ISSN 0956-9979 JULY 1999

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

Editor: **Francesca Thorneloe**

Technical Consultant: **Fraser Howard**

Technical Editor: **Jakub Kaminski**

Consulting Editors:

Nick FitzGerald, Independent consultant, NZ
Ian Whalley, IBM Research, USA
Richard Ford, Independent consultant, USA
Edward Wilding, Maxima Group Plc, UK

IN THIS ISSUE:

• **The top ten?** This month's comparative is relatively limited. Only ten products were submitted for testing in the latest *NetWare* tests. Check out the results on p.18.

• **Talking shop:** *NAI's* Peter Morley gets topical, raising the issue of the generic repair of file viruses. Catch his tutorial-style article on p.10.

• **Desert storm:** A coordinator of the *Syrian Computer Society's Information Security Interest Group* (*ISIG*) shares his views on the virus situation in his country on p.12.

• **Out and about?** A rash of new viruses has broken out in the last few weeks. Our news pages mention the most predominant ones, starting on p.3.

CONTENTS

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50** No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

COMMENT

Cyber Terroristicexpialidocious!

Do you know what a Ring 36 process is in your CPU? Chances are that you don't know how many volts are running through your ignition system in your car either. When that shifty looking mechanic tells you that he will have to replace the Flux Capacitor in your car because the Finnican Pin is out of alignment you'll wish you did know. So, if your computer is acting weird and a technician tells you that a virus is jumping to Ring 36 on your CPU to avoid detection and the only thing you can do is buy a new motherboard, what do you do? Well, just like the Flux Capacitor, there is no such thing as Ring 36. Shouldn't you know that?

"cyber terrorism – isn't that a great phrase?"

Have you seen those great *NAI* billboards in Silicon Valley with the pretty woman in the short skirt or the tattooed and pierced guy? They read 'While you are watching me, who's watching your network?' Or that great commercial with the two grungy looking twenty-somethings who email some poor company's executive salaries to all the employees? Scared yet? You better hurry and buy some very expensive security and anti-virus software. That is the hidden message, isn't it? Albeit not very well hidden. How about cyber terrorism – isn't that a great phrase? Well, no actually, it's not. It has been very loosely used lately to describe virus attacks on computers. I can only shake my head in disbelief at the irresponsible use of that phrase. As much as the IT community needs an education on computer security and viruses, scare tactics are not the way. They make for great advertising but they also try to lead you to a false sense of reliance on one solution. You must educate yourself or consult someone who has the education already.

A lot of companies tend to treat anti-virus as a peripheral issue. I worked for a company with 7,500 client machines and about 50 servers. My title was Senior IS Technician and I babysat a poorly designed network. The total number of technicians was around 20 to 25 in any given month. Each technician's salary was between $40K and $80K; I had five people working for me whose salaries totalled $300K. The average number of man-hours per week spent on anti-virus issues by my staff alone was about 125 hours. I'll do the maths – $187,000 were spent in labour alone every year to manage the enterprise anti-virus implementation. This does not count the value of lost data and logistical costs to reconfigure infected machines that had their hard drives erased. Upper management saw this as an acceptable cost to have anti-virus protection. The particular virus software being used by the company had no means of automatic updating of definition files or of remote administration of the machines. Quite simply, the installations of AV on the user machines were either not there or not updated to the most recent virus definitions. So even the people you employ to watch the network don't know what they are doing sometimes. Management of this company was told that the AV software they bought would detect and remove all known viruses.

Who is at fault here – the consumer who blindly purchases the shiny, candy-like box that promises protection from viruses, or the company that withholds certain key technical information that might not make them appear as good as they would like? You buy an alarm system for your house. While you are at the movies someone breaks in and steals your wife's jewellery and your brand new golf clubs. What happened? The alarm company's operator contacted the police after the alarm went off. They checked out your house about 15 to 30 minutes after the alarm went off. This does nothing to get your valuable possessions back. Is the alarm company at fault for not advising you to lock up your valuables even though you had an alarm? Or are you at fault for not buying a safe?

Anti-virus companies provide protection for your valuable possessions. These days digital information is often much more valuable than anything tangible. If you don't educate yourself, believe me when I tell you that no-one else will. Whether the AV company is responsible for educating you or not is a moral or financial question depending on your point of view. If they tell you everything they know you will likely not spend as much money on products because of the cost of educating yourself. Also you will find yourself in a position where you don't need to rely on one solution. So I ask you, 'Who is watching your network?' Shouldn't you be?

D J Forman, Data Fellows Inc, USA

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

NEWS

Turning Nasty

In March of this year we saw the first virus to take full advantage of email to accelerate its spread. WM97/Melissa exploited the features of MAPI and *Outlook*. The recently discovered ExploreZip uses MAPI commands too and spreads between organizations via email attachments in much the same way.

Melissa, though, was a nuisance but relatively harmless. ExploreZip is particularly nasty. It truncates all files with extensions C, H, ASM, CPP, DOC, XLS, and PPT to zero length. Not only is one machine affected, but also all networked drives writable from that machine. To make matters worse, it happens straight away.

ExploreZip spreads itself via email by searching an infected user's inbox. It will reply to unread mail with a plausible message and an attachment called ZIPPED_FILES.EXE. If the recipient launches the attachment, a message box appears saying that the file cannot be opened, because it is not a valid archive. Once the program has been run it will copy itself to the likely-sounding name EXPLORE.EXE. Then changes are made to either WIN.INI, on *Windows 9x*, or the registry, under *NT*, so that it will be run again each time the machine is rebooted.

As well as email the worm uses networks to spread. It can infect any machine that has a WIN.INI file that is writable from an already affected machine. ExploreZip must certainly be one of the nastiest viruses yet. It shatters any illusions that viruses through email are any kind of joke. A full analysis will feature next month ▮

Lines to Ponder

A new virus {Win32, W97M}/Coke.22231.A (also known as W32/WM/Cocaine.22271 and W32/Vecna.22231) was sent to several anti-virus researchers at the end of May. Far from stable, Coke has all a virus needs to become in-the-wild. Making use of the features Win32 provides, it follows some new directions in virus writing – cross-infection and Internet-oriented replication routines. It probably has the most complex polymorphic engine used in a Win32 virus and will cause a huge headache for anti-virus developers should it go wild.

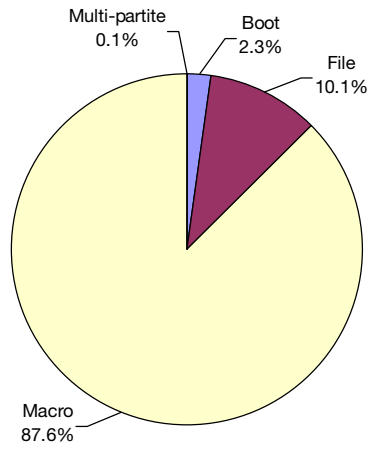
Coke works on *Windows 95* and *98* (its bugs will prevent it from working too long on *Windows NT*, but it is able to infect files under *NT*). The virus infects both *Office 97* documents and PE files and it is polymorphic in both cases. In the latter case, it is a multi-threaded, slow polymorphic virus and it uses multi-layered polymorphism and encryption. Its first polymorphic decryptor is placed in the code section of the PE files in eight pieces in a manner similar to that used by One_Half (see *VB*, October 1994, p.9).

Prevalence Table – May 1999

Virus	Type	Incidents	Reports
ColdApe	Macro	692	19.0%
Ethan	Macro	596	16.4%
Pri	Macro	319	8.8%
Cap	Macro	308	8.5%
Win32/Ska	File	249	6.8%
Class	Macro	234	6.4%
Marker	Macro	227	6.2%
Temple	Macro	115	3.2%
Appder	Macro	104	2.9%
Npad	Macro	97	2.7%
Melissa	Macro	84	2.3%
Concept	Macro	76	2.1%
Munch	Macro	48	1.3%
Footer	Macro	46	1.3%
Win95/CIH	File	46	1.3%
CopyCap	Macro	39	1.1%
Form	Macro	38	1.0%
Laroux	Macro	33	0.9%
AntiEXE	Boot	27	0.7%
Walker	Macro	26	0.7%
Parity_Boot	Boot	18	0.5%
Groovie	Macro	14	0.4%
Win32/Nuker	File	13	0.4%
Others [1]		191	5.2%
Total		3640	100%

[1] The Prevalence Table includes a total of 191reports across 56 further viruses. A complete listing can be found at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

In PE files, the polymorphic engine is initialized using the current day, resulting in slow mutation. The random number generator checks if there is a breakpoint where the routine is called, making analysis more difficult. It checks if the system date is eight months after the current file's infection time – if so, the payload routine is called.

In *Word*, Coke disables confirmations for loading modules that contain macros (virusprotection), upconversions (confirmconversions) and saving the NORMAL.DOT file (savenormalprompt). Also, alerts are disabled. Then, the virus drops a randomly named BAT file in the root of the C: drive. This batch file drops an infected executable, named W32COKE.EXE, executes it and remove all files involved in infection: C:\W32COKE.EX, C:\COCAINE.SRC, C:\COCAINE.SYS, C:\W32COKE.EXE and the batch file.

The virus is able to mass mail itself to several locations. This can happen when the user sends email. The virus hooks a MAPI function and adds an infected attachment to all out-going mails from the machine. Also, the virus sends an email with an infected attachment each time the user visits a web page which has a 'mailto' reference. The virus keeps a CRC32 list in a file on the machine about the addresses it sent itself to, ensuring that only one copy will be sent to each recipient ▮

A Walk in the Park

According to the developers of *AVP*, W32/PrettyPark (a virus with many aliases) was discovered on 1 June, although some reports date it earlier than that. It is an interesting piece of code for a number of reasons, not the least of which is how to classify it. The file seems to have originated in France and some of its code is specific to a French *Windows* installation.

When first executed PRETTYPARK.EXE drops a file named FILES32.VXD into the *Windows* system directory. If there is an error it will run a screensaver to mask itself, otherwise it will run a different screensaver under French *Windows*. It then reconfigures the *Windows* registry so that this file is executed every time any other EXE file is run. Thus, it places itself in the 'Executable Path' and becomes a companion virus! It tries to distribute itself using a range of networking mechanisms, including IRC and email, reading the mail addresses it sends to from the *Windows* address book. The icon associated with FILES32.VXD is a picture of a character from the cartoon series 'South Park' ▮

Don't Quote Me

WM97M/Quoter.A is a polymorphic encrypted macro virus. The polymorphism is old hat, in that it inserts random labels and variables into its code. The encryption is new though – the structure of the code besides the polymorphism is a variable containing the encryption key, lines of comment that contain the encrypted virus code followed by a decryption loop. The only code seen is the decryption loop, making the virus slip though the current generation of heuristics. The virus is a bit on the slow side, but on a high spec machine would probably not be noticed. Its payload is interesting. Unless the virus is active on the machine on which the document was infected, it will deny access to the documents' contents, displaying either nothing or the text:

```
Warning:
This is a protected document!
You have to activate the macros in the
document before you can see the document's
text! So, close the document, open it again
and click on 'Activate macros'!
```

If the Quoter virus is active the contents can be seen and a quick look at the decrypted code indicates where the documents are held. The virus creates a directory under %windir%\system\vmm32 called 'flitnic' where it stores the infected original files.

The virus writer has a couple more tricks up his sleeve. The virus will then proceed to rename the folder and add some non-printable characters to the name. This makes the directory an illegal one under *Windows* and *Explorer* cannot manipulate it. However, it can still be renamed from the command prompt, which the virus does each time it needs a file. To disinfect this virus you have to remember your old DOS directory manipulations and do some file copying ▮

Far from Perfect

To date, users of *Corel WordPerfect Office Suite* have been safe from the threat of macro viruses thanks to the use of *Corel's PerfectScript* as the scripting language for recording macros. However, the developers have changed all this with the recently launched *WordPerfect Office 2000 Suite* which, amongst its new features, supports *Microsoft's* Visual Basic for Applications.

Last month we reported the occurrence of the new virus CS/GaLaDRiel, written in *Corel SCRIPT* and capable of infecting *CorelDRAW*, *Corel PHOTO-PAINT* and *Corel VENTURA* files (see p.6 of this issue for a full analysis). The threat of existing and new viruses written in VBA will no doubt present more of a worry to *Corel Office 2000* users, especially if infected *Microsoft Office* files infect the *Corel* environment successfully. Watch this space ▮

Pots and Kettles?

In amongst its daily mail, *Virus Bulletin* recently discovered a Y2K questionnaire from a concerned UK-based developer of speech and language software. The form was aimed at Managing Directors and was centred around the immediate implementation of preventative corporate action.

So far, so ordinary. However, glaring from the top of the leading page was the header: YEAR 2000 COMLIANCE. *VB* humbly suggests that this particular speech and language software developer look to their proofers before sending out 'confidence-boosting' media to their suppliers ▮

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

LETTERS

Dear Virus Bulletin

[*When ColdApe emerged, Nick FitzGerald set up a message of explanation and instruction to be sent to the victim on each infection. Some people welcome it, others are not so grateful as these two typical responses illustrate. Ed.]*

You've Got to Give a Monkey's

I have been receiving emails every morning from your people for the last few months or so. Not only is it the most annoying thing I have ever experienced but it is also an invasion of privacy.

We have an IT section here which is more than capable of dealing with any problems that may arise. I initially requested that I be removed from your distribution list politely but you seem to have a problem understanding plain English. The abusive emails that I returned were a means of firstly getting your attention and also to give you a taste of your own medicine. BTW, are you also aware of the pathetic little email message that is automatically replied after your email from 'Nick Fitzgerald' has been received? Why on earth anyone would want to deal with people that have sex with monkeys is beyond me. Perhaps you need to have a serious look at yourselves.

Our PCs are all about to be upgraded and tomorrow is my last day here – I could not give two %*#~! about ColdApe. Please refrain from bothering me again. I feel sorry for others that are being harassed by this type of utter crap.

Anon
USA

Oops! Sorry about this I didn't realise what was really happening. I didn't understand the reason for receiving the emails from yourselves. I will go get some anti-virus to hopefully disinfect my PC. Thanks for your assistance in this matter.

Anon
UK

The Truth is Out There

Firstly, congratulations on your web site – very informative for us mere mortals new to virus recognition. Melissa and CIH I have personal knowledge of, being a former acting senior tech in gateway Europe. Your figures do not come close to the real infection of this virus. Having finished there on 1 May I can tell you that I had at least 20 cases of CIH to fix. In one case it actually affected the hdd controllers causing the client major heartaches. As regards Melissa, complete infestation of the gateway occurred, reaching companies not just affiliated with gateway Europe but to Australia where my cousin, also involved in the same industry, reported mass infection. Once again, thanks for the page. Will check in regularly for the updates.

Eoin O'Flaherty
Contract Tester Win2K/ie5
Republic of Ireland

Getting Wild

Following the publication of my article in the June issue of *Virus Bulletin*, it came to my attention that Vesselin Bontchev's negative comments regarding the competency of people like myself and our value to The WildList were not made on behalf of The WildList at all; in fact, *The WildList Organization* has recently (prior to the article) started adding qualified corporate reporters. Apologies to *The WildList Organization*!

David Phillips
The Open University
UK

It's No Flitnic!

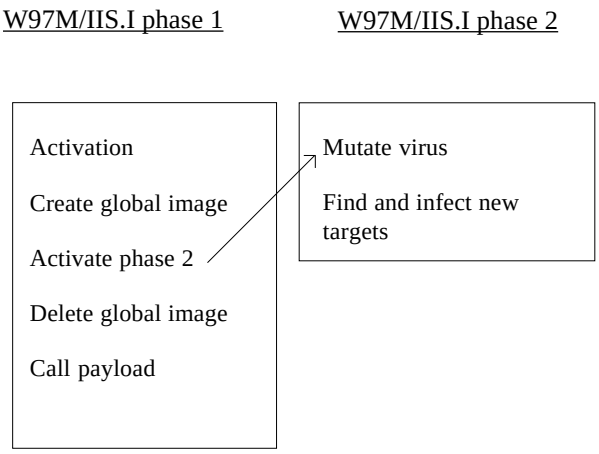
I thought I'd share my short analysis of W97M/IIS.I (also known as Flitnic), a relatively new macro virus which uses a novel way of activation and infection. The virus consists of a single VBA module, which is the class module in documents, and Flitnic in the temporary image of the virus in NORMAL.DOT. The code responsible for infection is in the Document_Open procedure which is renamed to AutoExec in the temporary global state of the virus.

The virus mechanism is based on a two-phase model. Phase one is responsible for the virus activation, phase two for the polymorphic transformation of the virus code and the actual infection.

The first phase activates whenever an infected document is opened. This action invokes the Document_Open procedure of the virus. It checks whether it is the first or second phase of the virus. As the second phase is established in a separate *Word* instance, this check is performed simply by counting the number of *Microsoft Word* applications mentioned in the task list.

Then the virus searches for a file named C:\TEMP.DAT which is generated by the second phase and removed after each target document gets infected. If the virus finds this file, it concludes that an infection process is already in progress and aborts the virus activation. This check is meant to avoid double activation – when the second phase of the virus opens a document which is already infected, the Document_Open macro of the target file is executed, but on detecting that it has got into the middle of an infection process, it aborts.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.



After that IIS.I exports its code (except for the first two lines) line by line into a text file named C:\FUCK.TXT. The first line of the exported file is replaced with 'attribute vb_name = "Flitnic"', the second line will read 'Sub AutoExec' instead of the original 'Sub Document_Open'.

After that the virus imports this text file into the VB project of the global template. As a result the new module will be called Flitnic, and it will contain the AutoExec procedure.

Then IIS.I launches a second instance of *Word* using the Shell command in a minimized window. The AutoExec procedure recently created in the global template is executed and the second phase of the virus is activated.

Finally the virus clears all of its signs. It begins by removing all the code lines from the class module of the active document, and then removes the Flitnic module from the global template. The second phase of the virus is activated by the AutoExec macro whenever the second instance of *Word* is executed.

Counting the number of *Word* application instances (at this point both the original, phase one *Word* instance and the new instance, launched by the former, are active), the virus detects that it must be the second phase, so it skips to the appropriate code.

First it hides the *Word* instance, then creates a polymorphic image of the virus. Then it creates the infection flag file (C:\TEMP.DAT), and infects every DOC file on drive C: and its subfolders. When all the documents have been infected, it deletes the flag file and finishes execution.

This two-phase model has one major drawback. The Shell command does not provide synchronization between the phases – this could lead to problems sometimes leaving the Flitnic module in the global template. Watch out for it, I hope this helps!

Gabor Szappanos
Computer and Automation Research Institute
Hungary

VIRUS ANALYSIS 1

Gala Première

Nick FitzGerald

Another virus 'first' recently hit the headlines (see last month's News p.3). CSC/CSV.A is a prepending virus written in *Corel SCRIPT*, the scripting language of recent versions of the popular *Corel* graphics programs *CorelDRAW* , *PHOTO-PAINT*, and *VENTURA* (v.8.0).

In CSC/CSV.A – as is so often the case with proof of concept viruses – there is little of significance in the code itself. The risk posed by this virus is low, both due to bugs in its code and the seemingly low incidence of script file exchange amongst users of the applications that support *Corel SCRIPT*. The virus is reported by others to work in version 7.0 through 9.0 of both *Corel* applications, but this analysis was limited to testing CSV under version 8.0 of *CorelDRAW* and *PHOTO-PAINT*.

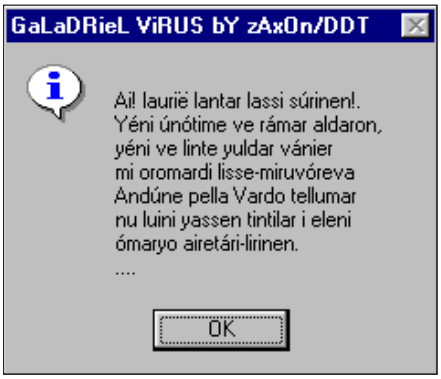
The Corel SCRIPT Platform

As VBA is to *Office*, so *Corel SCRIPT* is the programming language that provides macro automation capabilities to some of *Corel's* popular products. *Corel SCRIPT* is derived from BASIC and is different, though not greatly so, from the VBA language used in *Microsoft's Office* applications. The language obviously contains extensions to provide interfaces to *Corel* product-specific functionality.

Scripts are stored in plain text files with the conventional extension CSC and interpreted at runtime. Perhaps unusually, the CSC extension is not associated with the *Corel SCRIPT Editor* – the development environment for writing and debugging *Corel SCRIPT* files.

Script Wizardry?

It is fair to assume that CSV's writer is a Tolkien fan. Apart from some of the labels in the code being names from 'Lord of the Rings', the writer also wished the virus to be called GaLaDRiel. Further, if a script infected with the virus is run on 6 June a relatively harmless payload can trigger, displaying an extract from 'Galadriel's Song of Elamar' in a standard *Windows* dialog box. The song, another 'Lord of the Rings' reference, is in the language of High Elves.



VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

The trigger condition test assumes the date is presented in d/mm format, rather than the more common dd/mm or mm/dd forms. This means the payload can only trigger on machines set to display dates this way. Default regional settings for Australia, Belgium, New Zealand and Spain meet this requirement. Coupled with the extensive use of Spanish in the code, it is likely CSV was written in Spain.

Fifteen of the 62 lines of virus code are involved in testing for and delivering the payload. The rest of it implements a straightforward prepending infection mechanism.

Infection

Following the payload trigger test, a find-first/find-next loop is initialized looking for files matching '*.csc' in the current script directory. These files are checked to see if their first lines start with the words 'REM ViRUS'. The first file found that meets this criterion is assumed infected. The active virus uses this file as the source from which to copy its code during the ensuing infection. An uninfected host is also sought.

With an infected file and uninfected host file chosen, the virus goes about replicating. First, it renames the host to mallorn.tmp then opens the source file for reading and creates a new file with the host's name. One line of the source file is read into a string buffer variable then written out to the new host file. This continues line by line until a line starting 'REM END' is read and written (usually the last line of the virus code). The copy of the original host, mallorn.tmp, is then opened and its contents read line by line and written to the new host file. If the virus is running from an infected script, that script's code is now run.

Finally, the open files are closed and mallorn.tmp deleted. The latter is desirable from the virus writer's point of view, as most files in the various script folders in *CorelDRAW* and *PHOTO-PAINT* are displayed in the script selection lists, so deleting temporary files after use reduces the 'obviousness' of the virus' action. Watching the virus run, various delays, screen refreshes and disk accesses are noticeable. However, that is generally true of running most of the scripts supplied with *Corel* applications, so alone does not really contribute a 'giveaway' factor.

General Comments

It should not be surprising that a virus *can* be written for a modestly powerful scripting language that can be used to automate most tasks in powerful image manipulation software. This does not, of course, address *why* anyone should bother writing such a thing, but I will leave that question for others to ponder.

As with the solitary *Ami Pro* macro virus Green_Stripe (see *VB*, March 1996, p.11) and *Access* macro viruses (see *VB*, April 1998, p.15), *Corel SCRIPT* viruses seem unlikely to become a large threat. The small userbase (relative to *Microsoft's Office* products) diminishes the risk. Also, the fact that CSC scripts (macros) are stored in files independent of the drawing files of the host product – the files which are most likely to be exchanged – further conspires against rampant growth and distribution of this form of virus.

With CSV, there are also the bugs. Several researchers have reported that most (or all) of the tested CSC scripts supplied with the *Corel* host products fail to run once infected, giving rise to warnings like the one shown. This is caused by a *Corel SCRIPT* 'feature' I could not find mentioned in the documentation.

String variables are variously described in *Corel SCRIPT's* on-line help as capable of storing '32,765 characters' and '0 to 4 billion characters'. Despite these claims, it appears that the 'Line Input' operator cannot handle strings longer than 79 characters without corrupting them. Thus, the insertion of random characters into script code lines over 79 characters long is a 'feature' of CSV as it depends upon the Line Input operator to read its and its host's source code while creating the newly-infected instance of the host file.

Finally, a small logic error in the stopping conditions for the find-first/find-next loop can also give away the CSV virus' presence. After it has infected all viable hosts in a directory, subsequent invocations of the virus can result in error messages similar to this one being displayed.

In short, an altogether ordinary proof of concept virus. Greater originality may seem likely from a Tolkien fan, but apparently that is too much to expect from this one!

CSC/CSV.A

Aliases:	GaLaDRiel, CS/Gala.
Type:	Direct action, prepending *Corel SCRIPT* infector.
Self-detection:	CSC files whose first line starts 'REM ViRUS' are presumed infected.
Trigger:	6 June.
Payload:	Displays an Elvish poem from 'Lord of the Rings' in a message box (see text).
Removal:	Using a text editor or the *Corel SCRIPT Editor*, delete all viral code from the beginning of an infected script file to the end of the line beginning 'REM END OF ViRUS'.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

VIRUS ANALYSIS 2

ACG in the Hole

Adrian Marinescu
GeCAD srl, Romania

The MtE mutation engine was something quite new in virus programming, and led to important changes. Since then, polymorphism has been one of the ways virus writers have chosen to protect their creations from scanning engines. The development of code emulators and good crypto-analytic algorithms resulted in anti-virus products needing slight changes and/or updates in order to detect most of the new polymorphic viruses. Furthermore, there were a few cases of polymorphic viruses that could not be detected at all for a long time; Zhengxi (see *VB*, April 1996, p.8) and Uruguay (December 1992, p.12) are good examples.

All polymorphic engines were based on the same idea: maintain the virus body in an encrypted form, using a variable key/algorithm, and generate a polymorphic code that decrypts the rest of the body and executes it. Some of the first viruses *not* based on this idea were the members of the Ply family. Ply is not encrypted, but there are no parts constant enough to extract a reliable signature.

Using a slightly modified idea, the TMC family managed to become in the wild. TMC had many small constant parts, linked with jumps. That made algorithmic detection easy to write for this virus, but the door was now open. These kinds of virus were the first ones that could not be exactly identified, raising big problems regarding their disinfection.

Then the ZCME family used the same idea, mixing the code in a 16 KB buffer. The only weakness was that algorithmic routines still worked, because there were a lot of constant small pieces that could be used for detection. Last year, a new kind of virus came up. Called Lexotran, it was able to generate different looking forms, with the same result. The idea was to keep the mixing engine in encrypted form – the mixing engine itself processed the virus body during infection before creating new and highly variable shapes of itself. The drawback was that the mixing engine was linearly encrypted with 8-bit keys. That could be used to write a detection algorithm to search for the encrypted part in the virus body.

The author of the ACG family understood this disadvantage and developed a new idea – what if the encrypted body is not stored in one piece, but in more scrambled pieces spread through the entire virus image? The ACG family is not a dangerous one, but the polymorphic engine is well written and very stable. The main problem with it is that its engine could easily be used in other viruses, far more dangerous ones. Also, the idea can be successfully applied to *Windows* viruses, potentially making this kind of virus a big problem in the future.

ACG – A Storm in a CPU?

In mid-April I received an archive containing several different samples. All of them had the capability to mutate in a special way, being one of the most advanced polymorphic viruses written. However, another virus known as the Yowler family had the same engine (one of them looked quite similar to a version that I received). Using a string found inside an infected file, I tried to discover the author of the virus and found the name Lucky. I took a look at several other viruses written by him.

These viruses, dated this year, betrayed a mid-range assembler programmer, incapable (in my opinion) of coding such complex viruses. After a lot of research I found out that the nature of the problem is more serious. Back in 1997, a virus writer known as Mad Daemon released a package called ACG (Amazing Code Generator). The idea is not new, several creation tools were released before, but the implementation is outstanding.

The package contained an ACG compiler, some examples (two of them in the package I received) and some documentation. ACG is a high level language adapted for polymorphic viruses. Up to now, there have been several viruses that use the ACG engine. Some of them are direct action viruses, searching for all COM files in the current directory, while the others are memory resident. All of them are simple viruses, with no payload; they seem to be written only to demonstrate the features of the engine.

Non-resident Versions of ACG

There are several of these direct action infectors in existence. When an infected file is run, the virus receives control via a jump instruction which points to the virus body. Then it resizes the used memory to 64 KB and searches for the mask '*.COM'. If a file is found, it is opened in read-write mode. If the open fails the virus assumes that the file is already infected (the infection marker is the read-only flag in the file's attributes field). After that, it reads three bytes of the original file, which will be used to restore the control to the host program.

Next, it checks the file size – only files between four and 48,640 bytes are infected. ACG's engine needs another block of 64 KB, so the virus calls DOS' memory alloc function. In order to generate a new form, the virus needs to call its engine. It joins the pieces of the engine's image together, using a marker for each block which is followed by the next chunk's address. There is also a check which tells the virus that the engine can be called – if the end of the processed image is not 0x0F. The virus writes the message 'Internal compiler error!' (for the A variant) 'LUCKY B.R.D 1994-99' (for the B variant) and returns control to the original file.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

Next, it calls the mixing engine. The engine generates a virus mutation at a fixed address before giving control back to the main virus body. This writes the generated mutation into the infected file, frees the allocated memory and checks for more suitable files. Infected files grow by a random amount of between 0x3000 and 0x4000.

Resident Versions of ACG

These versions are simple COM/EXE infectors, using ACG's polymorphic engine. One of them is more advanced, (in fact, it is a debug version of a more advanced virus) avoiding several known anti-virus utilities, while the others are simple infectors. Since they use ACG's features, they are polymorphic both in files and memory.

When an infected file is executed, the virus receives control, checks if it is already resident and calls the polymorphic engine. This call is made only once, so all infected files will have the same shape until a new memory installation is made. Moreover, there is a chance to generate a light version of the virus, without the compiler inside.

The Yowler family contained such cases within it (this is the same virus, despite being labelled differently by anti-virus products). This particular feature makes these versions slow polymorphics and obtaining a good sample set is harder. Infected files have the maximum memory filed in the EXE header set to 0xffff.

ACG's Polymorphic Engine

The mutation engine receives control from various points, depending on the caller's version. It is stored in packed form (7,781 bytes) using a slightly modified LZ compression scheme, similar to the variant used in old executable packers. In its unpacked form it is 11,368 bytes long.

When an ACG mutation is built, the compiler takes good care of its instructions. It generates the respective (mutated) code for them, and also generates something like a byte-code definition for the next mutation which contains informations used by the next mutation. Polymorphism is achieved using ACG language when the virus is created. ACG language contains a powerful set of instructions, which can be used to specify the nature of an operation.

Also, for bigger blocks of constant data (like the compiler itself), ACG splits them into several chunks located at random offset in the entire virus body. Moreover, these chunks are encrypted using random keys.

This reduces the possibility of detecting this virus using its constant parts (or cryptoanalysis on these parts). The polymorphic engine is not as advanced as Lexotran's, but it can generate a large range of opcodes, including 32-bit types. The main idea is that you can use several processor instructions to obtain the same result. Each original instruction or block can be generated at any position, linked by JMP or CALL statements. All the offsets involved in any instruction are relocated. Despite its complexity, on a large test-set of about 10,000 generations, the polymorphic engine generated only valid shapes. This makes ACG one of the most complex but stable polymorphic viruses ever.

Detection and Cleaning

Detection of ACG is a serious issue. There is no reliable way to add exact identification to an anti-virus product. There are two options – using dedicated algorithms or using code emulation (CE). I think a scheme based on CE is better than algorithms, for two reasons. The false alarm percent is lower and CE can also be used at disinfection.

One idea would be to make use of the fact that the virus replicates and use a detection scheme based on Int calls and all the logical parts that are constant for these viruses. Both Lexotran and ACG can be detected by hooking the emulator's Int calls and checking the parameters (there are some constants such as memory allocation, search for files, etc). This is a simple method, but it requires the CE to handle all the opcodes generated by the viruses.

Also, we could trick the CE to make the virus restore the original file's state, in order to find disinfection information (e.g. if there are no files in the current directory, ACG simply restores the host and gives control. We could make the CE report no files in the current directory, emulate until we reach a certain instruction – for COM files the jump to the initial entrypoint or intersegmental jump for EXEs). This seems more like generic disinfection, but it can be backed up by a lot of checks.

Epilogue

Tools like ACG could easily be used in more dangerous creations. Its features could be used by any virus creator with no outstanding qualifications in virus coding. As classic polymorphism opened a new era in computer virus history, this development could mean a new direction in virus writing. This is why such exotic viruses, despite not being in the wild, should be considered a serious matter.

ACG family

Aliases:	Yowler, Kopie.
Type:	COM/EXE infectors.
Infection:	Direct action and resident, hooking the 'execute function' 0x4b.
Self-recognition in EXE files:	Maxmemory in EXE header set to 0xffff.
Self-recognition in COM files:	All infected files are read-only.
Self-recognition in memory:	Calling Int 21h with the edx register set to Acg! clears the carry flag.
Removal:	Boot from clean floppy, delete infected files and restore from backups.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

TUTORIAL

The Generic Repair of File Viruses

Peter Morley
Network Associates Inc

While I was working on Data Recovery back in 1992, I read an article by Frans Veldman, who already saw the need for the generic repair of file viruses and gave some clear pointers as to how to set about it. I read it three times (rare for me) and convinced myself that it would never work.

Frans had pointed out the fundamental truth; the easiest way to do generic repair is to make use of the virus code itself. I was wrong. It does work. The subject kept coming back like a song and a year or so later, when I moved into the Virus Lab, it became clear to me that we already had some of the necessary tools to start implementing it. So I did, treating it as a background task.

Generic repair of file viruses has a big advantage for anti-virus vendors. When they get a new virus, they can do a little extra work and then stand a good chance of having to do none at all on subsequent variants. The second or third variants may show the need for a slight change, but that is normal. This saves a lot of scarce resource. Unfortunately, it also leads to a permanent underestimation of the virus count because we do not (contrary to popular opinion) count the ones we have never looked at.

In addition to this, it also has a big advantage for customers, who stand a good chance of being able to detect and repair viruses the vendor has never seen, thus saving the costs of sending samples and awaiting responses. But do not write in and ask what the payload is!

In my article 'the Biggie' on processing the 15,000 new viruses which came in the VKit collection (see *VB*, November 1998, p.10), I noted a category which required no work. It consisted of those specimens which we had already detected and repaired satisfactorily. The deliberate omission in that article was the revelation of how many of the 15,000 required no work. That question has been asked by a few people and the answer is 'just over 12,000'. That was the point at which I stopped being cagey, and took the position that 'Generic Repair has saved us several months of hard work and we should use the resource saved to put further generics in place'. That same article also deliberately failed to make the point that if, in the future, further 'Biggies' arrived, we could not cope without generic repair tools in our anti-virus armoury.

This is about how to do it. If it contains anything helpful to our competitors, so be it. They are also moving in this direction and have already thought hard about the fundamentals. The days when every variant of every virus could be processed separately are now behind us, even if we wanted to do it that way.

Generic Repair Techniques for COM files

In the case of simple appending viruses the infection is implemented by changing a few bytes at the beginning of the file, to pass control to the virus code which is added at the end. After executing the virus code, the virus will often repair the changed bytes itself and execute the original file. Repair is straightforward.

a) Put a buffer at the beginning of the file.

b) Execute the virus code (in a controlled way, so it can do no damage) up to the point where it has replaced the original bytes in the buffer.

c) Write them to the file.

d) Chop the virus code off the end.

This technique is by far the most useful and normally works with kit viruses such as MPC, VCL and IVP. It also works with many other viruses. There are two cases where it fails. The first is when the virus does not behave in the simple way illustrated above, but still works. The second is where the bugs in the virus code prevent it getting there.

A big advantage of using this method is when the virus is encrypted. The virus has to decrypt itself to function, and it will decrypt itself for us, with no extra effort on our part. This method, combined with the simple EXE file method outlined below, is the one we had already used when the VKit collection arrived. Instead of handling 15,000 viruses, we had to attend to less than 3,000 and we were able to process them in groups rather than one at a time.

For other appending viruses there is another simple technique, which does not depend on executing the virus code. This involves scanning the virus code and stopping at a point which is the same distance from the repair data, for all the variants. We can then:

a) Skip the constant distance to the repair bytes, and read them.

b) Jump to the beginning of the file, and write them.

c) Chop the virus code off the end.

Obviously, if the virus is encrypted, *we* must decrypt it before we find and use the repair data. So this method is slightly less efficient, if just as effective, as the one above.

For prependers, a simple technique which usually works is to scan for the small piece of code which defines the length of the virus. Then take the length bytes and use them to

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

chop the right length off the beginning of the file. If the virus is encrypted, we may have to decrypt it first. This technique is used for several of the Pixel viruses and it seems to work just fine.

'Topsies' are infections where a large number of bytes from the start of the file are moved to the end and the gap is then filled by the virus code. The technique in this case is similar to that for prependers, but when you get the length bytes, use them to take the right number from the end of the file and replace those at the start. Then chop the same number off the end.

This is the technique we used on the 50 variants of NGV (Nuke Gp), which now all have the same detection and repair. Sadly, there are another 26 variants which are overwriting viruses and cannot be repaired. We resolved this small complication by reporting the topsies as NGV.GR and the overwriters as NGV.ow. They had the same detection, so how did we distinguish between them? Simple. The topsies must contain code which jumps to the end of the file. The overwriters must not. So if the code is not present, the variant is an overwriter.

There is another prepender and topsy technique. Where the length bytes are not available, it may be possible to scan for specific bytes which are a constant distance from the end of the virus code. Then use the data to define the virus length, and proceed as above.

Generic Repair Technique for EXE files

For EXE files, we have to find and replace the original four parameters in the EXE header: SS (Stack Segment), SP (Stack Pointer), IP (Instruction Pointer), and CS (Control Segment). We may also have to replace some 'fixups'. With any luck, the virus will find the four parameters for us, and place them in the appropriate registers. All we have to do is read those registers, replace the four pairs of bytes in the EXE header and then chop the virus off the end. This technique was already in place when VKit arrived.

For viruses where this particular approach does not work, the technique of scanning the virus code and stopping at a point the same distance from the repair data for all variants is also available. It works just as well for EXE files as it does for COM files.

Some viruses rebuild the EXE header themselves or keep it. When the time comes to execute the original file they have a complete original header to refer to. We can use it too. Just read it, jump to the start of the file and write it, using techniques similar to those for repairing a COM file.

Other Effects on the Anti-virus Industry

A simplified overview of the code we write to handle a virus can be broken into three sections – detection, the use of checksums to distinguish between variants, and repair, which may be different for each variant.

The amount of code in section three is dramatically reduced if generic repair is used. The second section is also reduced (and can be left out altogether, if there is now no need to differentiate). This gives a speed improvement when scanning collections in reviews, as well as less code.

Section one can usually be left as it is, although there may be pressure to strengthen it or to make it more generic. This provides a partial answer to Ian Whalley's *cri de cœur*, in his guest editorial in the April issue of *Virus Bulletin*. I quote 'creeping bloat… can only end with the products being too prone to error, and too slow to use'. It also explains comments I have heard like 'Scan and Repair files are not growing as fast as they were'. True. I am delighted that someone noticed!

Do We Need It At All?

The whole topic of repair can be brushed aside as totally unnecessary. 'File repair is unreliable. Correct procedure is to replace infected files with the originals'. It is also glossed over with the use of the word 'disinfection' by some vendors. To me, disinfection now means, 'we will delete all but a few of your infected files, and leave you to sort out the mess'. At present, this may be at its worst when the 'disinfection' code is from automated immune systems. However, it will not be. The *NAI* automated immune system will use generic repair from the start. So, do we need repair at all? If we do not, then I have wasted a lot of time. But I think we do, for two overriding reasons.

Firstly, it provides user convenience and productivity. A user can easily run it and continue working instead of calling his IT department and going into limbo for several hours. Secondly, it can be done on a server, without interruption of service and without using the manpower needed to replace hundreds of infected files.

This has been based on repair of File viruses. The philosophy, however, has spilt over and we have applied the guideline 'Make detection and repair as generic as possible' to macro viruses too. An early example was Wazzu. Our driver to handle that was made generic at a very early stage. As a result, this virus has rarely given us problems, even though there are now over 200 variants.

Occasional odd variants have required special treatment, but that is all part of the game! Repair of files affected by macro viruses can rarely be achieved by replacement of an original, and the normal method is deletion of the offending macro. Fortunately, this is not a serious problem, when one is attempting to be generic.

I must make one last provocative comment. If we *do* need repair, I think reviewers should give it a lot more attention than they do now, so that users are better informed. This comment applies not only to magazine reviews, whose reviewers may be overstretched to provide it, but also to professional reviewers, such as VTC Hamburg and the University of Tampere.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

FEATURE 1

Letter from Syria

Ahmad Yarob Kashoor
CompuKashoor, Syria

Syria is a country whose roots go deeply into the history of the human race. The first human settlement, which was established around 8500 BC, was in Syria. Aleppo, this country's second largest city, is the oldest conurbation in the world, having been continuously inhabited since 1780 BC. It was in Ugarit in the fifteenth century BC that the first alphabet was invented. This article traces computer viruses back to their beginnings in Syria, not that long ago!

The Early Bird

Computer viruses first appeared in Syria at the end of 1989. I was working as an AutoCAD instructor at a well-known local training centre. My knowledge about computer viruses at that time was limited to the articles published in computer magazines such as *BYTE* and *PC Magazine*.

When one of my students noticed that the volume of most of his 5.25-inch diskettes had changed to show "(C) Brain", I knew that our centre had been hit by the Brain virus. Further investigation proved that his colleagues had been infected by the same virus. At this early stage, we treated the infection simply by copying all the files that were on the infected diskette and re-formatting it. We later removed infections by using up to date anti-virus software. The Brain virus soon found its way to other sites with a remarkably large number of infections in Syrian University colleges and institutes.

The Haze Grows

Knowledge about computer viruses was very sparse, even among computer professionals, and people began to mix the facts with myths by comparing the new phenomenon with biological viruses. One manager in a governmental administration asked all of his computer staff to wear white gloves and to throw infected diskettes into incinerators!

The establishment of the *Syrian Computer Society* (*SCS*) by Basel Al-Assad (the President's son) in 1989 played an important role in encouraging the public and private sectors to automate their work procedures. This was reflected in the increased use of PCs (especially *IBM*-compatible ones).

The growth in international contacts, together with a growing number of graduates coming back from foreign universities had increased the virus problem. Stoned, Italian and Cascade were the dominant viruses between 1990 and 1992. Israeli viruses such as Jerusalem, Suriv and Frodo had crossed the border into Syria through illegally copied software obtained from Lebanese computer shops.

1992 – 1995

Michelangelo was the first computer virus to be mentioned in the Syrian media. At 11pm on 5 March 1992, Syrian television warned the country that a computer virus would strike the next day, and asked people not to use their computers that day. The number of infections was unknown due to the lack of official statistics, but I had the chance to add a sample of it to my virus zoo.

One_Half was the most active and damaging virus during this period. The fact that most anti-virus products available at that time had removed it without decrypting the encrypted clusters had caused serious data loss at many sites. In some cases we were forced to re-infect the system with the same virus in order to gain access to data on the hard disk, and to make backups before disinfecting it again. This virus still keeps its place among the most common viruses in Syria. A recent field investigation (1999), conducted by myself, revealed its presence on more than 30% of the PCs at different colleges in Aleppo's University.

The KVS or Kiev.1942 virus deserves a mention. Russia was the source of most infections caused by this virus, since it was transferred through the diskettes of a large number of Syrian students who had graduated from Russian Universities. The fact that most anti-virus software (including *Dr. Solomon's Antiviral Toolkit* and *McAfee VirusScan*) were unable to detect it for more than four months gave Russian *DialogueScience's DrWeb* a good reputation in the Syrian computer community, and it is still considered one of the best. KVS maintains its regional presence.

The first of January 1995 brought headaches to a number of companies in the north of Syria, since a file-infecting virus called Big-Bang had triggered its destructive payload on that day and overwritten their hard disks. AntiEXE, AntiCMOS, Jumper.B, Ripper, DieHard, Unashamed, Parity_Boot, and Junkie were also getting familiar. Some of them are still infecting as I write this.

The first macro virus to reach Syria was, of course, the Concept virus. Johnny, Cap, Npad, Class, DZT, Wazzu and finally Laroux followed it. In the meantime, Cap has the lion's share of all macro virus infections – in fact 85%.

Syrian Viruses

The first deliberate attempt to write a virus code in Syria was performed in 1994 at the computer labs of the Information Engineering College, University of Aleppo. A benign file-infecting virus was written to propagate among COM files as a graduation project to train students on how viruses work and how to write a disinfection program to disinfect them. This caused little damage except some problems when allocating memory and accessing the floppy drive.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

The same attempt was repeated in 1997, with two different types of macro virus in mind. The first one was a *Word* macro virus named HAMAH in reference to the students' home city. It spreads by infecting ANORMAL.DOT, which is the global template in the Arabic version of *Word*. It causes abnormal effects to the opened documents, and its destructive payload deletes all saved documents.

The second one was written as a proof of concept only. It was an *Excel* macro virus, which replicated through the XLSTART folder without any serious damage. All these viruses were kept within the computer lab boundaries and (to my knowledge) have never found their way to the wild.

The problem facing Syrian IT professionals is that writing a local anti-virus program is not profitable due to the lack of local viruses. The most used anti-virus packages continue to be *Dr. Solomon's Antiviral Toolkit, McAfee VirusScan* and *DialogueScience's DrWeb*.

The Catastrophe

Syrian Information Technology professionals will remember 26 April 1999 for a long time. Thousands of PCs were turned into metal boxes at the start of that working day. The incidents covered most of the big commercial corporations, schools, Universities, research centres and government administrations. At Sham '99, the biggest computer exhibition in Syria, which took place from 25 to 29 April 1999, many exhibitors were forced to close their stands on the second day due to CIH.

This large-scale infection and the tremendous ensuing damage was mainly attributed to the flourishing industry of illegal copying of computer software. The unintentional distribution of CIH on the cover CD of *ArabChip Magazine* (March 1999 issue) and *Middle-East Windows Users Magazine* (February 1999 issue) exacerbated the problem.

As a coordinator of the *Syrian Computer Society's Information Security Interest Group* (*ISIG*), I began to receive reports warning that most of the software copying shops were infected by the virus two weeks before the unlucky day. A detailed description of the virus, its effects and how to disinfect it was distributed, while a special TV program was prepared, but all of these efforts failed to dam the infection flood which was to come.

On the heels of this unprecedented large-scale damage, people became very sceptical and gullible about any rumours regarding new viruses. Therefore, a hoax that a dangerous virus was to strike on 1 June 1999 had spread like wildfire all around the country, and phone calls to our group were nonstop the day before, but nothing happened.

The local press had also contributed to the scare badly, since the official government newspaper *Techreen* had published on an article entitled 'Be careful of the AIDS virus sent via The Internet'. The editor claimed that a new computer virus AIDS II (which is in fact, a very old one and relates to the year 1989) was spreading worldwide through Internet email, and that it could destroy all kinds of computer hardware including memory chips, disk drives, cards and even speakers.

I am sure that we are now forced to spend more time fighting myths and hoaxes, rather than chasing real viruses and removing them. On the other hand, the CIH disaster was instrumental in illustrating some real advantages. Managers finally realized that controlled backup procedures must be their first line of defence and the implementation of authenticated anti-virus software together with strict anti-virus policies their second.

The Situation

The risk of infections arriving via the Internet is high in spite of the small number of official Internet users – less than 1,000 in Syria. There are more than 5,000 Syrian Internet users who subscribe illegally with Internet providers in the neighbouring countries, mainly in the Lebanon.

The fact that they pay expensive rates for their international phone calls prevents them from using on-line anti-virus scanners for checking downloaded materials, thus making their systems vulnerable to malicious code. Win32/Ska ('Happy99') entered the mailboxes of hundreds of them.

A detailed survey performed by *ISIG* in May 1999 resulted in the following list of the ten most common viruses in Syria, starting with the most prevalent: CIH, One_Half, Cap, Stealth_Boot.C, AntiEXE, Junkie, KVS, NYB, Laroux and Unashamed.

The Future

The virus situation in Syria may prove very difficult to keep under control for the following reasons. Most importantly, the majority of computer users (more than 80%) do not use original software and depend on pirated material which is frequently infected with viruses.

Along the same lines, only 5% of users buy authentic anti-virus software and update it on a regular basis. There is a false sense of security resulting from the use of anti-virus software found on the cover CDs of computer magazines, which are out of date all the time. This I can only describe as expired medicine.

Syrian law does not offer any protection to intellectual property software (copyright), and there are no laws against copying and selling pirated programs. Neither does it offer any protection to the victims of viruses – there are no laws against writing or distributing viruses.

Finally, opening the Internet door wide to the public increases the virus threat. Despite all this, Syrians will enter the third millennium with open minds on the implications and ramifications of Information Technology, encouraged by the continuous support of Dr Bachar Al-Assad, Head of the *Syrian Computer Society*.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

FEATURE 2

Virus Writers – Part 3

Sarah Gordon
IBM Research

So far in this series we have covered five of the most frequently asked questions concerning virus writers. In this, the third and final part of the series, we will examine the question that seems to raise the most heated debate of all: why do they do it?

Justifications

As a starting point, let us approach the topic from the viewpoint of the virus writers. In their own words, how do they *justify* their actions? Notably, the arguments outlined below have remained relatively unchanged over the last few years. Such arguments were frequently encountered upon some of the FidoNet virus echoes and BBS in the early days. Here, borrowed (and paraphrased) from the satirical commentary 'Why Computer Viruses are Not and Never Were a Problem' [33], is an examination of the most prevalent justifications that appeared several years ago:

'We are doing research. This is just our research. You can't tell us not to do research – We have the right to do it. We have the right to write viruses, too, and to make them available – It's about freedom. We have the *right* to do this research and the *freedom* to make viruses available – You want to keep this "top secret" virus knowledge to yourselves, but we will set it free. We will educate the people. Information wants to be free – We are not really hurting anyone, we don't force anyone to download our viruses, and we don't force anyone to use them. That is up to the individual – You AV guys are all bad, in it for the money, you need us – You just don't understand!'.

Of course, one need not go back through the archives to see examples of these arguments. You need only read *current* Usenet news posts, to see some of the same old arguments made today, by new people who believe with all their hearts that they know something the rest of us do not. Here are some more contemporary examples [34]. Note that I am unable to credit the authors due to the fact that I could not authenticate them:

'The only justification my code needs is furthering education, and knowledge. These are the greatest strengths the human race has… – my goal is accomplished… my reaserch (sic) and making available this information to those who are interested… '.

As you can see, not a lot has changed. At this point, however, I would like to make an aside, and interject a few comments. I hear these arguments all the time, for the virus writers reading this, take note, please.

Programming computer viruses is not some super-élite, arcane art-form. It does not require some top-secret type of programming skills known only to the cognoscenti. Virtually anyone with the interest can learn how to do it, and just doing it does not make it 'research'. Good research implies certain goals, guidelines and appropriate scientific technique [35]; this is worlds apart from randomly injecting a small piece of self-replicating code into an unsuspecting, unconsenting and uncontrolled computer-using population. It is just plain irresponsible to experiment with viruses in such uncontrolled environments, given the potential for viral interaction with the computers of human subjects. This includes experimentation using your college or employer's network without their consent. No matter how you look at it, it is *irresponsible.*

It is also irresponsible to set self-replicating programs free where this interaction is an inevitable consequence of that action. Some would argue that placing viruses on the WWW is in fact setting them 'free', that placing viruses on the Internet for other people to experiment with is irresponsible because the program's author cannot control what someone else does with the viruses once they are made available [36, 37]. This is a question with philosophical and cultural colourings beyond the scope of this article, but please think about it!

However, there is certainly a potentially expensive, destructive cycle that follows the life of an ItW computer virus. There are issues of negligence and liability when it comes to making these types of programs available, with concern being shown by more and more organizations as evidenced in these examples of Membership Agreements and Disclaimers [38–41]. This is one reason why many Internet Service Providers now have 'acceptable use policies' which prohibit the distribution of computer viruses [42–44]. They do not want to risk being involved in lawsuits related to negligence. To any virus writer reading this article: if you *must* experiment, keep your experiment to yourself, or you might find yourself in the middle of just such a legal action.

Now that is cleared up, let us continue with some of the current justifications: 'If my code was used to damage someone's computer, that is the responsibility of the person who's (sic) immature behaviour has resulted in damage. Open your mind, and expand your horizons… its (sic) a huge world out there, if you can just get over your fears – … this is nauseating… you feel you have the right to censor, and condemn the creativity of young, brilliant minds. you fear what you dont (sic) understand… '.

There are other justifications expressed from time to time. One is, was, and has been, and probably will be, 'if it was not for us, you guys wouldn't have a job'.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

This is not quite true. There is a relationship, certainly, but a relationship does not imply the existence of a positive justification. Consider the following statement: 'If it were not for people who shoplift, the store detectives would be out of a job.' Yet, shoplifters are not heroes and we do not consider new and novel ways of absconding with merchandise to be admirable or acceptable. Or: 'If people didn't throw trash on the ground at Disneyworld, the street cleaners would be out of a job'. Yet, we surely do not look upon those that throw lit cigarettes on the ground with admiration, do we?

The bottom line here is that while it is true that if virus writers did not release viruses, the users would not need anti-virus software (allowing most virus researchers to shift into other, equally interesting, areas of work), attempting to paint some lovely picture of healthy symbiosis is simply not supported by the facts.

Motivating Factors

So far, we have taken a brief overview of the justifications many virus writers use, and noted that most, if not all, of these reasons are far from new. Similarly, several of the motivational factors for virus writing are also unchanged.

Today, as yesterday, in some cases virus writers are motivated by simple intellectual curiosity. This is understandable, especially considering the free availability of viruses and the media attention given to viruses and virus writers. Virus writing continues to take place as a form of political expression; Stoned_June4th (Beijing) and Macedonia being two examples.

Viruses as an expression of love and admiration for the opposite sex or for peers continue to be keyed into existence. In the early days we saw viruses like Gergana, Neuroquila and the MtE 'demo virus'. More recently, we find 'love' expressed a bit more directly via viruses like Ivana, which proudly proclaims:

```
  'Na kraju, samo jos da kazem: volim te,
Ivana [by utik]
     'And finally, I would like to say: I love
you, Ivana [by utik] [45]
```

Another motivation behind virus creation is to designate 'turf'. In the past, viruses like NPOX and Vice planted the viral flag for NuKE; today, we have such ignoble creations as WM97/Antimarc [46]. In other cases, virus writing and distribution is positively correlated with being told 'thou shalt not'. It is widely agreed upon by behavioural scientists that the 'thou shalt not' approach may not prove very effective in situations where direct and immediate consequences cannot be observed [47]. Thus, despite much saber rattling on the part of the anti-virus industry and lawmakers, legislating away the virus creation problem seems an unlikely solution.

Being 'one of the boys' appears to continue in importance, too; the need for peer approval is illustrated by gravitation toward 'groups', with group affiliation providing a form of social identity [48]. While there have been several cases of female virus writers documented over the past several years [49–53], females do not appear to have made a significant contribution to the population of those viruses in the wild. Currently, while females play a minor role in the virus-writing community as a whole, their presence appears to be a moderating influence in the community. There appears to be very little gender-related sexual bias within the community. Further research into gender issues related to group involvement and technology, and virus writing specifically, might provide some additional insights.

Finally, some virus writing has been the direct result of various forms of provocation by anti-virus researchers themselves. This was more common during the early days of the virus problem, when a (thankfully) small number of anti-virus researchers would insult the virus writers, calling them names [54–59] or claim they were too stupid to create a virus that 'did xyz'. Shortly thereafter, we would find a virus doing or attempting to do 'xyz'.

Disparaging remarks have been made regarding the young person's appearance, or physical characteristics. While there is simply no point to this sort of 'discussion' [60], this cycle does unfortunately repeat itself from time to time today [61, 62], though with lower frequency. Such negative interactions continue to produce negative responses as well as negative impacts on users and should be avoided. Young people learn through transitive interaction, not debasement.

Recently, there has been a trend toward adapting the 'open source' mindset for publication of viral code, and this has not been lost on the virus-writing community. This is apparently done in an effort to warn users of the dangers of certain design philosophies. To quote one virus writer, 'Some good virus writers like my friend VicodinES (who retired) are here to demonstrate the vulnerability of badly written softwares, like all Microsoft offerings. They don't like destruction.'

It should be noted that it is not virus writers alone who think there may be some merit to the publication of viral source code. Some users report that on-line publication facilitates a wide understanding of exactly what viruses and payloads do; some believe such publication is actually essential in keeping corporate security people up to date.

However, not everyone shares this view. In particular, many in the anti-virus community believe such public dissemination of information is irresponsible. David Chess of *IBM's Thomas J. Watson Research Center* has this to say about the issue. 'The moderators should never let such things through. Unlike bug exploits, where at least a case may be made that it's a valid last resort if a vendor has been notified of a bug and ignored it, viruses don't go away when you just fix a bug' (63).

It should be further noted that the differences between open source and availability for software in general, for security exploits and for computer viruses are substantial. Beyond

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

the scope of this article, the effect on the user when these worldviews collide will be discussed in Vancouver at VB'99 in 'When Worlds Collide' [64].

For now, it should suffice to say that if virus writers are attempting to influence *Microsoft* or any other corporation by showing real or alleged vulnerabilities in the product line, it would seem a more responsible course of action would be to do so *without* the replicative mechanism. Certainly, it must be done in a private way that does not endanger other computer users' rights to safe computing.

Some virus writers are more honest with themselves. Here is an example of the reasoning given to me recently by one active virus writer [65], who will remain anonymous.

'i fully agree with you about it being irresponsible, i don't know why i release them on a web page. viruses have always fascinated me since i got infected myself the first time (it was parity boot b), since that i'm (lets call it) addicted to studying them by collecting and writing them myself. i don't feel good if i have nothing to do with viruses, no matter if VX or AV wise (AV, i'm doing alot of "anonymous" antivirus support for people on alt.comp.virus and i have been active on #nohack for some time helping people to get rid of their mIRC_worm infections..

so it doesn't make a big difference for me, i just like the VX people more then AV's – AV's like nick fitzgerald who believe that anybody who doesn't share their opinion has no right to exist). when someone says that he is writing viruses just for "educational purposes" it is a lie in my opinion (i think i have said that in some interview a long time ago also, and it was a lie).

i have often thought about why i'm really doing this (i could probably spend my time with more productive things) though till now i never really found out why. the best thing i came up with is that it is a "hobby"... you don't really know why you go play tennis, why you watch football matches, why you collect stamps... you just like doing it, and if you can't do it for some time you feel bad i guess you can't understand this.. do you smoke? i don't, and never did, so i really can't imagine why its so hard for people to stop smoking... i believe them anyway because i know that if i'd start smoking i'd also understand how/why they think so.'

I do not smoke, but I do understand. Becoming fascinated with viruses is not an alien concept to me. Like many other anti-virus experts (and virus writers), I became interested because I suffered the impact of a virus.

However, after understanding and appreciating the impact viruses can have on human beings in terms of their work and personal lives, many of which center around computers, it never occurred to me that creating and releasing more and more viruses was an acceptable way to behave. Too much depends upon the stability of computers for this sort of silly experimentation and potentially dangerous game.

Yet, for some virus writers, our societal dependence on computers is *exactly* the motivation for virus creation and distribution, and it is not a game to them. According to some virus writers, our society entrusts far too much important information to computerized technologies, to the point where there is a moral responsibility to take a form of action which forces us to reconsider this dependence. To them, the end justifies the means, and while this implementation of the civil disobedience *is* new, the concept is ages old.

The Songs Remain the Same

Why, you might ask, are we seeing the same old arguments, over and over? This is mostly due to the replacement factor, a direct result of the 'ageing out' phenomenon I described in part one of this series. This factor has a tremendous effect on the overall virus writing subculture.

By and large, the members of the virus writing community are in a constant state of flux. As mature adults exit from one side of the population, new, ethically normal but undeveloped adolescents enter at the other.

In turn, this continual flux provides a certain lack of development within the community. Hence, each new batch of virus writers is essentially discovering these arguments for themselves, leading to oft-repeated debates between the 'white hats' and the 'black hats'. Finally, those members who remain in the community are all somewhat ethically underdeveloped, further skewing the population, and making the role models there decidedly less than perfect.

It does not appear to be the case that virus writers are becoming more malicious *per se*. There may be more malicious viruses circulating nowadays, but this is probably attributable not so much to the fact that *people* are more malicious as to the fact that the *number* of people (some of whom by sheer chance are more malicious than the norm) having access to Internet technologies has increased dramatically. People are not getting worse. There is just more opportunity for those bad apples that have already rotted and fallen off the tree.

The Way Things Are

Despite the similarities, there are some differences in virus writing which are unfortunately becoming more and more common. These were first noted in 'The Generic Virus Writer II', presented at the *Virus Bulletin Conference* in 1996, where I introduced the concept of the 'New Age Virus Writer'. This concept became a prime-time news headline with the introduction of the Melissa virus into hundreds of thousands of networked computers.

Today, more and more virus writers have an increased awareness of connectivity issues that simply was not present in the early days. It should not be surprising that an increase in networked environments would lead to an increase in opportunities for people to learn about net-

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

works. The sorts of innovations that have come to exist in the past five years certainly add a new dimension to the problem of viruses.

Payloads now have the capacity for compromising an entire network, and more than a few virus writers are beginning to explore more general security issues. Melissa was not a one-shot-deal; Explorezip (see p.3 of this issue) indicates very much the shape of things to come.

This trend will probably continue over the next several years, and it is likely that there will be an increased cross-over between the security and virus worlds. In light of this, response time to new viruses will become paramount, as the presence of viruses on the corporate LAN may well become more than the current nuisance it is now and a matter of considerable urgency.

What has happened to the original groups that held all of these beliefs and had these motivations? In some cases, individuals have simply repositioned themselves, taking a 'leadership' role. In most cases, however, the members of old groups have grown up, realized that creating and releasing computer viruses is not a good or admirable thing, and moved on.

Some former virus writers have taken jobs in various computer-related industries; some have found other professional fields more rewarding. In a few years, most of the current crop (at least, the ethically normal ones, which we hope would be most of them) will probably 'age out' of this behaviour. However, unfortunately, there are always new ones to take their places.

Those that continue writing and making viruses available to the general public will be seen as 'irresponsible' at best, and criminal at worst (depending on one's geographic location and what one does with the viruses once they are written). That said, it is interesting to note that while some have argued for stronger legal action, research into adolescent at-risk behaviour finds that youths are *not* significantly motivated by fear of legal reprisal or involvement with the criminal justice system. They are more likely to be influenced by peers, family and significant others whom they like and respect.

The Last Word

Fear of the law does not appear to be a major demotivator for many virus writers and it appears that for now, the community continues to play itself out over and over again. Until we begin to tackle the root causes of virus writer motivation, this will continue to be the case; a multi-disciplinary approach is required to solve a multi-faceted problem. Anything less is oversimplification.

33. Gordon, S. 1994. Why Viruses are Not and Never Were a Problem. *From the Proceedings of the 1994 EICAR Conference*. St Albans, U.K.

34. Public communication. 1999. Alt.Comp.Virus.

35. Lawrence Berkeley National Laboratory. (1999). ELSI in Science. The need to KNOW vs. the need to GROW. http://www.lbl.gov/Education/ELSI/research-main.html.

36. Gordon, S. 1993. Virus Exchange BBS: A Legal Crime? Legal, Ethical and Technical Aspects of Computer and Network Use and Abuse. *American Association for the Advancement of Science*. Irvine, California.

37. VIRUS-L Digest. 1994. Volume 7, Issue 83.

38. Owens, M. 1999. Killing Two Birds With One Stone. *Good To Know*. http://www.trade-attorney.com/goodtoknow.html.

39. Membership Agreement. http://www.sexyclips.com/agreement.html.

40. Disclaimer. http://www.snapit.com/PEG_disclaimer.html.

41. Disclaimer. http://www.roweandmaw.co.uk/5const.htm.

42. Acceptable Use Policy. http://www.verio.com/policies/aup.shtml.

43. Acceptable Use Policy. http://www.telemanage.ca/disclaimer.html.

44. Acceptable Use Policy. http://www.vas.net/policies.html.

45. Ivana description from: http://www.Europe.DataFellows.com/v-descs/ivana.htm.

46. W97M/Antimarc description from: http://www.Europe.DataFellows.com/v-descs/antimarc.html/.

47. Klein, H. 1975. Behaviourally oriented treatment for juvenile offenders. *Corrective & Social Psychiatry & Journal of Behavior Technology, Methods & Therapy*. Vol 2, pp. 17-21.

48. Kipke, M. & Unger, J. 1997. *Adolescence*. Street Youth, Their Peer Group Affiliation and Differences According to Residential Status, Subsistence Patters and Use of Services. Vol. 32, Issue 127.

49. Anonymous. 1995. Private email and in-person communications. Used with permission.

50. Anonymous. 1996. Private in-person communication. Used with permission.

51. Anonymous. 1997. Private in-person communication. Used with permission.

52. Anonymous. 1999. Private IRC communication. Used with permission.

53. Anonymous. 1999. Private email communication. Used with permission.

54. VIRUS-L Digest. 1991. Volume 4, Issue 126.

55. VIRUS-L Digest. 1992. Volume 5, Issue 174.

56. Skulason, F. 1992. Virus Trends. *From the Proceedings of the Virus Bulletin Conference*. Edinburgh, Scotland.

57. VIRUS-L Digest. 1994. Volume 7, Issue 56.

58. VIRUS-L Digest. 1994. Volume 7, Issue 4.

59. Bennehaum, David. 1998. Heart of Darkness. *WIRED*.

60. Solomon, A. 1994. The Computer Virus Underground. *From the Proceedings of the International Virus Bulletin Conference*. Jersey.

61. Sterling, B. 1997. Sterling Versus Virus Writers. http://www.av.ibm.com/.

62. Dufner, E. *Dallas Morning News*. 1999. Virus Writers make a science of mischief.

63. Chess, D. 1999. Private email communication. Used with permission.

64. Gordon, S. & Ford, R. 1999. When Worlds Collide. Preprint.

65. Anonymous. 1999. Private email communication. Used with permission.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

COMPARATIVE REVIEW

Comparing Wares

It was initially intended to use *NetWare 5* as the operating system for this review, but with only a minority of the products offering features specifically for the latest *Novell* platform, and for the sake of speed (the tests run comparatively slowly on the machine used as the test server) *NetWare 4.10* was in fact used for the bulk of testing.

Perhaps the introduction of *NetWare 5* is responsible for the relatively small number of products that were submitted for testing. Quite a few products are currently receiving some sort of facelift, and so only ten developers sent us their wares, one less than in the previous *NetWare* Comparative Review (see *VB*, July 1998 p.11).

Test Procedures

In common with all recent comparatives, various aspects of each scanner's properties were investigated. Detection rates for on-demand scanning have been determined using a test-set consisting of standard, macro and polymorphic viruses. In addition to this, each product has been tested against a virus set aligned to the April 1999 WildList. Given the submission deadline of 30 April (for product shipping) this In the Wild test-set gives us a realistic impression of how well each product copes with the viruses that are known to be prevalent in the real world.

New additions to the WildList since the last comparative include the *Microsoft Office*-infecting O97M/Triplicate.C, W97M/Pri.B and the infamous W97M/Melissa.A. As for all Comparative Reviews, additions were also made to the other test-sets. Making their debuts in the *VB* test-sets this month are {Win32,W97M}/Beast (analysed in last month's *VB*, June 1999 p.6) and the polymorphic file infector Win32/ACG (see p.8 of this issue). For a complete listing of the test-sets used for testing, see the URL listed at the end of this review.

The performance of the on-access (real-time) scanner is fundamental to the usefulness of any anti-virus product. Beside the obvious importance of detection rates, the overhead such a scanner imposes upon the server must also be considered. Irrespective of how good its detection rate is, a scanner that log-jams the server, reducing its performance, is undesirable. Thus the overhead of each of the on-access scanners has been tested, by monitoring the time taken to copy a set of 200 files (100 COM/EXE and 100 OLE2) between directories on the server. By normalizing the results to the average baseline (with no on-access scanner loaded) of 28 seconds, the results presented within this review are expressed in units of time, as well as in terms of percentage overhead.

Perhaps of less importance to the day-to-day running of anti-virus software, the scanning speed of each of the on-demand scanners has also been investigated. For this, two file sets have been used. The first is a 5500 file COM/EXE collection (520 MB), and the second a 373 file OLE2 (DO? and XL?) collection (65.3 MB). These sets are virus-free, and so also provide a false positive test for all the products.

A slight change has been made to the format of the main results tables this month. The detection rates have been calculated as usual, and are expressed in the usual percentage format. However, instead of listing the number of detected samples, the tables now list the number of *missed* samples. The detection rates are also listed beneath each of the product headings (for on-demand scanning unless indicated otherwise). Since detection rates are normalized with respect to the number of samples of each virus, products that miss the same number of samples do not necessarily achieve the same percentage detection rate.

CA InnoculateIT v4.5 (13/04/99)

ItW File	100.0%	Macro	99.7%
ItW File (o/a)	100.0%	Macro (o/a)	99.7%
Standard	99.9%	Polymorphic	96.8%

It appears that the old *InnocuLAN* name badge from the *Cheyenne* days has gone, and as with the rest of the *Computer Associates* range, the product is now adorned with the suffix '*IT*'.

In addition to the server-based virus-scanning component, *InnoculateIT* provides the user with the option to install a centralized management component. Using this, the administrator has full control of deployment, configuration and scanning right across the network.

Anti-virus protection is initiated by simply running the NCF file that is created during installation. The server console is designed with real-time anti-virus protection as its main focus. On-demand scans can be configured and initiated however, and multiple tasks can be created and placed in a job queue.

The first product this time around to detect all the in-the-wild viruses during both on-demand and on-access scanning, *InnoculateIT* maintains the high standard it set in the previous *Windows 98* comparative. Detection elsewhere in the test-sets was equally commendable.

In terms of scanning speed, *InnoculateIT* performs just above average when compared to other products featured in this review. Scan rates of just over 430 KB/sec and 1290 KB/sec were obtained for scanning of the executable and OLE2 files on the hard disk respectively.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

On-demand tests	ItW File		Macro		Polymorphic		Standard	
	Missed	%	Missed	%	Missed	%	Missed	%
CA InnoculateIT	0	100.0%	7	99.7%	175	96.8%	1	99.9%
Command AntiVirus	4	99.7%	33	99.2%	173	96.8%	0	100.0%
CA Vet NetWare	20	98.1%	139	95.8%	423	95.8%	3	99.6%
DialogueScience DrWeb	3	99.7%	34	98.8%	107	98.0%	35	95.8%
Kaspersky Lab AVP	3	99.7%	25	99.1%	50	99.2%	3	99.6%
NAI NetShield	16	99.0%	31	99.1%	428	95.8%	0	100.0%
Norman FireBreak	13	98.8%	47	98.5%	174	96.8%	1	99.9%
Proland Protector Plus	77	89.4%	1,626	42.5%	11,095	22.3%	852	32.4%
Sophos Anti-Virus	4	99.7%	43	98.6%	174	96.8%	12	99.5%
Symantec Norton AntiVirus	0	100.0%	13	99.4%	175	96.8%	0	100.0%

The overhead of the on-access scanner is similar to that for the other products, reaching approximately 40% when scanning both incoming and outgoing files.

Command AntiVirus v4.54 SP2 (24/04/99)

ItW File	99.7%	Macro	99.2%
ItW File (o/a)	99.7%	Macro (o/a)	99.0%
Standard	100.0%	Polymorphic	96.8%

An *InstallShield* installation routine is used to copy the necessary server and workstation files to their desired destinations. An option to install *AlertTrack*, an NLM to manage alert notification across a network, is also provided.

Admirably high detection rates across all the test-sets were observed, although complete In the Wild detection was hampered by O97M/Tristate.C-infected *PowerPoint* and *Excel* files. Despite changing the configuration to scan All Files, the *PowerPoint* files were not scanned and so remained undetected.

Configuration of the *F-PROT* virus engine can be achieved either from the server console using the wealth of command line switches that are available, or using an administration utilty at the workstation. Scanning speeds were certainly not quick, even when the tests were repeated with the limits upon the CPU usage removed. Perhaps more relevant to the day to day use of the product though is the overhead the on-access scanner of *Command AntiVirus* imposes. Tests showed only a small overhead (just under 50%) when scanning both incoming and outgoing files.

CA Vet NetWare v9.9.4

ItW File	98.1%	Macro	95.8%
ItW File (o/a)	98.1%	Macro (o/a)	95.6%
Standard	99.6%	Polymorphic	95.8%

Yes, the title is correct. This is the second product from *Computer Associates* this month, thanks to their recent acquisition of the *Vet AntiVirus* products.

Installation of *Vet NetWare* has to be performed from the workstation, and is achieved by running the setup program on the supplied diskettes. Subsequently, configuration and initiation of scans is initiated from either the server console or via RCONSOLE from the workstation. *Vet NetWare* employs configuration sets for saving and loading multiple configurations, which allows up to 16 set-ups to be stored.

One slight annoyance with *Vet NetWare* is that there is no indication of scan progress on the server console, merely a message box stating that a scan is in progress. The detection rates observed were slightly lower than those expected from recent performances by the Australian product, but it should be noted that this is predominantly due to the omission of quite a few file types from the default file extension list.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

On-access tests	ItW File		Macro		Polymorphic		Standard	
	Missed	%	Missed	%	Missed	%	Missed	%
CA InnoculateIT	0	100.0%	7	99.7%	176	96.8%	1	99.9%
Command AntiVirus	4	99.7%	45	99.0%	173	96.8%	0	100.0%
CA Vet NetWare	20	98.1%	142	95.6%	423	95.8%	2	99.7%
Kaspersky Lab AVP	17	98.9%	28	99.1%	296	98.2%	12	99.4%
NAI NetShield	16	99.0%	31	99.1%	428	95.8%	0	100.0%
Norman FireBreak	13	98.8%	47	98.5%	174	96.8%	1	99.9%
Sophos Anti-Virus	4	99.7%	43	98.6%	174	96.8%	12	99.5%
Symantec Norton AntiVirus	0	100.0%	13	99.4%	175	96.8%	0	100.0%

As expected, repeating the tests whilst scanning in All Files mode improved the detection rates markedly, although W97M/Pri.B-infected documents were still missed from the In the Wild test-set.

To avoid overloading of the server, *Vet NetWare* employs a fast scan for scheduled and on-access scanning, looking for viruses in files according to the methods used to infect such files. When a scan is started on-demand however, a more thorough full scan method is used, where each byte of every file is scanned. The difference between the two scan methods was only in evidence once, with a sample of Cantando.857, which, interestingly, was missed during the full scan yet detected during a fast scan.

DialogueScience DrWeb v4.06β (30/04/99)

ItW File	99.7%	Macro	98.8%
ItW File (o/a)	n/t	Macro (o/a)	n/t
Standard	95.8%	Polymorphic	98.0%

Installation of *DrWeb for NetWare* (*DWNW*) was a straightforward affair – simply copying the relevant files to the server manually, and then loading the relevant module. In keeping with other *DialogueScience* anti-virus products the interface of *DWNW* is simple and efficient to use if somewhat dated. In its default settings, *DWNW* scans files by extension and content. Thus, if the extension or the content of a file shows it to be either executable or pertaining to *Microsoft Office*, it is examined by the virus engine.

Detection-wise, *DWNW* performed well across all the test-sets. As with other products in this review the virus engine appears to be unfamiliar with the format of *PowerPoint* files, missing O97M/Triplicate.C infected samples. On the positive side, *DWNW* was one of only three products to detect samples of Win32/ACG – a newcomer to the Polymorphic test-set. This was thanks to *DrWeb's* heuristics (enabled by default), which reported 67 out of the 174 samples to be infected with a COM virus. The downside of such keen heuristics was in evidence during the speed tests however, where 19 clean files were flagged as suspicious.

It was not possible to test the performance of the real-time scanner because upon its activation access to all files on the volume (infected or clean) was denied. Discussion with the *DialogueScience* developers suggested that this was a problem assoiated with the LIBUPI patch applied to the *NetWare 4.10* installation. However, reinstalling *DrWeb* onto the server with various combinations of older patches applied did not solve the problem.

Kaspersky Lab AVP v3.0.121 (30/04/99)

ItW File	99.7%	Macro	99.1%
ItW File (o/a)	98.9%	Macro (o/a)	99.1%
Standard	99.6%	Polymorphic	99.2%

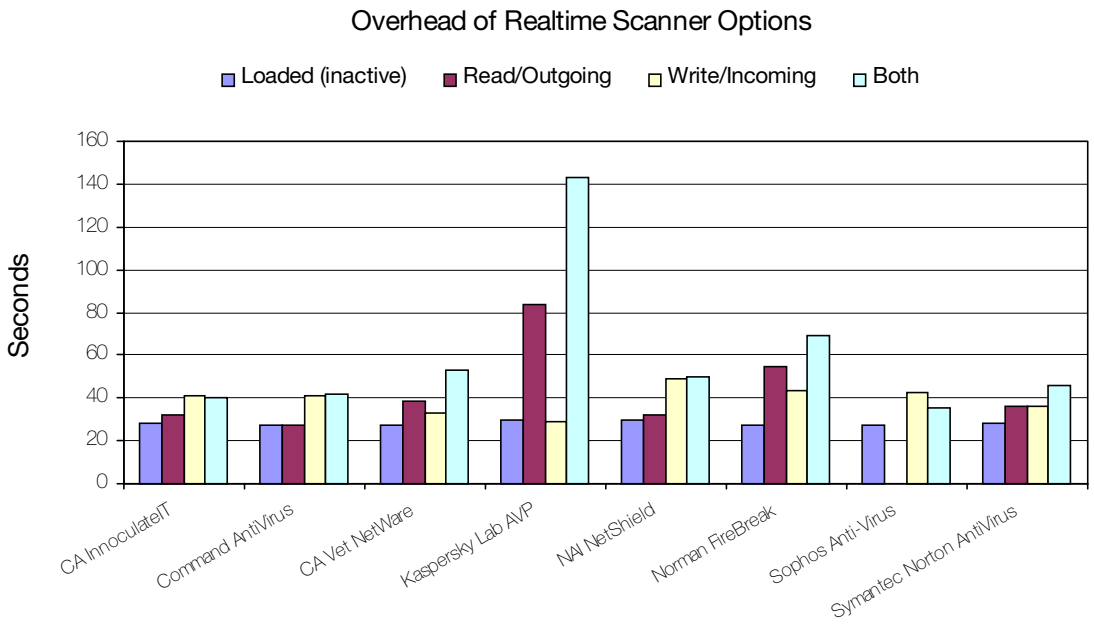
AVP has always had a tradition of high detection rates across the test-sets, and is in the enviable position of having detected 100% of the ItW viruses thrown at it during the last seven comparatives. Such a performance was not to be repeated this time around however, thanks once again to the O97M/Tristate.C-infected *PowerPoint* samples.

Detection rates elsewhere in the test-sets though high, were not as high as have come to be expected of *AVP*. Changing the configuration such that packed files were unpacked

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

during scanning improved the results slighlty - although the bulk of the misses were registered against the infected *PowerPoint* files.

In the last comparative the weakest area of *AVP's* detection was against polymorphic viruses. This seems to have been remedied, and *AVP* correctly detected all but 50 of the ACG samples.



Scanning speed has never been a strong point of *AVP*, and little has changed in this respect – with this product the emphasis has always been upon accurate detection at the expense of speed.

NAI NetShield v4.0.2 SP1 (26/04/99)

ItW File	99.0%	Macro	99.1%
ItW File (o/a)	99.0%	Macro (o/a)	99.1%
Standard	100.0%	Polymorphic	95.8%

Installation and administration of *NetShield for NetWare* can be performed either directly from the server console, or more easily from the workstation using the *NetShield Console*. Loading of the console is password-protected, avoiding unwanted changes to the scanner's configuration. The password protection is perhaps somewhat over-eager, since immediately after installation access is prevented. Fortunately the default password is 'NetShield' which was guessed by the reviewer after a few attempts.

This is the first appearance of the *NetWare* product in *VB* tests since the swallowing of *Dr Solomon's* by *Network Associates*. The awkward interface that was reported previously has certainly been remedied during this take-over. Out of all the *Windows*-based administration consoles featured in this review, the *NetShield Console* proved to be the most straightforward and efficient to use.

An extremely limited file extension list proved once again to be *NetShield's* downfall. With viruses currently in the wild capable of infecting SCR and a range of *Office* files, such extensions simply have to be included in the default list. Rescanning in All Files mode showed the expected improvements, although *PowerPoint* samples infected with O97M/Tristate.C were still missed from the ItW test-set. In addition to the ACG samples, a handful of Marburg samples in the Polymorphic test-set were also missed.

Norman FireBreak v3.97 (30/04/99)

ItW File	98.8%	Macro	98.5%
ItW File (o/a)	98.8%	Macro (o/a)	98.5%
Standard	99.9%	Polymorphic	96.8%

Another product of uncertain identity, *FireBreak* (or is it *Virus Control*?) from *Norman Data Defense Systems* once again put in a strong performance across the *VB* test-sets.

Detection of the in the wild viruses was not up to the usual *Norman* standard, with two viruses slipping through the net. Firstly, as with most of the other products in this review, O97M/Tristate.C-infected *PowerPoint* samples were missed, despite the fact that PPT and POT files are included in the default extension list. In addition to this, misses were registered against Raadioga.1000 samples, a virus that has successfully been detected by *Norman* products in previous *VB* comparatives. Consultation with the product developers identified the problem and it has since been fixed.

Administration of *FireBreak* is a simple, no-frills affair, performed entirely from the server console. Centralized surveillance and reporting can be enabled in a multiple server environment, by designating one server to be a communications hub.

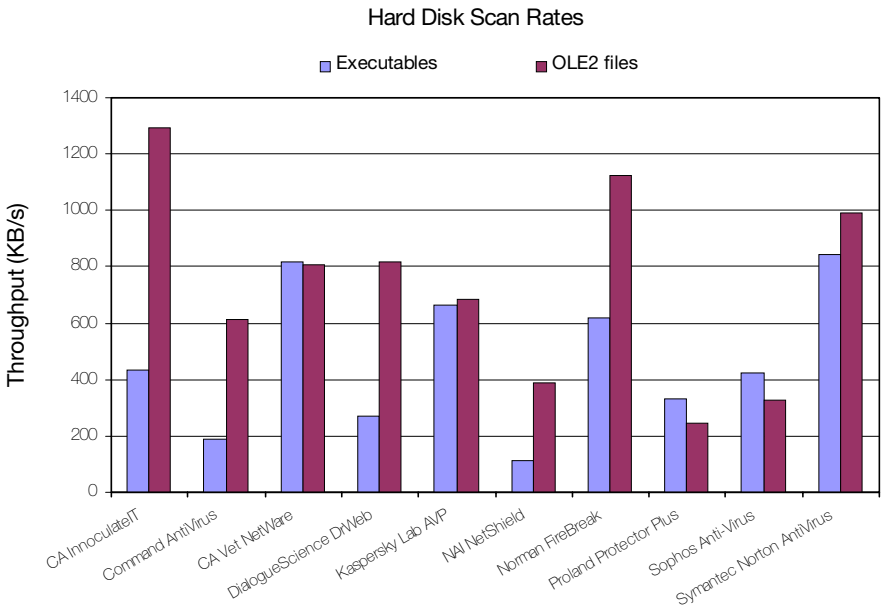
Proland Protector Plus v6.6.A.01

ItW File	89.4%	Macro	42.5%
ItW File (o/a)	n/t	Macro (o/a)	n/t
Standard	32.4%	Polymorphic	22.3%

A regular entrant to *VB* Comparatives Reviews on other platforms, this is the first appearance of the *NetWare* version of *Protector Plus* from the Indian anti-virus company *Proland Software*.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

Upon finding infected files, *PPN* attempts to cure them by default. Suprisingly, this option can neither be changed nor disabled – something that needs to be addressed. A further hinderance in testing the product is connected with the log files that are produced. A separate log is produced for each directory scanned, and deposited within that directory. Beside the fact that trawling through deep directory structures for log files is undesirable, a centralized log containing all infection reports would be far more sensible.



On the positive side however, achieving a detection rate of 89.4% against the ItW viruses is indicative of good progress by *Proland*, and their highest rating thus far in *VB* Comparative Reviews.

Sophos Anti-Virus v3.21

ItW File	99.7%	Macro	98.6%
ItW File (o/a)	99.7%	Macro (o/a)	98.6%
Standard	99.5%	Polymorphic	96.8%

As mentioned in the last *NetWare* comparative, installation of *Sophos Anti-Virus* (*SAV*) is achieved by copying a single NLM to the server manually, and then loading it.

For the first time since March 1998, *Sophos Anti-Virus* failed to detect all the ItW viruses. *PowerPoint* and extensionless *Excel* samples infected with (yes, you guessed it!) O97M/Triplicate.C were missed during both on-demand and on-access scanning. Repeating the tests scanning All Files with *SAV*'s default 'Quick' scan (compared to the more thorough 'Full' scan) resulted in detection of the extensionless *Excel* sample, but not the elusive *PowerPoint* samples.

In order to make the results more comparable with those obtained using other products, the results quoted for on-access detection are those obtained using the real-time monitor that is provided with *SAV*. The familiar *Intercheck* component was not tested. One notable point is that the scanning speeds reported in this review are those for the first scan, in which *InterCheck* checksums are created. Subsequent scans were performed at almost twice the rate (for the executable set).

Symantec NAV v4.04

ItW File	100.0%	Macro	99.4%
ItW File (o/a)	100.0%	Macro (o/a)	99.4%
Standard	100.0%	Polymorphic	96.8%

Having dropped its guard against Win95/Fono during the *Windows 98* comparative, *Norton AntiVirus* (*NAV*) brings up the rear in style this month – the only product managing to detect the full complement of in-the-wild viruses thrown at it.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

	On-Access Scanner Overhead				Hard Disk Scanning Speed					
					Executables			OLE2 files		
	Loaded Inactive	Read or Outgoing	Write or Incoming	Read and Write	Time (min:sec)	Throughput (kB/s)	FPs [susp]	Time (min:sec)	Throughput (kB/s)	FPs [susp]
CA InnoculateIT	1.4%	15.5%	46.6%	42.5%	21:02	433.4	0	0:53	1292.5	0
Command AntiVirus	-1.9%	-2.6%	46.8%	49.4%	48:37	187.5	0	1:52	611.6	0
CA Vet NetWare	-2.4%	39.1%	17.1%	90.2%	11:10	816.3	1	1:25	805.9	0
DialogueScience DrWeb	n/t	n/t	n/t	n/t	33:30	272.1	[19]	1:24	815.5	[1]
Kaspersky Lab AVP	4.9%	200.0%	2.9%	410.0%	13:44	663.8	0	1:40	685.0	0
NAI NetShield	7.0%	14.9%	76.1%	78.7%	81:00	112.5	0	2:56	389.2	0
Norman FireBreak	-2.2%	95.0%	54.6%	146.9%	14:45	618.0	0	1:01	1123.0	0
Proland Protector Plus	n/a	n/a	n/a	n/a	27:19	333.7	5	4:38	246.4	1
Sophos Anti-Virus	-2.6%	n/a	52.9%	25.5%	21:32	423.3	0	3:30	326.2	0
Symantec Norton AntiVirus	-0.8%	28.6%	30.3%	63.8%	10:47	845.3	0	1:09	992.8	0

In common with the bulk of the products reviewed, installation of *NAV* is performed from the workstation. Configuration of the scanner is achieved using the *Windows*-based configuration utility on the workstation. Once configured, initiation of an on-demand scan can be achieved from either the workstation or the server console. Passwords can be used to prevent anyone other than the Administrator loading the main *NAVNLM* module, altering the program configurations or disabling real-time protection and scheduled scans.

Summary and Conclusions

It is encouraging to see most of the products achieving high detection rates across the bulk of the test-sets. However, the fact that only two out of ten products managed to detect 100% of the In the Wild test-set samples may perhaps alarm some readers.

By altering product configurations, detection of some of the missed samples across the Macro and ItW test-sets was achieved. Are the misses definitive therefore? Well, to be honest, yes. With reference to the missed O97M/Tristate.C-infected *PowerPoint* samples, the issue of file types is somewhat immaterial (the only additional product that would have detected 100% of the samples if the tests were run in 'All Files' mode was *DrWeb*).

Even if this were not the case, the simple fact is that the products should be continually developed to protect users from viruses they are most at threat from – i.e. those viruses that are in-the-wild. Some of the products – 'designed to provide optimum protection' – actually advise users not to adjust the default configurations.

With regards to the missed O97M/Tristate.C samples, all the products detected the infected *Word* and *Excel* files successfully. So, despite infected *PowerPoint* files remaining undetected, the products *will* detect the continual reinfection of the *Word* and *Excel* environments. The ensuing telephone calls to Technical Support would then no doubt resolve the problem. This is not sufficient protection, however. Examination of how the products performed against other non-ItW *PowerPoint* viruses shows the problem to be due to the inability of the majority of products to deal successfully with files of *PowerPoint* format. At the time of testing, only two of the products featured here (the *iRiS* engine-powered *InnoculateIT* and *Symantec's Norton Anti-Virus*) managed to detect such samples successfully.

So, is it disconcerting that eight out of ten products did not detect all the ItW viruses? Perhaps so, but not that surprising. However, from the discussions *VB* has had with the product developers it would appear that *current* versions of those products have learned how to scan *PowerPoint* file formats. With the submission deadline for the next comparative looming, whether or not this is true will soon become apparent.

Technical Details

Test Environment: Server – *Compaq Prolinea 590*, 90MHz Pentium with 80 MB of RAM, 1 GB hard disk, running *NetWare 4.10*, with 410PT8B, LIBUPI, DS410N and STRTL7 applied. Workstation – 166 MHz Pentium with 4 GB HD, CD-ROM and 3.5-inch floppy, running *Windows 98* with *Novell's Client 32*.

Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NetWare/199907/test_sets.html.

A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

ADVISORY BOARD:

Pavel Baudis, Alwil Software, Czech Republic
Ray Glath, RG Software Inc, USA
Sarah Gordon, WildList Organization International, USA
Shimon Gruper, Aladdin Knowledge Systems Ltd, Israel
Dmitry Gryaznov, Network Associates, USA
Dr Jan Hruska, Sophos Plc, UK
Eugene Kaspersky, Kaspersky Lab, Russia
Jimmy Kuo, Network Associates, USA
Charles Renert, Symantec Corporation, USA
Roger Riordan, Computer Associates, Australia
Roger Thompson, ICSA, USA
Fridrik Skulason, FRISK Software International, Iceland
Joseph Wells, Wells Research, USA
Dr Steve White, IBM Research, USA

SUBSCRIPTION RATES

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, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YP, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

Virus Bulletin, 18 Commerce Way, Woburn, MA 01801, USA

Tel (781) 9377768, Fax (781) 9320251

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.

This publication has been registered with the Copyright Clearance Centre Ltd. Consent is given for copying of articles for personal or internal use, or for personal use of specific clients. The consent is given on the condition that the copier pays through the Centre the per-copy fee stated on each page.

END NOTES AND NEWS

A Practical Anti-Virus Workshop will be run by *Sophos* on 15 and 16 September 1999 at the organization's training suite in Abingdon, UK. For more details or to reserve your place, contact Daniel Trotman; Tel +44 1235 559933, fax +44 1235 559935, or visit the company web site at http://www.sophos.com/.

CompSec'99, the 16th World Conference on Computer Security, Audit and Control will take place from 3–5 November 1999 at the QE2 Centre, Westminster, London, UK. A Directors' Briefing will be held on 4 November. Conference topics include malicious software, firewalls, network security and Year 2000 contingency planning. For more details contact Tracy Stokes at *Elsevier*; Tel +44 1865 843297, fax +44 1865 843958, or email t.stokes@elsevier.co.uk.

In Brussels, Belgium, from 4–7 March 2000, **the ninth annual *EICAR* conference**, also known as the first European Anti-Malware Conference, takes place. For more information, to place a booking or to order a timetable visit the web site at http://www.eicar.dk/.

WebSec'99 takes place from 9–13 August 1999 in San Francisco, California. For more information contact Adam Lennon at the MIS Training Institute in the US; Tel +1 508 8797999 ext 336 or email alennon@misti.com.

The *Computer Security Institute's* 26th annual conference and exhibition is to be held from 15–17 November 1999 at the Marriott Wardman Park Hotel in Washington DC. For more information on the 85 featured presentations or pre- and post-conference seminars, contact *CSI*: Tel +1 415 9052626 or visit http://www.gocsi.com/.

Virus Bulletin 9th Annual Conference

30 September–1 October
The Hotel Vancouver
Canada
Call Jo Peck
+44 1235 555139
email jo@virusbtn.com
http://www.virusbtn.com/

In late May *Symantec* announced its Digital Immune System (DIS) strategy which amalgamates *Symantec's* anti-virus technology, *IBM's* automated virus analysis and *Intel's* management technology. It will be released over the next eighteen months with the first phase to include a total managed solution built on top of *Norton AntiVirus Corporate Edition* (due out this summer). *Symantec's* DIS will include tools and utilities for systems and policy management, virus protection, server performance, desktop configuration, diagnostics, systems stability, remote system operation, management of remote users and disaster recover – all from a single console. For more details contact Charlotte West at *Harvard PR*; tel +44 181 7590005, or email charlotte@harvard.co.uk.

Following *Norman Data Defense Systems Inc's* takeover of *ESaSS* in March 1998, **the company began advising customers that *Norman Virus Control* (*NVC*) and *ESaSS's ThunderBYTE* were merging to become one product**. After a year's interim, during which customers were encouraged to switch to *NVC*, this transition has come to a close, with *ThunderBYTE* products no longer commercially available, and no support offered. Carl Bretteville of *Norman* explained how there was potential for confusion following the acquisition solely of *ESaSS* and not of all *ThunderBYTE* distributors, 'This has gone on for a year because we didn't want to leave anyone hanging. We wanted the transition to go as smoothly as possible. Anyone still under contract to ThunderBYTE will, of course, be transferred to an NVC equivalent.' More information about this can be obtained from the company web site http://www.norman.no/.

***Network Associates Inc* announces a new system known as *AVERT* Risk Assessment**. *NAI's* Anti-Virus Emergency Response Team has begun to rank the relative danger of new, in-the-wild viruses. The final categorisation of viruses depends on three criteria: prevalence, danger of payload and commonality of infection vehicle. Under the new system CIH is classed as a high-risk virus, along with the recently reported Melissa. According to Sal Viveros, Group Marketing Manager for *NAI's Total Virus Defense* division, most in-the-wild viruses fall into the medium-risk category, but viruses can be reclassified should their histories or activities change. For further details about the new classification system, visit the company web site at http://www.nai.com/avertlabs/.

VIRUS BULLETIN **©1999 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /99/$0.00+2.50**
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.