIFE-
```

on a cleared screen, waits a few seconds and then continues normal operation

Conclusions

The internal security of some anti-virus packages is obviously called into serious question by this virus. It is not my place to reverse-engineer commercial anti-virus software but during investigation into this virus I had occasion to check the primary operation of several packages against the various routines that were obviously attempting subversion. The *Central Point* checks were certainly the most obvious and seemed to take no special security precautions against being targeted.

I am seriously concerned that such a widely used package as *Central Point Anti-Virus* is so open in its internal construction that targeting becomes extremely simple. Writing anti-virus code that does not incorporate in-depth security checking is a little like designing a brand new lock and then placing it inside a transparent casing - with a little inspection, anyone can pick it!

Becuase of the complexity of this virus, my commented listing is available to anti-virus vendors who may recognise a potential threat to their own package.

TREMOR

Aliases:	None known.
Type:	Resident, Parasitic file infector.
Infection:	COM files less than 60,001 bytes, and EXE files below 1,048,576 bytes.
Self-Recognition:	
File	Year field in file Date/Time stamp is greater than 100.
System	INT 01h handler present.
Hex Pattern:	No simple recognition pattern is possible.
Intercepts:	INT 21h (most functions) for stealth and infection, INT 24h for internal error handling, INT 15h for trigger 2, INT 03h for armouring, INT 01h for tunnelling.
Trigger:	Vibrating screen effect or message display routine.
Removal:	Specific and generic disinfection is possible. Under clean system conditions, identify and replace infected files.

VIRUS ANALYSIS 2

Eugene Kaspersky

Batman - Robbin' Users of Security

Whenever a new virus lands on my desk it falls into one of four broad catagories, which do not really align themselves with the usual system for classifying viruses. In my personal system the first group, which is also the most numerous, contains all the standard variations on the theme of COM and EXE infectors or boot sector viruses. These viruses are now sufficiently common that they usually receive no attention save their inclusion in the *VB* list of known PC viruses. Most do not utilise stealth techniques, and they are usually bug-ridden scraps of code - really nothing more than computer graffiti.

Another group of viruses are those which are polymorphic, or which use increasingly complex methods of stealth in order to hide their presence on infected files or disks. Examples of viruses which fit into this group are MtE-based viruses, Girafe, and the Uruguay series to name but a few. These virues are frequently more difficult to detect reliably, and can be a cause of much head scratching among those developing anti-virus software.

The third group contains attempts to write the shortest possible virus. Due to this self-imposed length restriction, these viruses contain no trigger routines. Although none of these viruses has a significant chance of spreading the virus authors seem to be continually trying to outdo one another. The whole thing looks rather like a competition between the virus writers, though I have no idea what the prize is!

The most interesting group of viruses are those which use new algorithms for infection or disguise. All virus researchers will remember, for example, the first stealth file virus (4K or FRODO), the new method of the file infection used by DIR-II, or the way the STARSHIP virus infects the boot sector of a disk. Although the vast majority of virus writers are unusually inept at assembly language, every now and then they have the unfortunate habit of coming up with the ocassional cunning idea.

The latest new development I have seen is a memory-resident virus which is capable of infecting batch files. While there are a couple of supposed batch files infectors known, this virus infects in a somewhat unusual way. The virus has no trigger routine, and while it is relatively easy to remove, has a high nuisance value. It is to be hoped that Batman does not herald a new age of infectors which target slughtly more unusual objects.

Simple Tricks

It is easier to show part of the text of an infected batch file than to explain the technique used by the virus. The text inserted into the batch file is very simple:

```
@ECHO OFF
REM <<< binary code: jmp installation, int_21
handler part 1 >>>
copy %0 b.com>nul
b.com
del b.com
rem <<< binary code: TSR installation, int_21
handler part 2 >>>
```

(Note: the brackets <<< >>> mean that the non-text bytes of the virus would normally be located here.)

The unusual thing about this virus is that the code can be executed as one of two different file types: either as a batch file or as a COM file.

When this file is executed as a batch file it can be seen that the virus will create a copy of the batch file with the extension COM by using the command

```
copy %0 b.com
```

The %0 parameter substitutes the name of the batch file as it typed at the command prompt. The newly created COM file is then executed.

Inside The COM File

When the batch is is renamed to b.com the start of this file still begins with the text string

```
@ECHO OFF
REM
```

However, because this file now has a .COM extension, this text is interpreted as the i8086 instructions

```
INC AX              ; @
INC BP              ; E
INC BX              ; C
DEC AX              ; H
DEC DI              ; O
AND [BX+46],CL      ; <SP>OF
INC SI              ; F
OR AX,520A          ; <CR><LF>R
INC BP              ; E
DEC BP              ; M
AND ??,??           ; <SP>
```

These 'junk' instructions do not influence COM program execution (just as the remarked binary code does not effect the execution of the batch file). Once they are executed, the virus code stored in the batch file after the REM statement is executed. It is this binary code which allows the virus to become memory-resident.

The virus uses standard DOS function calls to go resident, and makes no attempt to hide its presence. It hooks INT 21h by using the DOS functions GET INTERRUPT VECTOR and SET INTERRUPT VECTOR and then goes resident exiting used an INT 27h (TERMINATE AND STAY RESIDENT) call. The virus does not have an ‘Are you there?’ call, and so will multiply infect memory every time an infected batch file is run.

The viruses INT 21h handler only intercepts the WRITE FILE function (INT 21h, AH=40H). The virus checks the beginning of the write buffer for the string ‘@ECHO’. This string is commonly found at the start of many batch files. If it is present the virus writes itself into the file before saving the contents of the buffer. Therefore any batch files which start with this text string will be infected upon creation, copying or modifying.

When this part of the virus returns control to the batch file, the newly created COM file is deleted from the disk, thus leaving no evidence of the foul play which has gone on.

Poor Coding

This virus is not well debugged - it looks more like the trial of a new idea than a serious attempt at writing a virus. There are several errors in the infection algorithm.

Every now and then the batch file will cause DOS to display the error messages

Bad command or file name or File not found during execution. This is caused by the presence of redirection signs (‘>’ or ‘<‘) or pipe (‘|’) in the virus code commented out by the REM instruction.

The virus stores the original INT 21h handler address in its own code and that address in ASCII form can contain any characters including ‘>’, ‘<‘ and ‘|’. When the batch file is run DOS rather surprisingly interprets these signs and will report an error.

The second error manifests itself on execution infected batch file by typing in the batch file name without its extension. The %0 batch parameter will be equal to file name only and DOS can’t execute the command

```
copy %0 b.com
```

because that file is absent. In this case DOS display the message:

```
File not found
Bad command or file name
File not found
```

However, this error is trivially avoided.

Due to the lack of an ‘Are you there’ call or any form of self-recognition on files, the virus also multiply infects files. In the extreme case, when there several copies of the virus resident, each resident copy of the virus will infect the batch file before it is written to disk.

What will be next?

This virus is not a significant threat in itself, as it is unlikely to spread. However, it does illustrate that new methods of infection are being thought of all the time. Any object which can form executable code is a possible target for infect. This includes object files or even files which infect C source code or libraries. Therefore it is important that those relying on integrity checkers as their main line of defence are aware of which objects they need to protect.

Some manufacturers exclude certain files (for example, CONFIG.SYS) argueing that this drastically reduces the number of alarms given by the product. If so, then great care should be taken when deciding whether a system is truly virus free. What next - a multipartite BOOT-SYS-BAT-DLL infector? - the possibilities are endless Well, we have no choice but to wait and see. Meet BATMAN - the memory resident BAT file infector.

BATMAN

Aliases: None known.

Type: Memory-resident, Parasitic file infector.

Infection: Batch files which start with the text @ECHO OFF.

Self-Recognition:

Files None

System None

Hex Pattern:

```
4045 4348 4f20 4f46 460d 0a52
fc40 756f 9c50 5351 5256 571e
```

or the text string:

```
copy %0 b.com>nul
b.com
del b.com
```

Removal: Under clean system conditions delete the first six lines of infected batch files using any text editor.

TUTORIAL

James Beckett

The Danger Within

Though recognition of the virus threat is now growing among even casual computer users, many are unaware of the subtleties of their system configuration files. Often these will be created as defaults by installation programs or copied as examples out of manuals. The finer points are usually left to obscure appendices or the in-depth manuals intended for programmers.

Most users simply do not have the time to investigate all the nuances of their system, they just accept the system as it is, and get on with their job.If you know what AUTOEXEC.BAT, CONFIG.SYS and COMMAND.COM actually *do*, you're ahead of the average PC user.

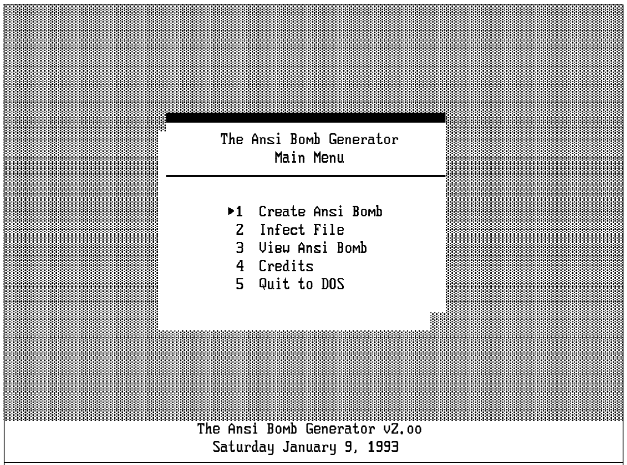
Most PCs one comes across are likely to have an innocuous-looking line 'DEVICE=\DOS\ANSI.SYS' in the CONFIG.SYS file; often users do not know why it is there or what it does - 'If it ain't broke, don't fix it'.

Colourful Utility

Anyone who uses electronic Bulletin Boards on a serious basis will have heard of ANSI graphics - often the first question asked when you log on to a new board for the first time. When on-line on a plain BBS, the text rolls forth in stark black and white (or grey and green, or a nasty off-pink depending on the disposition of your monitor) and this can get rather monotonous (or monochromous). There is great psychological value in appropriate use of colour to highlight inportant points, warnings and so on, and while sometimes over-used it does brighten up the world.

Special non-printing control characters are embedded into the text, which are interpreted by the terminal emulation software as commands to change text colour, to warp the cursor to another screen position for graphics, animation, and screen output optimisation. (Optimisation is the original reason for terminal control - if a four-character sequence can be used to erase a whole line, a great saving is achieved over having to send 80 spaces to clear the line)

All of this is provided internally by the terminal emulation software, and many standards exist, implemented by many different packages. DOS itelf offers the same system. Using ANSI.SYS, mere batch files can output coloured text and graphics, put colour in DOS prompts, and captured BBS logs can be replayed outside the terminal program.



```
           The Ansi Bomb Generator
                  Main Menu

           ▸1  Create Ansi Bomb
            2  Infect File
            3  View Ansi Bomb
            4  Credits
            5  Quit to DOS

        The Ansi Bomb Generator v2.oo
          Saturday January 9, 1993
```

'The ANSI Bomb Generator' allows very quick and simple ANSI escape sequences to be inserted into host programs.

In addition to these benefits the system can make for greater portability of applications programs which display graphics - if standard terminal sequences are produced by a program, it can be easily ported to any system which can support them, not just DOS, removing reliance on the BIOS character routines. This is rarely used though these days for reasons of efficiency.

All of the above sounds rather useful. Unfortunately, as always, there is a catch.

Taking The Con

The standard DOS system console driver CON is not very sophisticated - output characters are sent to the screen, and pressed keys are provided to the running program as input. About the smartest thing is does is clear the screen, a function provided by the BIOS anyway (the same applies for the scroll function).

ANSI.SYS can be loaded at boot time to provide an alternative driver - this looks for ASCII ESCAPE characters in the output stream, and interprets recognised sequences. For example:

`ESC, '[', 'K'`	erases the current line.
`ESC, '[', number, 'm'`	sets the colours to be used.
`ESC, '[', '2', 'J'`	clears the screen.

In combination with the IBM PC high-bit graphics characters, quite impressive effects can be achieved with relatively little effort, and no knowledge of colour graphics adaptor programming is required.

A simple text file on a distribution disk can contain colour and animation, all initiated simply by viewing the file - no program need be run to create the effect.

But with all this ‘useful’ cursor and graphics control there is an extra feature which is rarely used by applications programs - ANSI.SYS takes control over the keyboard as well as the screen.

Text File Danger

Using ANSI.SYS, one can program a key to produce several characters - for example unused function keys can be set to execute commonly-used DOS commands. The sneaky thing is that as with the graphics commands, you do not need to run a special program to do this key re-mapping. All that is required is to output control characters to the screen, and these could come from anywhere. Simply TYPE-ing a text file could invisibly reprogram your ESCAPE key to run the FORMAT program when pressed at the DOS command prompt.

Users are now reasonably careful about executing unknown programs, but usually see no harm in looking at the contents of a text file. There are several cases where unknown characters will be echoed to the screen without user control. Given a shareware disk, the first thing a user might do is display the README file; even taking a directory of a disk echoes the filenames to your screen, and they and the volume label come through unfiltered. No legitimate DOS filename will contain escape characters but they can easily be entered with a disk editor.

ANSI Bombs

The virus writing fraternity have been aware of the ‘fun’ which they can have with ANSI Bombs (destructive ANSI sequences) for some time. There is even an ANSI Bomb creation package available. Using this package it is easy to redefine any key to represent any keystrokes required.

The package claims to be the result creation of ‘The Jolly Anarchist’ and, using a simple menuing system, allows keyboard re-assignments to be embedded into any chosen file. Simply type in your desired string, and the package does the rest.

Pack Up Your Troubles

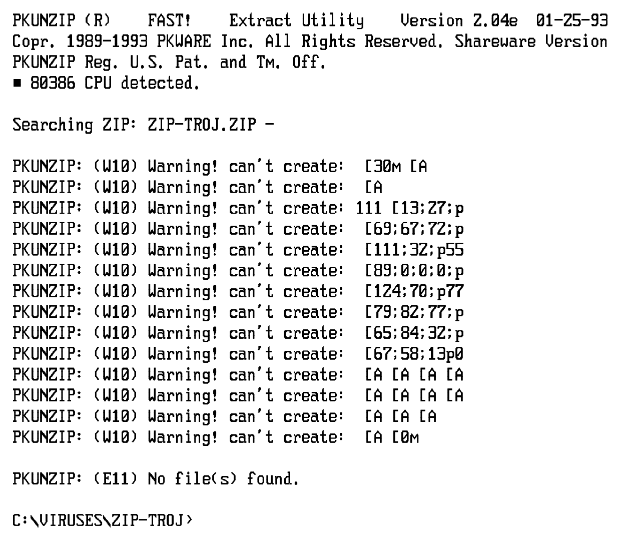
The specific case we are going to consider here is the highly popular shareware compression/uncompression package PKZIP. Other packages have similar problems.

PKZIP can compress many files into a single wrapper, including complex recursive directory structure information, and is often used as a distribution form for software packages and any groups of related files. There has always been a certain amount of risk from Viruses and Trojan horses hidden in ZIP files, as one can’t immediately see the contents of them. Most anti-virus products will not scan inside the compressed file, partly because the system is copyright and often updated. Licensing fees are expensive and so is constantly changing your software to account for new features of someone else’s.

Trojans are hidden as effectively as viruses; for example one of the files we uncompress may be deposited in the current directory with the same filename as a commonly-used command (eg XCOPY), and the next thing we do is to XCOPY the files somewhere else, then the file which is run is the newly-created XCOPY.EXE, which could go ahead and do anything. Or, if we have several ZIP files to process, and the first one extracts a replacement PKUNZIP.EXE (and if we don’t notice) then we get bitten uncompressing the next file [*Blush. Ed.*]. Careful examination of the file names before starting (using PKUNZIP -T) will reveal this easily, though, and the problem is more deeply rooted.

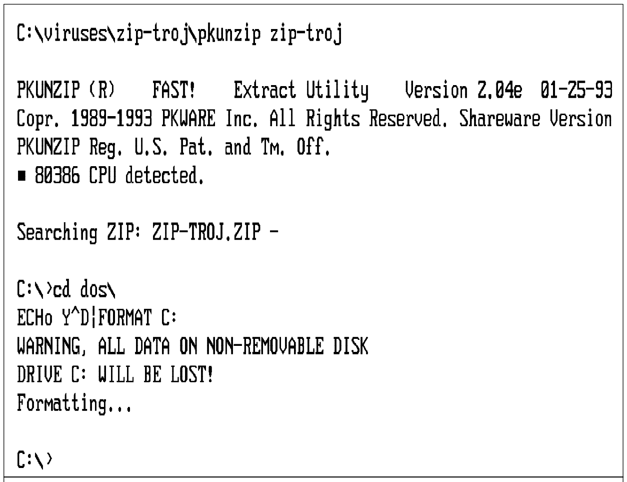
A zipfile can also contain a comments area - not an extractable file but a piece of text displayed whenever a file is extracted. The ANSI bomb problem was recognised some time by PKWARE for some time, the default action has been to inhibit ANSI from being displayed from the comments text, though there is a command line qualifier to enable it. So that is not it, either.

A zipped file called ‘CON’ will be decompressed and sent directly to the screen - DOS does not discriminate between device names and file names - via a loaded ANSI.SYS. The effect of this will not be seen until you next press RETURN and your PC responds , for example, with FORMAT C:, with an ominously flashing disk light. But we can see



```
PKUNZIP (R)    FAST!    Extract Utility    Version 2.04e  01-25-93
Copr. 1989-1993 PKWARE Inc. All Rights Reserved. Shareware Version
PKUNZIP Reg. U.S. Pat. and Tm. Off.
■ 80386 CPU detected.

Searching ZIP: ZIP-TROJ.ZIP -

PKUNZIP: (W10) Warning! can't create:  [30m [A
PKUNZIP: (W10) Warning! can't create:  [A
PKUNZIP: (W10) Warning! can't create: 111 [13;27;p
PKUNZIP: (W10) Warning! can't create:  [69;67;72;p
PKUNZIP: (W10) Warning! can't create:  [111;32;p55
PKUNZIP: (W10) Warning! can't create:  [89;0;0;0;p
PKUNZIP: (W10) Warning! can't create:  [124;70;p77
PKUNZIP: (W10) Warning! can't create:  [79;82;77;p
PKUNZIP: (W10) Warning! can't create:  [65;84;32;p
PKUNZIP: (W10) Warning! can't create:  [67;58;13p0
PKUNZIP: (W10) Warning! can't create:  [A [A [A [A
PKUNZIP: (W10) Warning! can't create:  [A [A [A [A
PKUNZIP: (W10) Warning! can't create:  [A [A [A
PKUNZIP: (W10) Warning! can't create:  [A [0m

PKUNZIP: (E11) No file(s) found.

C:\VIRUSES\ZIP-TROJ>
```

Hmmm... these file names aren’t what were expected. Lucky ANSI.SYS was not installed...



```
C:\viruses\zip-troj\pkunzip zip-troj

PKUNZIP (R)    FAST!    Extract Utility    Version 2.04e  01-25-93
Copr. 1989-1993 PKWARE Inc. All Rights Reserved. Shareware Version
PKUNZIP Reg. U.S. Pat. and Tm. Off.
■ 80386 CPU detected.

Searching ZIP: ZIP-TROJ.ZIP -

C:\>cd dos\
ECHo Y^D¦FORMAT C:
WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE C: WILL BE LOST!
Formatting...

C:\>
```

Unzipping the file with ANSI.SYS loaded: Whoops - goodbye hard disk. Even innocuous text files can play the same trick.

filenames like that too, so even this doesn't *have* to catch us by surprise. As with all things concerning virus protection, a little thought and preperation can save a lot of trouble.

In fact several DOS device names are points of attack for Trojans - the printer device PRN or LPT1 can be sent junk to spew out and ruin reams of paper, a modem on a COM port can be sent dialling strings to dial expensive international numbers or chat lines, even the PCs real-time clock has its own device name (CLOCK$).

It seems we can always get out of it by looking at the file names first. So some bright spark has gone and put the ANSI bomb in the filenames themselves - if you run through the uncompression or even just look at the file index, the embedded escape sequences are accepted by ANSI.SYS and acted upon.

The simple example we have seen forces a FORMAT of C: when RETURN is pressed, but we can foresee more subtle bombs appearing. It would be easy to include a new virus in a ZIP file and make a key mapping to run it. If a resident infector, the user may not notice any extra delay at the time. An alert user might expect to notice if an unwelcome file were created, but we have already seen a countermeasure in this ANSI FORMAT bomb: It includes additional ANSI sequences to move the cursor back up a line so that the next thing printed erases the evidence. Unless a very old and slow PC is used, the flicker of text is unlikely to be noticed.

The actual codes used can be found in most DOS reference books, and even the standard DOS 5 system manuals explain how to do it, so no harm can come of repeating them here:

```
ESC, '[', key-code, 'string', p
```

will set the key specified by key-code to produce the string. The 'p' is the operation code for 'program'. The codes can be entered using most editors or word processors in non-document mode, with the ESCAPE character often appearing as ^[or \E. Most of the key-codes are the standard ASCII character codes, but the extra keys on the keyboard have special numbers, or even two numbers (zero, then another). So, for example:

```
^[[13;'Boo!'p
```

will disable the return key, causing it to act as if you had instead typed "Boo!". Take care if you try this, as you cannot then enter *any* commands and so cannot undo the effect without rebooting! You could first use

```
^[[10;13p
```

to program the shift-return combination (normally LINEFEED) to give a RETURN code; prepare a file with

```
^[[13;13p^[[10;10p
```

to return them both to their defaults.

Reducing The Risks

What can be done? Well, the simple solution is to disable ANSI.SYS if you do not really use it; this can be quite subtle though, as on some systems the ANSI driver also partially controls the VGA 43- and 50-line modes, so you can only use the DOS command shell in 25 line mode.

Alternatively you could patch your copy of ANSI.SYS to ignore the re-map command. This would require a lot of detailed knowledge about how the driver worked and 80n86 programming in general, and would need doing again with every release of DOS you install.

Replacement drivers are available - NANSI.SYS has been offered for some time now by Daniel Kegel and contains command line switches to disable the reprogramming feature. The program is shareware and for the registration fee of $10 it provides an additional degree of security. It is relatively easy to obtain from various software libraries accessible from the Internet.

The best of both worlds? Well, if you like all that colour, maybe, or if you cannot stand the way that a certain application always leaves the screen a nasty green-on-blue, sure. But many would ask, why should we go and buy NANSI - surely *Microsoft* should be providing this feature as standard? The problem is well known to PC professionals, but DOS still blissfully supports this feature. Sadly, the changes required may be on a wish list somewhere, but are probably way down the list of priorities.

PRODUCT REVIEW 1

Mark Hamilton

Leprechaun - A Secure System?

Our editor is a hard man - ask anyone who writes regularly for *Virus Bulletin*. Apart from demanding huge amounts of error-free copy to near-impossible deadlines, he always attempts to get the best value for money from contributors. His latest trick has been to send me two packages disguised as one: *Leprechaun Software's C:Cure*, a write protection device for IDE hard drives, and their scanner, *Virus Buster*.

Last time I reviewed *Virus Buster* in *VB* (November 1991, p.21 - 23) I said that the package was 'a rising star and looks set to figure prominently in *VB's* comparative reviews.' Nearly a year and a half has passed - how has *Virus Buster* been shaping up 'Down Under'?

The Black Box

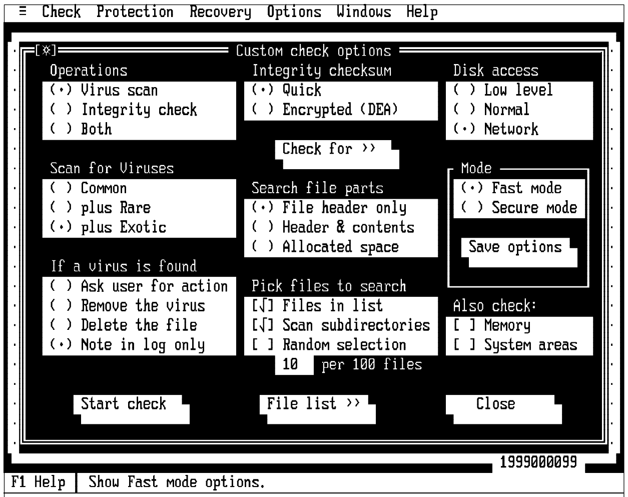
C:Cure is a hardware device which sits between the hard disk controller and hard drive, providing write protection for certain areas of the disk. The product consists of a small black plastic box which houses a circuit board 6.5cm by 3cm. It has two 40-way pin connectors, a power lead which attaches to the drive's power connector and a short (actually, too short) length of ribbon cable. Also on the PCB is a buzzer, a four pole switch block, a chip (probably a PAL) and a couple of discrete electronic components.

The documentation supplied with *C:Cure* is extremely confused, and, if the product is to be a success, must be thoroughly revised. When faced with situations like this, there is only one possible solution: consign all the manuals to the top shelf and place one's trust in instinct and luck. Hopefully, things will go well and the manuals can stay there gathering dust. In this instance, installing the device turned out to be simple - the user first installs *Virus Buster* (see below) and then runs the *C:Cure* installaton routine.

C:Cure is only available for IDE drives. I do have a machine with an IDE drive, but I do not use it for anti-virus testing, as it is my development machine, and its hard disk is full of very expensive development tools. Suddenly, right now seemed like a very good time to do a backup!

Divide And Conquer

To use *C:Cure* the user must first repartition the hard drive of the machine into two partitions. When the package is running, one of these partitions will be write-protected. The software lets the user decide on the size of the C: drive (this is the drive that will be write-protected). The software gives you the choice of a partition of 16, 32, 64, 128, 256, 512 or 1,024 tracks. I opted for the minimum size which gave me a C: drive of nearly three megabytes - just enough for a barebones operating system drive.



```
≡  Check  Protection  Recovery  Options  Windows  Help

[*]                    Custom check options
 Operations               Integrity checksum       Disk access
 (•) Virus scan           (•) Quick                ( ) Low level
 ( ) Integrity check      ( ) Encrypted (DEA)      ( ) Normal
 ( ) Both                                          (•) Network
                             Check for >>
 Scan for Viruses                                  Mode
 ( ) Common               Search file parts        (•) Fast mode
 ( ) plus Rare            (•) File header only     ( ) Secure mode
 (•) plus Exotic          ( ) Header & contents
                          ( ) Allocated space        Save options
 If a virus is found
 ( ) Ask user for action  Pick files to search
 ( ) Remove the virus     [√] Files in list        Also check:
 ( ) Delete the file      [√] Scan subdirectories  [ ] Memory
 (•) Note in log only     [ ] Random selection     [ ] System areas
                             10   per 100 files

    Start check              File list >>             Close

                                                        1999000099
F1 Help | Show Fast mode options.
```

Virus Buster is an extremely flexible package. The package's many screens allow the user to set up all aspects of the software's operation, from selecting scan type to editing search strings.

Having selected the partition size the *C:Cure* installation routine, BIPART, displays a chart showing the position of the four switches should be in. In my case, only switch one needed to be changed to the 'On' position.

BIPART then repartitions the physical drive and moves files around such that, in my case, just my stripped-down DOS directory and the old root directory remained on the 'new' drive C, everything else was now on drive D. This operation completed, the user is instructed to physically install the *C:Cure* hardware device.

Installation requires moderate knowledge of the innards of the PC, and involves disconnecting the ribbon cable from the hard drive and recconnecting it into the *C:Cure* box. The output from the box is then connected to the hard drive. The next step is to attach *C:Cure's* power connector to the drive's power connector. From this point onwards all commands sent to the drive are monitored by *C:Cure*.

Beside the connector that leads to the IDE drive there are two jumper pins. These pins control whether or not write operations are allowed to the protected drive. The manual suggested connecting them to a spare reset or turbo switch. As not every computer has a spare switch, *Leprechaun* provides a rear panel blanking plate to which a toggle switch has been fitted.

Safe - But Practical?

The PC was then rebooted. Try as I might, I could not write to drive C. I have an old test program which uses a number of different ways to write to disk drives, one of which uses direct port access to the drive controller. Each time I tried to write to the protected drive (even using non-standard methods) the alarm went off and a DOS ‘Abort, Retry, Ignore, Fail’ error message appeared on screen.

This is unsurprising, as *C:Cure* operates by monitoring *all* reads and writes to the IDE drive - when it is in its write protected mode, it is *impossible* to write to the C: drive. As the system relies on hardware (once installed there is no software component) it cannot be stealthed, tunnelled, turned off or otherwise subverted.

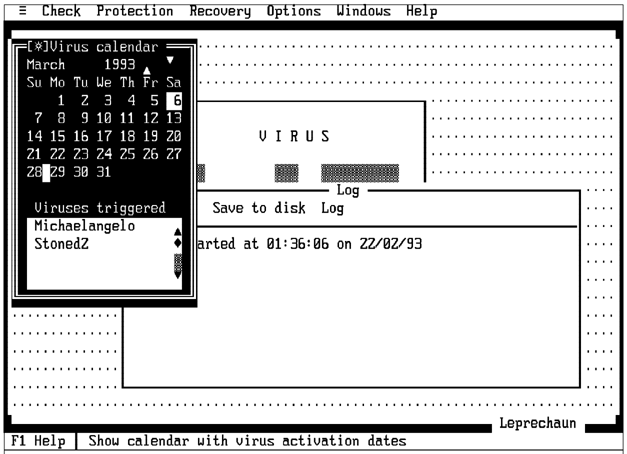
There is no such thing as a free lunch, and this device suffers from the inevitable trade-off between security and useability. Having half of your hard drive write-protected could prove to be something of a nuisance, though with careful setting up this can be minimalised. Eventually it all boils down to how much security you actualy need.

Virus Buster

The *Virus Buster* package basically consists of BUSTER, the main program which searches for viruses and does integrity checking; FIDO, a *Windows* program that ‘barks’ if you attempt to run an infected file under *Windows*; VBTSR, a small memory-resident checker; VINFO, the signature file editor; and GETSIGN, which produces search patterns for the TSR monitor.

The documentation has shrunk from 348 pages to a mere 77 and frankly is now little more than a quick reference card: there are vast areas of the product that are inadequately documented. For example, VINFO is scantily described and you need to know considerably more than the manual tells you to be able to use this utility correctly. It might be admirable to save an Amazonian rain forest, but this is taking things to extremes. As with the *C:Cure*, the documentation lets the product down.

The installation process is batch file driven using batch enhancement utilities that draw pretty dialogue boxes on screen - these programs (two in number) are briefly documented in the manual, so that the user can use them to jazz-up your own batch files.



```
March      1993
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Viruses triggered
Michaelangelo
StonedZ

arted at 01:36:06 on 22/02/93

Show calendar with virus activation dates
```

Yes - it is Michelangelo day once again! The pernicious spread of the virus calendar continues...

Once all the files have been copied, BUSTER starts up and scans all the hard drives it finds and creates its integrity database of file names, checksums, etc. It is at this point that you should leave the PC to its own devices - it can take an inordinate amount of time to complete. I installed the software on a *Compaq DeskPro* 386 (which houses all the review anti-virus software) which has two 40MB drives and 6MB of memory. Before scanning each drive, BUSTER scans all memory. You would think that when scanning multiple drives it would not repeat this memory scan for each disk drive it scans; unfortunately this is not the case.

Highly Configurable

BUSTER is a highly configurable piece of software - there are just too many options to list here, but the screen shots give some idea of what is on offer. Unfortunately, it is neither one of the faster nor one of the more secure anti-virus packages. It failed to detect all five infections of Spanish Telecom 1, the four of Spanish Telecom 2, and the Dark Avenger 2100, Frodo, Keypress, Mystic and SBC infections which are included in the ‘In The Wild’ test-set. It missed many more in the *Virus Bulletin* ‘Standard’ test-set which is very disappointing since this test set is unchanged since before an earlier review of this product. Viruses missed include Anthrax, Diskjeb, Fellowship, Int 13, Jerusalem PSQR, Liberty, Number 1, Raubkopie and Sentinel 1. It also failed to detect *any* of the Mutation Engine samples. These results are simply not good enough.

Generic Protection

The generic side of BUSTER is far better. The software offers two distinct checksumming methods, which it calls ‘Quick’ and ‘Encrypted’ though the documentation provides no details on the algorithms employed. Although not particularly speedy, the checksummer is highly accurate and spots the following: changes to file size; changes in file header; changes to the content of files; date/time differences; changes of file attributes; changes to directory; and changes to physical location on disk. In short, it provides a good deal of security, and offers more checks than other competing products. It certainly spotted all the changes I made to files on the disk, including single-byte changes within files. An impressive result.

VB - TSR Components

VBTSR is the memory-resident monitor that is designed to prevent write operations to the boot record writes to COM and EXE files; and, prevents writes to read-only files by preventing alterations to the read-only attribute. It can optionally disable warm boots using Ctrl-Alt-Del and execution of programs from floppy disk. Curiously, it does not use the same virus signature file as BUSTER, it has its own very small subset of signatures which are stored as plain ASCII text in a different file.

I am wary of monitors that prevent write operations to COM and EXE files because quite a large number of programs write to their own executable quite legitimately to update configuration information and such programs cease to work with these monitors installed. *Leprechaun* has thought about that one and provides a simple utility program (GETSIGN) which extracts signatures and optionally appends them to VBTSR's signature data file.

The final part of the package is VINFO and is a new addition to *Virus Buster*. This program provides access to BUSTER's virus signature database and allows you to add, modify or delete virus signatures - even those provided by the manufacturer. I applaud the ability to be able to add, edit or delete *user*-supplied signatures, but I question the wisdom of providing users the means to compromise the integrity of the software by modifying or deleting the supplied set of signatures.

Virus Buster also contains a number of built-in utilities of varying degrees of usefulness, ranging from creating a 'rescue diskette' (highly useful) and viewing CMOS information (useful) to a Virus Calendar (for trivia buffs).

Conclusion

I first reviewed *Virus Buster* just over a year ago and determined that it was basically a good product that needed some finishing touches. Unfortunately, something appears to have gone wrong since the last review, as the results of the scanner test were poor. This needs to be addressed if *Virus Buster* is to be recommended.

C:Cure on the other hand, does exactly what it claims to do, and does it well. If *C:Cure* is installed and used correctly, boot sector viruses cannot spread, and a clean boot is guaranteed. For a 'mission critical' PC this product may be a very good choice. The weakest point of the product is its confused documentation - this needs to be tidied up.

VIRUS BUSTER

Scanning Speed

Hard Disk:

Turbo Mode (218 Kbytes/sec)	1 minute 14 secs
Secure Mode (44 Kbytes/sec)	6 minute 24 secs

Floppy Disk:

Turbo Mode	5 secs
Secure Mode	14 secs

Scanner Accuracy

'VB Standard' Test-set[1]	327/364	89.8%
'In The Wild' Test-set[2]	113/128	88.2%
'MtE' Test-set[3]	0/1536	0.00%

Technical Details

Product: *Virus Buster*

Version: v4.00.14

Author: *Leprechaun Software Limited*

Distributor:

Telephone:

Fax:

Price: *Virus Buster, C:Cure*

Test Hardware: All tests were conducted on an *Apricot Qi486* running at 25Mhz and equipped with 16MB RAM and 330MB hard drive. *Virus Buster* was tested against the hard drive of this machine, containing 1,645 files (29,758,648 bytes) of which 421 were executable (16,153,402 bytes) and the average file size was 38,370 bytes. The floppy disk test was done on a disk containing 7 files of which 3 (25,508 bytes) were executable.

For details of the test-sets used please refer to:

[1] Standard test-set: Virus Bulletin - May 1992 (p.23).

[2] 'In The Wild' test-set: Virus Bulletin - January 1993 (p.12).

[3] 'MtE' test-set: Virus Bulletin - January 1993 (p.12)

PRODUCT REVIEW 2

Dr Keith Jackson

‘The *McAfee* Utilities’

This month’s review examines the various anti-virus programs that are offered by *McAfee Associates*, as they have not been looked at by *VB* since April 1991. These arrived as a whole suite of programs, and I will mention most of them in this article (although for brevity the *NetWare* versions are not discussed). *McAfee’s* programs follow a common design concept that three separate programs are required (each for a specific task) - a combined scanner and integrity checker (VIRUSCAN) is used to check that any disk(s) are free from viruses, a memory resident utility (VSHIELD) can be used to ensure that things remain that way, and a disinfection program (CLEAN-UP) for when things go wrong.

All of the *McAfee* programs were provided on 360 Kbyte, 5.25-inch floppy disks, and I was pleased to see that these disks were provided in write-protected form. However, the actual type of disk media used is irrelevant, as the main method of distribution used for these programs is shareware, where a suite of files is usually obtained in compressed form from a BBS or the Internet.

Documentation

The documentation that accompanies each of the *McAfee* programs is contained in ASCII format in a file held on disk. Although this is mandatory for a shareware product, it is possibly the weak point of the whole package. The documentation contains no Table of Contents and no Index - indeed very little structure at all. Nevertheless, it does contain a fair description of how to use the product.

Details of all the viruses currently known to VIRUSCAN are provided in a separate file, and in common with the operating instructions for the *Windows* based version of VIRUSCAN, this is available on-line for inspection via the *Windows* help system. The documentation describes ways in which foreign language support can be installed (specifically French and Spanish), but the version provided for test did not contain examples of these files.

Installation

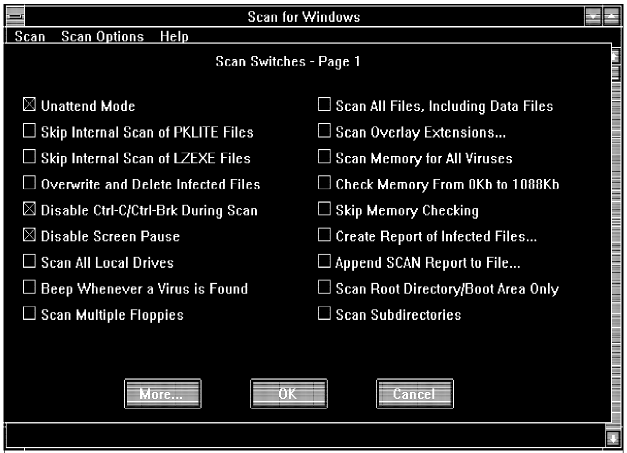
All of the various *McAfee* programs are very simple to install and, with the exception of the *Windows* scanner (WSCAN) which comes with its own installation program, installation is merely a matter of copying a set of files to the desired location. Following my usual practice, I tried various methods and locations for installation, but did not come across any problems in the procedure.

High Integrity

Every executable program distributed with the package is provided with a written description of two validation codes, and a program capable of verifying that the codes are intact (and therefore that the files are undamaged/unaltered). These validation codes are useful, but given that the product is distributed electronically they can be troublesome to obtain (it is clearly useless distributing the validation codes in this way). There has been warning of Trojanised versions of *McAfee* programs before (eg v65 March 1990, v70 October 1990), and this no doubt will happen again when some maniac decides to alter one of the programs, and add a few new ‘features’.

VIRUSCAN can be used to add validation codes to a program. It has two basic methods of operation: one of which calculates a simple 10 byte validation checksum, the other calculates an enhanced 52 byte validation and recovery data checksum. Both can be appended to files, but the latter can also be stored in a separate log file for safekeeping. VIRUSCAN maintains a file containing software known to amend its own executable (eg many of *Borland’s* compilers, *WordStar*), and files that already use this technique to validate themselves (eg the PKZIP compression program). As this ‘exceptions’ list is an ASCII file, it can be extended as desired by the user.

Validation and subsequent verification works quite well, though I do query the sense of deliberately introducing changes to executable files - this could cause problems when the package is used with other anti-virus software.



Scan Switches - Page 1

☒ Unattend Mode
☐ Skip Internal Scan of PKLITE Files
☐ Skip Internal Scan of LZEXE Files
☐ Overwrite and Delete Infected Files
☒ Disable Ctrl-C/Ctrl-Brk During Scan
☒ Disable Screen Pause
☐ Scan All Local Drives
☐ Beep Whenever a Virus is Found
☐ Scan Multiple Floppies

☐ Scan All Files, Including Data Files
☐ Scan Overlay Extensions...
☐ Scan Memory for All Viruses
☐ Check Memory From 0Kb to 1088Kb
☐ Skip Memory Checking
☐ Create Report of Infected Files...
☐ Append SCAN Report to File...
☐ Scan Root Directory/Boot Area Only
☐ Scan Subdirectories

SCAN for *Windows* simply provides a pretty GUI for the DOS scanner, at the cost of a certain amount of speed.

Although I prefer to keep my executable files intact, I can think of many environments (universities and large networks to name but two) where this method of file validation would work well.

Scanning

VIRUSCAN is also capable of scanning files for known viruses and can be used to check the entire system, an individual disk, a sub-directory or an individual file. If a virus is found, the name of the virus is displayed, along with an identifier called the ‘Virus ID’ The latter is needed by the CLEAN-UP program to eradicate the virus. VIRUSCAN will perform both an internal and an external scan on programs that are dynamically compressed with either of the LZEXE or PKLITE utilities. The compressed file will first be scanned in its raw form, then scanned again for an internal infection.

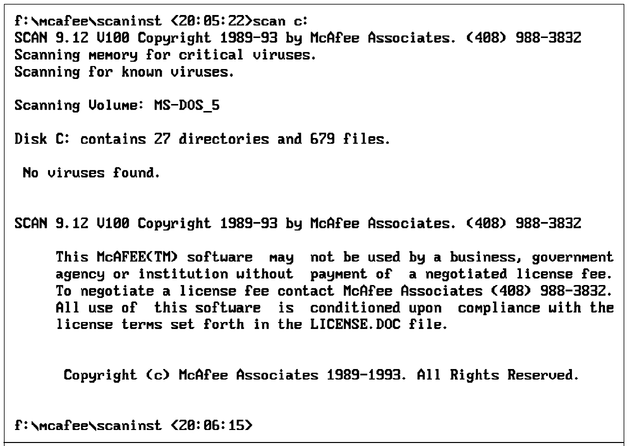
One of VIRUSCAN’s numerous options tells it to search memory for all of its known viruses rather than just a selected list (142 viruses long) of those which are known to cause problems if memory-resident. Activating this memory search slows down scanning somewhat, but if the system has not been booted from a clean system disk is advisable.

It is quite difficult to decide exactly what to quote as an example time for the rate at which VIRUSCAN can inspect a disk - there are so many options that can be set by the user which affect scanning performance that I will have to mention quite a few figures.

The DOS scanner running under DOS took 48.9 seconds to inspect my hard disk containing 24 Mbytes of files (678 files). Introducing the *Windows* version increased the scan time to 54.0 seconds. By omitting memory checks and not searching inside compressed files, the scan time could be reduced to 42.1 seconds, but curiously the greatest decrease in scan time was obtained by instructing VIRUSCAN to stop writing the name of the file being scanned on the screen. This reduced the scan time to 30.3 seconds. For comparison purposes, *Dr Solomon’s Anti-Virus Toolkit* performed the same scan test in 14.9 seconds, and *Sweep* from *Sophos* took 55.7 seconds for a complete scan, and 14.0 seconds in quick scan mode.

Scanning Accuracy

VIRUSCAN proved to be very good indeed at detecting viruses, as it detected all of the viruses in the test-set bar one (Rat). There are currently 1561 unique viruses known to VIRUSCAN and identified in the file VIRLIST.TXT, and this is an order of magnitude larger than the total of 129 viruses that VIRUSCAN knew about in the previous *VB* review (*VB* September 1990, p.22 - 23.)



```
f:\mcafee\scaninst <20:05:22>scan c:
SCAN 9.12 V100 Copyright 1989-93 by McAfee Associates. (408) 988-3832
Scanning memory for critical viruses.
Scanning for known viruses.

Scanning Volume: MS-DOS_5

Disk C: contains 27 directories and 679 files.

 No viruses found.


SCAN 9.12 V100 Copyright 1989-93 by McAfee Associates. (408) 988-3832

     This McAFEE(TM) software  may  not be used by a business, government
     agency or institution without  payment of  a negotiated license fee.
     To negotiate a license fee contact McAfee Associates (408) 988-3832.
     All use of  this software  is  conditioned upon  compliance with the
     license terms set forth in the LICENSE.DOC file.


      Copyright (c) McAfee Associates 1989-1993. All Rights Reserved.


f:\mcafee\scaninst <20:06:15>
```

The DOS based version of the scanner is minimalist in its approach - a must for all command line lovers.

When tested against 1024 Mutation Engine samples, then VIRUSCAN performed flawlessly, detecting all of them. There was a high preponderance of different virus names reported, but this probably reflects different naming conventions used on either side of the Atlantic, rather than mistakes in virus detection.

Memory-Resident Utility

VSHIELD is the *McAfee* memory-resident program that prevents viruses from entering a computer system by monitoring and scanning programs as they are loaded. Features that are monitored and detected include validation codes (as added by VIRUSCAN), virus signatures, and allowing only certain (certified) programs to be run. The difficulties that can be encountered on running this type of program on a network, such as overheads added to the file loading process and removal of a virus if one is detected, are all thoroughly discussed in the documentation.

VSHIELD is supplied in two versions, the simplest of which requires only 6 Kbytes of memory, but is only capable of checking validation codes. Increasing levels of protection are offered by the more complicated memory-resident monitor which can require anything up to 40 Kbytes of memory, although all but a few hundred bytes can be loaded into upper memory if desired. There is a version of VSHIELD available for networks (unsurprisingly called NETSHIELD).

Although VSHIELD seemed to work exactly as described, and I found no faults during testing, I am at a loss to see why the user has to put up with two essentially different versions of the same utility. The command line switches used by the two options are completely different - the simplest version has a plethora of command line switches and the more complicated version just two, yet the functionality offered by the programs overlaps considerably. Why? Surely there is a need for further development here in integrating these two options into a single utility which decides at execution time which portions to install as memory-resident, and which features to activate. Why should the user have to do all the work?

Removing Viruses

CLEAN-UP is the name of the *McAfee* program that removes viruses, and attempts to repair or delete infected files. It identifies the virus which is to be removed by means of the 'Virus ID' (a short name) provided by VIRUSCAN. Therefore as described above, VIRUSCAN must have been executed first to find out what type of virus infection is present (if any). Careful note must be taken of the details of any viruses detected, though as long as a report is always written to disk, then the information required by CLEAN-UP will be available for future reference in a disk file. This is actually more of a trouble during testing rather than during a 'real' virus outbreak where only one virus is likely to be found, and its Virus ID will very soon be emblazoned on everyone's heart.

Personally I would always reinstall software from 'known clean' master disks, rather than try to remove a virus, but I accept that this is not always possible; a 'disinfection' program can often be useful. The documentation which comes with CLEAN-UP claims that is available to remove any of 104 viruses and restore the original non-infected program. I confess that I did not test all of these, but the ones I looked at performed correctly. Suitable warnings are contained in the documentation about using this program, and I would at least caution people to use it in conjunction with some method of program verification.

Other Programs

TARGET is a well thought-out utility program which is a 'multi-purpose file finder and manipulator.' It can initiate actions such as virus scanning, copying, renaming, deleting and archiving on any specified group of files. The specification process can be quite complex, including such features as only looking at those files which are smaller than a given size. TARGET even has facilities to show the amount of disk space that is wasted by storing many small files individually rather than combining them into a single large archive.

The design of this program is such that from the start it assumes that the user knows what commands to use (no prompting front end is provided) but this should not prove too difficult as there are not too many to learn. TARGET proved to be very fast indeed, which makes a refreshing change after some of the ponderous *Windows* nonsense available. I have never come across this program before and I like it enough to think about using it on a permanent basis. [*High praise indeed. Ed.*]

The final disk which was provided for test purposes contained a suite of programs known collectively as the 'Configuration Manager', which provides secure access, boot protection, configuration locking, and optional hard disk password protection. Be warned that at least one of these programs installs a new master boot record on the hard disk. Indeed, this is mandatory if secure access control is offered without any help from external hardware.

Conclusions

With the exception of gaining a *Windows* front-end, the scanner program has not changed much structurally since the April 91 review. The scanning time is acceptable, and its detection capability is very good.

The viruses known to VIRUSCAN are updated very frequently (every couple of weeks), and are distributed around the world via various electronic conferencing systems and bulletin boards. At times, the updates have arrived so frequently that the electronic conferencing system that I frequent (CIX in the UK) has been full of messages from people complaining (somewhat tongue-in-cheek!) that they cannot keep up with the new releases.

If you want a virus detection system which provides a pretty GUI, and large books of documentation to use as bedtime reading, then this is not the product for you. Its interface is basic, even old-fashioned in appearance, but it works well enough. Performance counts most in my eyes as far as keeping out viruses is concerned, and for virus detection VIRUSCAN is hard to beat.

Technical Details

Product: *McAfee* Anti-Virus Programs

Developer (and Vendor): McAfee Associates, 3350 Scott Blvd. Bldg. 14, Santa Clara, CA 95054-3107, USA.
Tel. +1 (408) 988-3832. Fax. +1 (408) 970-9727.

Availability: PC or compatible. Requirements vary.

Version evaluated: 9.2 V100

Serial number: None visible

Price: SCAN

Hardware used: (a) 33MHz 486 PC, with one 3.5 inch (1.44M) floppy disk drive, one 5.25 inch (1.2M) floppy disk drive, and a 120 Mbyte hard disk, running with MS-DOS v5.0, Stacker v2.01, and Windows 3.1 (b) 4.77MHz 8088, with one 3.5 inch (720K) floppy disk drive, two 5.25 inch (360K) floppy disk drives, and a 32 Mbyte hard card, running under MS-DOS v3.30

For details of the test-set used please refer to *Virus Bulletin*, December 1992, p.22

CONFERENCE REPORT

IVPC '93 West

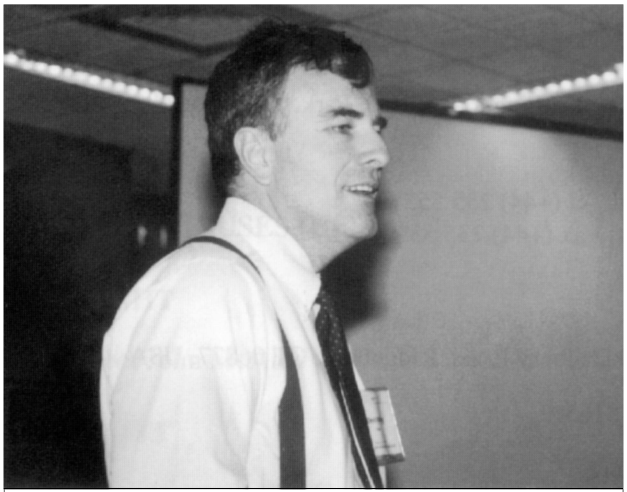
It was with little heartbreak that *VB* decided to attend the *NCSA's 2nd International Virus Prevention Conference and Exhibition* in San Francisco. Sunny California seemed like an infinitely better offer than another drizzly day in Oxfordshire! After a long and gruelling flight from England, there was nothing better than to relax in the pleasent surroundings of the conference venue, drinking ice cold beer and watching the sun set above the San Francisco skyline.

The trip from the airport had not been without event. 'Jeez, this country hasn't got no laws anymore - everyone is free to do as they please' droned the cab driver. 'See those flats over there - they're crack houses.' The rest of the trip in to the centre of town was filled with other such uplifting sights. 'Freedom, stuff it.' he concluded, 'just give me law and order!'

The idea of freedom within American society reared its head so many times during the conference that the idea of freedom of speech became very much a conference theme.

Your Fist, My Nose

On the evening before the conference proper there was a panel session to discuss possible ways to curb the virus problem. The discussion very quickly centred about whether the sale of virus code was, or even should be, illegal. An easy question to answer for those on the European side of 'the pond' but apparently a much harder one for Americans.



David Stang *(ICSA)* speaking on what should be in a virus buster's toolkit.

Intellectual freedom is a fine goal, and is certainly worth fighting for. However, as Dr Solomon pointed out 'Your freedom to swing your fist stops at the place my nose starts.' While this definition is somewhat too vague to be turned into law the sentiment behind it is absolutely correct - by distributing virus the freedom of the PC user community is gradually being encroached upon. There is no question whether virus authors and distributors should be able to hide their unsavoury practices behind the shining banner of 'freedom of speech'.

Bug Guns, Small Numbers

The conference proper was attended by many of the big names in the anti-virus industry, with representatives from nearly all of the big vendors. Unfortunately, the paying public had not turned out in such large number: the conference attendence can only be described as dissappointing.

John McAfee gave the first talk and brought users up to speed with a brief guided tour through the history of computer viruses. If you were not involved in the industry at the end of the Eighties, or wanted to be reminded of what really happened, this was the talk for you.

The most interesting talk given at the conference was presented by Winn Schwartau, who discussed virus prevention as part of the bigger data security issue. Viewing a computer virus as 'a small mobile hacker' he discussed how many companies could benefit from installing even the most basic security measures. The idea that system security itself seems to have been forgotten somewhere along the 'my scanner detects more viruses than your' argument, and this return to basics was a welcome breath of fresh air.

At conferences like this the most interesting discussions often take place in the bar, where jaded researchers, now devoid of their marketing staff 'minder' gleefully discuss future 'nightmare scenarios' and other such happy thoughts.

Just to provide a ray of hope to delegates Frirdik Skulason's talk took an all too realistic look at what the future may hold in storw for computer users. Increased polymorphism and readily available virus construction toolkits were just two of the horrors lying ahead. However, to cheer us up still further, he explained that there were plenty of other potential traps ahead, but he 'did not want to mention them in public in case he gave anyone ideas!

As the *VB* editor sat back and relaxed over a Gin and tonic on the flight home, there was little to do but sit back and think about the complex problems of developing legislation against virus writing and distribution - well, that and sunning himself on all those lovely Californian beaches!

END-NOTES AND NEWS

***The Xtree Company* is set to leave the anti-virus industry.** *Xtree*, the makers of *ViruSafe*, *ViruSafe/LAN* and *AllSafe* has announced that the company ‘will discontinue publishing and/or developing any anti-virus and/or security products... All existing users who purchased anti-virus and/or security products from *Xtree* will be supported for one year, ending January 31st, 1994.’ This move comes as no surprise to seasoned observers who have been predicted a slimming down of the number of anti-virus software manufacturers. Tel. +1 800 964 2490 ext. 3.

A *Department of Trade and Industry* report carried out by *Cooper & Lybrand Deloitte* says that **the *1990 Computer Misuse Act* is suffering from poor awareness** and patchy understanding. According to the report many firms believed that a prosecution under the Act would indicate a weakness in their business systems to shareholders, potential customers and competitors which could undermine confidence in them.

The *Computer Security Specialist Group* is holding ‘The Specialist IT Security Conference’ at the *Penns Hall Hotel* near Birmingham. The conference, held on 12th - 13th March, aims to increase awareness of the dangers which can arise from computer insecurity. For further information contact Cliff Potter. Tel 0895 631039 (evenings/weekends).

The *Federation Against Software Theft* is targeting UK electronic bulletin board systems in a bid to stop the spread of computer pornography, virus programs, pirate software and the illegal use of public telephone networks. ‘Pirate bulletin board operators are a significant problem in the UK and despite what the operators say there is no excuse for their action’ said Bob Hay, *FAST* chairman.

***S&S International* has released a 1993 Virus Calendar** which highlights ‘virus free days predicted for 1993.’ For a free copy of the calendar or further information contact Jo Wheeler. Tel. 0442 877877.

Total Control have announced the launch of a new service, the *Virus Information Service Bulletin Board*. The board aims to provide details about viruses as well as free cure programs for some of the more common viruses. For further information contact *Total Control*. Tel. 0488 685299 or call the BBS directly on 0488 681291.

A new virus is reported to be in the wild in Germany. The sample arrived just as *VB* was going to press so it has not been fully analysed. The virus is 4000 bytes long, polymorphic and uses stealth techniques. It has also been reported specifically to evade *Central Point Anti-Virus*, which is believed to be included as part of *MS-DOS* Version 6.

A good piece of anti-virus policy has been passed on to users from *International Data Security*. In a flyer for *McAfee Utilities* users are told that if the checksum of any file they ship differs from those stated ‘it may have been damaged or have options stored in it with the /SAVE switch. Run the program with only the /SAVE option to remove any stored options and then re-run VALIDATE.’ Hmmm...

VIRUS BULLETIN

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

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

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.