ISSN 0956-9979 JUNE 1999

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Oh dear, oh dear, oh dear *Virus Bulletin*:** At long last the Letters page makes a comeback in this issue. Love it or hate it, it's your way to have your say, starting on p.4.

• **Define your terms:** Find out all you need to know about polymorphism. Two researchers from *Kaspersky Lab* set the record straight on p.14.

• **Don't panic!** Our tutorial this month paves the way for future corporate case studies. In clear and easy-to-follow steps, the actions to take in the event of a virus or malware outbreak on your system are documented on p.16.

CONTENTS

COMMENT

Flashback – When the Chips were Down

Black Monday, 26 April, was doomsday for thousands of computers in many countries. Corporations stopped working due to destroyed PCs, newspapers were neither printed nor delivered, about 50% of banks in one ex-USSR country were unable to process transactions, one company was unable to send the year’s report to the tax office. Tech support phones were hot as hell, tech support staff had no time for tea or cigarette breaks. Destroyed hard disks, corrupted Flash BIOS chips, lost work, time and money – these are true stories, and I personally saw a woman crying because her husband lost several years’ worth of work. She asked about recovering the hard drive.

“ It was a real world computer catastrophe… ”

The black trail of destruction started in the Far East, where day begins, and followed the time zone, covering the world. People entered offices, turned on their computers and lost them. Those who had infected PCs at home lost data later, when they returned from the office. It was a real world computer catastrophe, and I can’t recall a more powerful one. I approximate the result of the CIH epidemic as half a million incidents – five hundred thousand computers destroyed in one day!

The US and western Europe escaped the carnage, maybe because they had already been scared by Melissa. Newspapers, TV and radio waved flags about it for several days – people duly updated their anti-virus software and the virus had no chance of surviving. Maybe there were fewer incidents in these countries because people there had already got accustomed to using anti-virus software and paying attention to the virus problem. What about the victims of the disaster? Don’t say they didn’t know about it – they did. I’m sure anti-virus companies the world over beat their biggest and loudest drums about 26 April. Did users dismiss it as the usual advertising trash? Does that mean there is too much trash in anti-virus advertising, so users switch off? Maybe.

Did the victims follow the rule ‘it will never happen to me’? That’s more likely. They did know about the virus, they did know about black Monday, they did pay attention to the anti-virus press releases, but they did not obtain and run – don’t mention ‘a complete anti-virus package’ – even a single free small utility to detect the virus in the system (there were several such programs released by anti-virus companies). And so the lemmings died.

What about the Taiwanese originator of the disaster? He has been caught and the whole incident investigated. I’m just curious – will they cart him off to prison or give him a medal? Taiwanese hardware manufacturers will enjoy boom sales because of the incident. When the Flash BIOS is destroyed, with most PCs it is not possible to recover the damaged chip and it is necessary to replace the entire motherboard. In some cases it is cheaper to replace the whole computer.

The CIH episode also epitomised one of the anti-virus axioms – ‘can a computer virus destroy hardware?’ Answer – ‘modern hardware components are protected from possible danger caused by software programs, including viruses’. So, this is no longer valid. By the way, this is the second one; the first was ‘are viruses able to infect text files?’ Answer – ‘definitely not, because the viruses are not able to live there’. Definitely not, if a text file has no macros inside…

So, viruses *are* able to destroy hardware. And they do. The story does not end there – CIH’s author released the source code of his viruses, and the Flash corruption routine is now public knowledge. Any virus-maker is able to replicate it. It is very possible that other DOS and macro viruses will use this routine to kill computers. Indeed, this has already begun – a new multi-partite DOS/boot virus that I saw recently has CIH’s Flash destruction routine in its code.

Until hardware manufacturers fix this problem, users are in big danger. Any new software they download from the Internet may be infected by a virus. Any new virus may have CIH’s routine in its code. This routine may be activated at any time for whatever reason the virus author chooses. Be careful. And check the Flash write-protection switch on your computer’s motherboard. Don’t let a virus cook a ‘Chips omelette’ there.

Eugene Kaspersky

NEWS

Two Be or not Two Be

On Friday 7 May CS/GaLaDRieL, also known as Gala, was discovered. This is believed to be the first virus written in *Corel SCRIPT* which can infect files within *CorelDRAW, Corel PHOTO-PAINT and Corel VENTURA.*

When an infected script is activated, it searches in the current folder for other *CorelDRAW* scripts (.CSC files), parses their contents and finds the names of infected scripts and scripts that are not infected yet. On 6 June a message is displayed. This virus has no destructive capabilities. First research reports conclude that it has a very low prevalence in the wild and can be classified as low risk. A full analysis is scheduled for next month's issue.

Mid-May saw the discovery of Emperor, a CIH-style virus (not a CIH variant) capable of erasing the data on the hard drive and corrupting the Flash BIOS. This is a memory-resident, polymorphic, multi-partite virus about 6,000 bytes long. It infects (16-bit) DOS COM and EXE programs, over-writing the MBR of the hard drive and boot sector on floppies. It uses several anti-debugging tricks, stealth techniques and a difficult algorithm to disable built-in anti-virus protection

ThanY2Ks for the Memory!

VB's old adversary Mark Ludwig has closed down his publishing business *American Eagle*. A somewhat paranoid and doom-laden flyer he sent to the prospective buyers of his remaining stock attempted to explain his motives for publishing computer virus information in the first place. In his effort to 'empower' people – computers being 'Big Brother's tools of choice for enslaving you' – he went on to warn of a national emergency and government cover-ups and conspiracies surrounding the year 2000 date change.

Not content to sit back and watch his life's work declared a national secret, he vowed to sell his books off cheaply and quickly for the good of society. *The Little Black Book of Computer Viruses*, *The Virus Creation Labs* and other 'classics', along with inexplicable but perhaps more intriguing titles such as *The Quest for Water Planets* were offered at cut prices to like-minded survivalists.

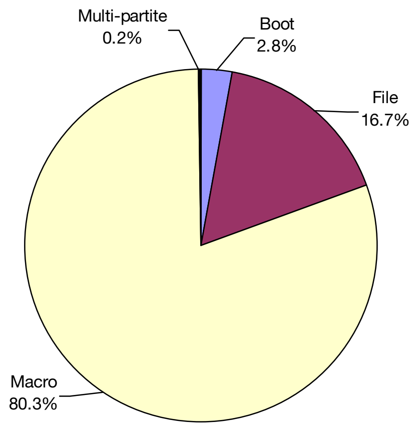
A slave to his own perceptions, Ludwig warned about financial meltdowns and infrastructure crashes come the year 2000. Justification, perhaps, for demanding cash –'any major currency, we're easy – backed up by longwinded explanations about 'politically incorrect' companies having had credit card funds seized. Enough said. His illogically juxtaposed predictions for widespread panic by 'December 1999 or February 1999' seem unlikely but *VB* joins him in suggesting if you haven't purchased a year's worth of dehydrated food yet it's probably too late anyway

Prevalence Table – April 1999

Virus	Type	Incidents	Reports
ColdApe	Macro	926	18.3%
Cap	Macro	505	10.0%
Ethan	Macro	495	9.8%
Win32/Ska	File	463	9.2%
Class	Macro	432	8.5%
Marker	Macro	419	8.3%
Win95/CIH	File	347	6.9%
Pri	Macro	275	5.4%
Laroux	Macro	161	3.2%
Npad	Macro	125	2.5%
Concept	Macro	97	1.9%
Temple	Macro	88	1.7%
Melissa	Macro	80	1.6%
Tristate	Macro	94	1.9%
Appder	Macro	68	1.3%
Munch	Macro	66	1.3%
Footer	Macro	52	1.0%
CopyCap	Macro	36	0.7%
Form	Boot	24	0.5%
Walker	Macro	23	0.5%
AntiEXE	Boot	22	0.4%
Parity_Boot	Boot	21	0.4%
Protected	Macro	19	0.4%
Others [1]		215	4.3%
Total		**5053**	**100%**

[1] The Prevalence Table includes a total of 215 reports across 48 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

[*The* VB *Letters page contains material that is printed at my discretion but* VB *takes no responsibility for the diversity of opinion which can range from anti-virus professional through single user to disgruntled customer. This month's topics include misinformation and generic repair, on which a tutorial will be published in the July issue. Enjoy. Ed.*]

Information, Information, Information

I'm startled by how closely your May Comment page echoes some of the rationalizations of virus writers/ distributors – that Melissa and her siblings are the fault of:

> computer users for visiting (ahem...) recreational sites and newsgroups, or trusting the object because they trust the source
>
> *Microsoft*, for putting functionality ahead of security almost every time
>
> unprepared system administrators trusting in out-of-the-box configurations and out-of-date virus definitions to cope with brute-force mailstorm attacks
>
> managers who haven't locked down systems and implemented draconian policies

It's true that some very nice people are utterly clueless about virus and anti-virus technology. The trouble is, quite a few of them work in security, and some even work in AV. It's not what a virus does that matters: it's what it's believed to do. Melissa spread so quickly beyond its original targets that spammers expressed an interest in it as a model marketing tool in a field where the shotgun approach has always been popular. (Never mind the target, see how widely you can spread the pellets!)

On Parnassus, Gods of Security suddenly became virus experts. Instantly outdated Sendmail recipes based on email headers became popular. Intrusion Detection and Response experts claimed Melissa as an object lesson, even though it was very clearly *not* a targeted attack.

One security organization issued a quasi-advisory statement which not only quoted struggling systems administrators as 'best practice', but also advocated stout firewalling as an anti-virus measure (despite all the doubts of experts in that field). In addition to this, it included the (trivially modified) virus source code, thus encouraging system administrators *et al* to experiment with virus code and practically forcing them to create new variants.

By all means guard against the guardians; not just the home users, not just the corporate users and system administrators, but the marketroids and consultants, tech support boffins and all. Viruses are not primarily a technical problem, they're a social problem. The real enemy is misinformation, and that abounds both sides of the pro-virus/anti-virus divide.

David Harley
Support & Security Analyst
Imperial Cancer Research Fund, UK

[*I thought it only fair to let the author of the May Comment respond for himself. Ed.*]

The hype due to the Melissa scares brought out the worst in the AV marketroids, computer industry and mainstream press. The incidents would not have happened, or their effects would have been minimized, if a few simple precautions had been implemented.

In a 'free society' we can believe that part of the problem is due to inadequacies of the workspace. The fact that the problems do exist mean that malcontents will exploit them. The enemy is misinformation, but the good information is sadly lacking and when it does exist it is in a form that is indigestible. Users look to *Microsoft* for guidance and if they get things wrong or cloud the issue, we get the stick.

Will They Ever Learn?

Some folks think that Y2K will cause disruptions. What happened with CIH in Asia? What happened when 300,000 or more computers stopped working on 26 April?

Did they get them going? Are they on paper now or doing without computers? Can this in any way be a lesson for preparedness for the date roll-over? How come I haven't seen any follow up?

Patrick Neary
Naturopathic Doctor
USA

E-dentify or Not, That is the Question!

Last month a customer sent me a couple of documents which were supposed to be infected with a new macro virus. It was dropping the file 'ETHAN.___' in the root of his C: drive – as some of you may have already guessed, it was a new version of Ethan. Our customer had already checked a different anti-virus site, where some of the most widespread viruses are listed.

Sure enough, Ethan was listed as W97M/Ethan, with no version suffix. As far as our customer was concerned it was enough for him to accuse me of not detecting a perfectly well-known in-the-wild virus. The virus was unknown to me, and as far as exact identification goes it was a totally new version with a couple of new lines added.

It was a version of Ethan.A, similar to most variants floating around but still a new, unknown one. That would not be a problem, but Ethan is known to avoid heuristics of some AV products, including ours. To make matters worse, there are some anti-virus programs out there which simply don't care about Ethan versions, and detect them generically by adding a fixed size CRC.

There are products which can detect almost any new version produced by the Ethan virus snatching a perfectly innocent macro which had the bad luck to be stored before infection in the ThisDocument module. That includes viral macros – new Ethan versions can easily appear from multiple NORMAL.DOT infections.

Many products, including my own, are using exact identification (well, maybe not so exact, but exact enough to distinguish between Ethan variants) and they don't detect new variants of it created in a simple and likely way (NORMAL.DOT files with non-empty ThisDocument modules are not common, but they do exist). There are even many useful programs which plant macros in the ThisDocument module, which, if infected with Ethan, will create a new virus. What is the solution? Our customer will gladly switch to one of those generic-detection programs if they can detect his viruses and my program can't. Exact identification is a noble purpose, but losing a customer just for that is not exactly fun or noble. There is a neat solution – Subroutine-based identification. Each subroutine in a module is checksummed separately, and the virus is detected on a subroutine basis.

The small disadvantage is the increasing amount of data required to identify a virus. Advantages include better detection of new variants, which is the issue here. However, switching from module-based to subroutine-based detection requires some effort from the AV producer, and I don't think I can do it in a short time.

Solution two is to use a fixed-size CRC, either for all viruses, which will lose exact detection, or maybe only for some of them, for example, for our good friend Ethan. Until I decide to switch to Subroutine-based CRCs, I think this is the best for my clients, as it provides detection for new versions, which can even help sometimes in comparative reviews. After all, if we look at some macro comparatives, for example the one from a respectable German university, we will see that the best macro detection is achieved by generic detectors rather than exact-detection-based engines.

The only difference from other programs is that I decided *RAV* reports such modules as Suspicious rather than reporting a known virus. I think this is a fair solution, as I don't want to loose my exact (all right, exact enough!) identification, which is still a noble cause to me (and to few other respectable people out there!)

Costin Raiu
RAV team leader
GeCAD, Romania

When is Protection not Protection?

Why don't anti-virus products reset the 'virus protection' options in *Word*, *Excel* etc while disinfecting macro viruses? More and more often I deal with users of *Microsoft Office* products who are on their second (or greater) infection incident and who have had this option disabled when the virus was contracted.

When questioned they deny having received the warning 'The document you are opening contains macros or customizations'. On further questioning, they recall seeing it months ago when they contracted the first virus. Most are surprised, when they think about it a bit more, that they did not receive that warning 'this time'. Of course, the reason why is simple for us experts. Anti-virus software detects viruses and removes them when found. Changing someone's software configuration is a big no-no from way back.

It is time this changed. 'Way back' most PC users were relative experts. Nowadays this is not so. Most virus victims are surprised that their anti-virus software does not reset all the changes made by the virus they were unfortunate enough to have run. That these changes often cannot be correctly reversed programmatically is a good reason not to attempt many of them. The case of the virus protection options in *Office* applications is different.

Viruses are more prevalent than ever and macro viruses make up the bulk of infections. More new viruses arrive from customer sites than in the past. The only chance most users have against new macro viruses is the built-in warning options *Microsoft* provides. Successful macro viruses disable this weak line of defence the first chance they get – any form of 'protection' should do the opposite.

Demand this feature from your anti-virus developer. Insist that it should not be a configurable option – if it is, the bad guys will just add another few lines of code to their creations to disable this 'fix'. The piffling number of users who 'need' the warnings disabled will know how to 'fix' this, and if they really are expert enough to do without this warning, they should never need to clean a virus anyway, so it will not inconvenience them.

Nick FitzGerald
Ex-Editor of VB and independent consultant
New Zealand

VIRUS ANALYSIS 1

Beast Regards

Péter Ször
Data Fellows

Anarchy.6093, a DOS COM and EXE infector, and also the first known binary virus able to infect *Word 6* documents, appeared almost two years ago (see *VB*, October 1997, p.6). Not surprisingly, we have not seen many other viruses like it since hacking document formats to add macros into them is not an easy task.

On the other hand, Navrhar, created by a Slovakian virus writer, tries to infect known VxDs when an infected document is opened by dropping a 32-bit dropper. There are a few other examples, but all of them share the same kind of limitations. These viruses often corrupt documents during infection since the algorithms used are not at all reliable. Navrhar saves its binary dropper to the end of the documents. Using this approach the dropper code can be lost from documents when an infected, fast-saved document is saved without the fast-saving option.

In such a case, *Word* optimises the document's structure, leaving the dropper code out of the new image. Quite a few similar limitations exist, but most importantly the document structures are just too difficult for the average virus writer to hack, especially when it comes to VBA format. So far all known binary viruses with document infectors attacked the old *Office 6* documents.

A new virus from Russia is the first known binary virus to infect VBA documents. {Win32,W97M}/Beast.41472.A (the new *CARO* standard for naming such multi-partites) is written in *Borland Delphi* and compiled to 32-bit Portable Executable format. Personally, I think that the internal structure of *Delphi* applications is really ugly. Analysing the actual virus code is not a pleasurable experience for any virus researchers, but it is certainly challenging. The disassembly list of the virus is about 1 MB and only 22,000 lines long. Fortunately, the major part of this code is in standard *Delphi* – it can be eliminated easily with the latest version of IDA (Interactive Disassembler) which arrived on my desk just in time!

Beast uses a new method of infection compared to other binary viruses which infect documents. Instead of hacking the VBA format on a bit by bit level, Beast's author uses OLE APIs to inject macro code as well as embedded executable code into documents by using the internal OLE (Object Linking and Embedding) support of *Word* itself.

That makes certain things simpler and much more reliable. Since the virus code is in high level language, the OLE functions are not too difficult either. Interestingly, the binary part does not infect other executables. The virus is active in memory as a process and executed from the registry … \CurrentVersion\Run field with the name of an existing DLL, but with an EXE extension placed in the *Windows* system directory. Beast is already in the wild in Russia and Spain.

Opening an Infected Document

The virus arrives on a new system in an infected VBA document. When an infected document is opened for the first time, the embedded executable gets control from the AutoOpen macro placed in it. First the macro checks if three hours have passed since the virus was active last time. This is performed by checking a registry entry under SOFTWARE\VB and VBA Program Settings called \3BEPb\Startup which is created and updated by the binary part of the virus code. If this was the very first time, the ActiveDocument.Shapes(3BEPb).Activate command will execute an embedded object called 3BEPb (meaning 'beast' in Russian, hence the virus' name).

The actual embedded object is the virus code itself which is 41,472 bytes long and named C:\I.EXE. Normally a visible icon represents an embedded executable. However, Beast hides this icon so it cannot be seen by the *Word* user. *Wordpad* under *Windows98* can show the icon of the embedded object and save the document in old *Office 6* document format leaving the macros out. The embedded objects can easily be deleted in *Wordpad*, making this a solution for manual disinfection. Unfortunately, it is not a blanket solution since *Wordpad* under *Windows 95* and *Windows NT v4.0* does not support VBA documents. When the embedded virus code is executed the virus installs itself in the system.

Installation of the Binary Code

Firstly, standard *Delphi* library functions are called from the entry point of the 32-bit virus code. These functions perform important initializations for *Delphi* libraries and OLE functions and also check for a CD-ROM device, initializing it if there is one since the virus' payload routine is related to CD-ROM devices.

Important texts are encrypted in the data area of the virus body and decrypted one by one when they should be used. The actual encryption is based simply on a shifted XOR key. Each character of the encrypted string is decrypted by using XOR with the actual position of the character in the string, starting from one. The first string which is decrypted is 3BEPb.

After decrypting this string the virus checks if it is already active. Actually this routine has some logical bugs in it and can fail sometimes, meaning that the virus can install itself into the registry accidentally many times. The virus tries to create a file mapping object called 3BEPb and checks if this object has the FDEEF40h mark at its beginning. If it is not placed there the virus assumes that it is not in memory yet and tries to install itself, otherwise it will simply terminate the executed dropper.

The virus code uses Structured Exception Handling most of the time to protect itself from visible crashes. It gets the *Windows* system directory and searches for a DLL name there and randomly selects one (e.g. SHELL.DLL). Then the virus checks if SHELL.EXE exists. If not, Beast tries to copy the executed embedded copy into the system directory as SHELL.EXE. If this procedure is successful the virus tries to update the … \CurrentVersion\Run field to include the name of the created executable, SHELL.EXE in this example. Finally it executes this copy and terminates the other copy executed by the AutoOpen macro.

Installing the 3BEPb Window Procedure

When the new copy of the virus is executed Beast checks again for its existence in memory. Since this procedure has bugs, in some cases it will install a new copy of the virus in registry again. Finally one instance will call a procedure which registers a new window procedure called 3BEPb. This procedure is created as a hidden window of the actual instance of the virus. (*Microsoft's Spy* application can be used to find this window easily, otherwise it is hidden.)

After this the virus sets a timer. A WM_TIMER message will be generated by *Windows* very frequently with the 645h ID from then on. Then the virus waits in a loop until shutdown and halts itself. The hidden window procedure waits for the WM_TIMER messages all the time.

The Window Procedure

The 3BEPb window handles three incoming windows messages. The virus kills its timer or halts execution in the cases of WM_DESTROY and WM_CLOSE. When a WM_TIMER message is coming in, the virus checks if this timer messages has the 645h ID and if so it terminates the timer temporarily.

Next it updates the registry under … \3BEPb\Startup with the actual time. Then it starts to use OLE functions and tries to get a handle to the active document object of the running the *Word* application. When this function fails, the virus checks for the payload conditions and calls its payload routine. After this it restarts its timer again and returns.

When a handle to an active document is available the virus calls its infection module. First it tries to check if there are any embedded objects in the document, but in some cases this routine seems to fail since the virus sometimes adds multiply-embedded executables into the documents. Then it tries to add the executable C:\I.EXE as a Shape into the document named 3BEPb. If this procedure succeeds, the virus tries to add the AutoOpen macro into the default code module using the AddFromString function. The AutoOpen macro code is placed in text format in the data area of the virus body and it is encrypted with the method described above. Beast decrypts the short macro code then adds it to the active document. Document infection is complete.

Finally, the virus tries to close the Visual Basic Editor. As long as Beast is active in memory and the user tries to access the Visual Basic Editor, the virus will close the application in a second at the end of its window procedure.

Payload

The payload is called if the CD-ROM device has been detected during initialization and the time is between 9.35pm and 7.12am. In this time-frame the virus opens and closes the door of the CD-ROM shelf. This happens about every 10 seconds all night long and can cause hardware failures pretty soon. I can imagine the face of a guard watching a few hundred infected PCs in a large computer lab at night with opening and closing CD-ROM doors!

Conclusion

We can expect more and more modern multi-partite viruses which support executable/document infections. This method will replace the BOOT/COM/EXE multi-partite type in the very near future.

Unfortunately, the OLE functionality makes it relatively easy to create such viruses. This means that products which claim to handle macro viruses only will face new challenges since it is not enough to remove the virus module and embedded objects from documents – the virus has to be removed from memory and all executable copies should be deleted also. Only high quality anti-virus products will be able to solve all of these problems.

{Win32,W97M}/Beast

Aliases: 3BEPb.

Type: Win32, *Windows 97* macro virus.

Hex Pattern in EXE files:

```
83EB 0274 1683 EB0E
740A 81EB 0301 0000

7416 EB23 E887 B0FF
FFEB 2A8D 55F0 8B45

08E8 A2FF FFFF EB1D
81FA 4506 0000 7515
```

Payload: Opening and closing CD-ROM door between 9.35pm and 7.12am.

Removal: Use reliable and recently updated anti-virus software to remove the virulent macros and embedded objects from your documents.

VIRUS ANALYIS 2

Papa Don't Preach

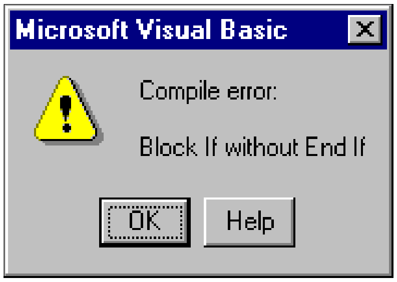
Costin Raiu
GeCAD srl

There are times in an anti-virus researcher's life when the VX scene is calm and there are no terrible new viruses around. Even better, there are times when known, in-the-wild viruses are added to the product a long time before customers are actually hit. However, sometimes we also have hard times when everything seems chaotic – a virus, especially a virus with worm-like capabilities, spreads around the world in a matter of days, and stopping the epidemic requires lots of work and attention.

In the past two months we have witnessed two such unfortunate examples, namely the Internet worm Happy99, also known as Win32/Ska (see *VB*, April 1999, p.6) and the recently encountered macro virus W97M/Melissa (*VB*, May 1999, p.5). There are also similar viruses or worms, perfectly able to spread over the world and hit thousands and thousands computers, which tend to get overlooked because of the success and notoriety of some of their bigger brothers. A case in point is the *Excel 97* worm X97M/Papa.

The History

A few days after the Melissa virus began its ascension, an X97M/Papa.A dropper document was posted to USENET. Fortunately, this 16,896-byte document contained an intended version of the worm. In other words, an 'End If' instruction was missing from the source, so the respective code threw up an error when the document was loaded in *Excel*.



Microsoft Visual Basic

Compile error:

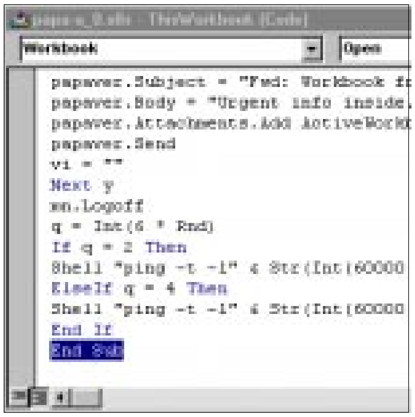
Block If without End If

The document itself was supposed to contain a list of XXX passwords. Instead, it contains the following text, located in the E column:

```
E1:     http://all.net
E4:     For ALL the Latest XXX Passwords
E5:     Free!!!
```

If this original document is loaded in *Excel*, an error message box is opened, stating 'Compile error: Block If without End If'. At the same time, *Excel* opens the VBA Editor, with the worm source highlighted at the faulty line.

Thus, X97M/Papa.A can be seen as 'intended' because of the missing 'End If' from the code. More precisely, it is an intended worm, not an intended virus – except for the bug in the source, the code is designed to replicate from computer to computer, without infecting local files. However, on Monday 29 March 1999, a 'fixed' version was posted to USENET. This new version contains a fully functioning copy of X97M/Papa, in the form of a file called XPASS.XLS, 17,408 bytes long. This version was named X97M/Papa.B, and as stated before, it is not a virus by itself, but a worm. The only important difference between the two versions is that the missing 'End If' from the A variant is now in place for the B version.



```
papaver.Subject = "Fwd: Workbook fr
papaver.Body = "Urgent info inside.
papaver.Attachments.Add ActiveWorkb
papaver.Send
vi = ""
Next y
mn.Logoff
q = Int(6 * Rnd)
If q = 2 Then
Shell "ping -t -l" & Str(Int(60000
ElseIf q = 4 Then
Shell "ping -t -l" & Str(Int(60000
End If
End Sub
```

In addition to this, the source of the B strain is formatted in a different manner: 36 empty lines have been inserted at the beginning of the macro code and the source has also shifted 110 positions to the right. The only reason behind these changes is probably to make inspection of the virus code a little bit harder. Unfortunately, they are not the only measures taken by the author to prevent source inspection.

Functionality

Papa's code is stored in Workbook_Open(), a function which is part of the ThisDocument module. This technique has become very popular as it is an easy way to prevent most users from seeing any suspicious macros in the Tools/ Macro dialog. When a document containing the worm is loaded in *Excel*, the Workboook_Open function takes control. First, the code disables the CTRL+BREAK, ESC or COMMAND+PERIOD combinations which can be used to stop the macro from running. After that, it resets the native random number generator of *Excel*, and tries to instantiate an object of the *Outlook* type.

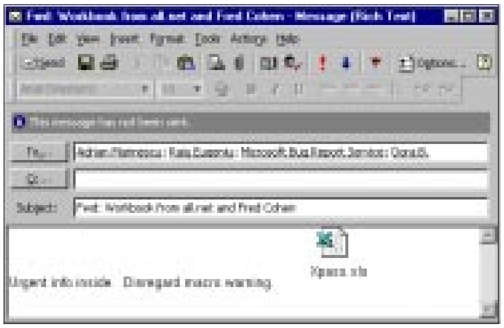
Needless to say, this only works if *MS Outlook* is installed in the local machine. If it is not installed, or if the current system runs *Outlook Express*, a reduced version of *Microsoft's Mail Client*, the worm will not be able to spread. However, if *Outlook* is installed in the system, the worm will try to spread.

First, it will try to log into the default *Outlook* profile. After that, it simply goes through all the *Outlook* address lists, and selects the first 60 entries. It will also create a new email, and add the first 60 entries of the respective address list to the recipients field of the newly composed message.

This message will have the subject 'Fwd: Workbook from all.net and Fred Cohen' and contains the following text in its body:

```
  Urgent info inside.  Disregard macro warning.
```

Finally, the current active workbook (including the worm macro) will be attached to the newly formed message, and the message sent to the Outbox queue. As stated before, the worm will send such a message for each address list in the current *Outlook* configuration. Therefore, the next time *Outlook* is run, the respective messages can be seen in the Outbox. This is what a worm-generated message looks like.


Fwd: Workbook from all.net and Fred Cohen - Message (Rich Text)
File Edit View Insert Format Tools Actions Help
Send
Options...
To...
Cc...

This message has not been sent.

To...: Adrian Marinescu ; Raiu Costin ; Microsoft Bug Report Service ; Dora B.

Subject: Fwd: Workbook from all.net and Fred Cohen

Urgent info inside. Disregard macro warning.

Xpapa.xls

After the messages including the worm host are sent, the virus logs out of the MAPI provider, and runs a short payload routine. The payload is activated on a random basis, with a one in three probability. To do that, the virus generates a random number between zero and five, and if the respective number is two or four, a PING command is launched in the background.

The PING command is run with the '-t' switch – this will make it run forever, pinging the respective host until the process is killed in one way or another. Also, the PING command is run with a random packet size, from zero to 59,999 bytes. Finally, the worm sets the ping timeout to one millisecond, to avoid unwanted delays.

Two IP addresses – 207.222.214.225 and 24.1.84.100 – are targeted by the PING commands. These two hosts are in the ALL.NET and HOME.COM domains, respectively. It is worth mentioning the ALL.NET domain is registered to Fred Cohen, the same name as the one used in the subject field of the emails sent out by the worm. Dr. Cohen was recently involved in the shutdown of a VX site (related to the Caligula macro virus) – a decision which clearly upset the author of the Papa worm.

According to a some reports, they got around 1,000 PINGs per day, which gives an idea of how prevalent the Papa worm is (hopefully, 'was'). On the other hand, there are rumors that the PINGs were not generated by the virus, but by its author(s) in order to trick the anti-virus world into believing that the worm is becoming widespread.

This is not unlikely – spoofing a PING source address is very easy, and there are plenty of tools available. This second hypothesis is also sustained by the fact that there were no Papa reports in the April WildList. Of course, the respective ICMP packets could have been generated by Papa, but without further information, this is pure theory.

General Observations

We should first note that, unlike Melissa, X97M/Papa does not set a marker if the macro was sent out to other users via email. This can be seen as a design flaw, because each time an 'infected' workbook is opened, the worm will send itself to the first 60 recipients in each *Outlook* address list. There is really no reason to send someone a contaminated workbook more than once, except maybe in the hope that if the user does not run *Excel* to check the workbook the first time, they will eventually get bored after five or six such emails, and open the worksheet to see what it does.

Another interesting feature of this worm is the GUIDs in the original documents posted to USENET. With Melissa, they provided some useful information about the computer the virus was written on, as the MAC address of the network board was included in the GUIDs. For Papa, the same information is useless.

The author of the Papa worm was probably familiar with this technique, as we can see from a message posted to alt.comp.virus, in which he blames *Microsoft* for violating users' privacy via inclusion of personal information in the GUIDs. However, while he was overwriting the GUIDs with trash (0x36), he also patched some fields of the _VBA_PROJECT_CUR/PROJECT stream which resulted in a password-protected VBA Project module. Therefore, a suspicious user trying to inspect the emailed document will not be able to see the worm source without special tools.

Conclusion

The Internet is starting to play an important part in our lives. Services like email are becoming available to more and more computer users and the danger of getting a virus (or worm) this way is becoming a reality. Cases like Happy99 (Ska) and Melissa are not accidents, and we now need to educate and inform our users more than ever about the new threats arising at the dark rim of the 'Net.

X97M/Papa

Name: X97M/Papa.A and X97M/Papa.B.

Aliases: Macro.Word97.Papa.A/B.

Type: Email propagated worm, written in VBA and running in *Excel 97*+.

Trigger/Activation date/Payload:
With a one in three probability it launches PINGs directed to one of the following two IP addresses: 207.222.214.225 or 24.1.84.100.

Detection and removal:
Use an anti-virus program able to detect and remove the worm. An anti-virus solution implemented at the mail router/server level is recommended.

A DAY IN THE LIFE

Educating Who?

David Phillips
The Open University, UK

First things first – what is *The Open University* or *OU*? It is a UK teaching establishment that has a central office in a town called Milton Keynes. In the middle of the English countryside, it is fifty miles from London or Birmingham and two hours from the nearest beach.

The University employs around 7,000 staff of whom approximately 3,000 work in the head office. How many students? None. We are a distance learning establishment handling the largest number of students in the world – 200,000 to be precise.

Where Do I Fit In?

I work in the Technology Department looking after a few servers and getting involved in different projects within the University. I started looking at the virus problem here in 1993. Now I look after Technology, advise other areas within *The Open University* and run an advice conference for 60,000 students. Some of the courses we run require the students to have Internet access and we give them access to a conferencing system called Firstclass.

Within this system are a set of sub-conferences related to their course plus a set of other conferences for general discussions and information. I started, and now run, one of these conferences for virus information and help. In its turn, this conference has a sub-conference that holds copies of the anti-virus package *Vet Anti-Virus* which students can download for free. We have agreed a licence with *Computer Associates* (who acquired developers *Cybec* in January 1999) to allow all *OU* students to download this software from our closed conferencing system or, if appropriate, a password-protected web site.

I also post update files for other packages so that students can continue to use any other brand of software that they may have purchased and keep up-to-date drivers. We are certainly not in the sales market of saying ‘use this or that software’ – if it turns out that students have already bought a package, we try to offer some support. If they have not, at least we can offer one for the duration of their study time at *The Open University*.

This idea of supplying software and help to such a large number of students is probably unique in education but the method of teaching employed by *The Open University* lends itself to this type of dissemination of information. It takes teaching a stage further, in that we teach the main courses but offer advice on virus problems facing all students who use computers in study and work.

Email Anyone?

So, how does my day start? The first thing I have to look at when I get in to work is my email, all seven mail boxes plus two main conferences. Two of the mail boxes feed in to the third and they are on the Firstclass system where most students post messages. On Firstclass, I have the mailboxes for an IT support conference which I monitor and the main Anti-virus Support conference, the most important conference.

Here is where students post the highest number of messages, ranging from problems with the implementation of *Vet Anti-Virus*, through virus reports to the obligatory ‘I know this is not the right place but…’ notes. Of the messages received in this conference, 95% are virus-related and on average, there are about 10 to 15 each morning.

Sifting through them can take time. Each one has to be read and I have to categorise the question. This means working out if this a virus report, a *Vet* software enquiry, some other anti-virus package problem, someone answering a query or a ‘Is this a real virus or a hoax?’ question.

On the subject of the latter, I try to keep a sub-conference up to date on all reported hoaxes, getting my updates at Rob Rosenburger’s excellent web site ‘Computer Virus Myths’, *CIAC* or different anti-virus vendors, but students and staff still ask before reading the conference.

One of the other considerations I have to take into account is ‘Is the user computer literate?’. Some users’ questions can be quite technical and they can address issues which range from asking what a virus does, requesting instructions on how to download the anti-virus software, even down to the question ‘Does downloading the software protect my machine or do I need to run something?’.

OU students come from all walks of life. Some of them are very competent computer professionals and even UK sales managers of anti-virus companies. For others it may be the first time they have used a PC. I therefore have to figure out the level of expertise the user has and apply the correct level of explanation to match that expertise.

For those of you who can just walk up a flight of stairs to the user and give them a quick lesson in computer know-how – I wish! These users are somewhere in the UK or, in some cases, around the world and I do not have the luxury of popping along to see them. I have had to learn how to educate users from all lifestyles and at all computer literacy levels about viruses using email. On the odd occasion I do get to talk on the telephone. That takes care of three mail boxes and a couple of conferences. I also check the Firstclass server that is for staff that mostly reside in the Technology Department but this is never too active.

Next, I look at my main *Open University* mail box, which also has a secondary mail box linked to it from the *Lotus Notes* server I administer, which in turn is running anti-virus software checking the servers mail boxes. The main box is where the messages from *CIAC*, *Network Associates* (*NAI*), *Vet*, *Command* and others arrive. Here is where I start checking for new alerts etc. Thankfully, these do not happen too often and I have to say that reports from *NAI* have virtually dried up since the *Dr Solomon's* takeover was announced in June 1998. The *Open University* has a *Dr Solomon's* site licence and I was in regular contact with the Aylesbury (UK) office, but since the takeover I do not get many messages from them.

This mailbox is also the place where virus reports from other Technology Department staff arrive. We have the software that they use set up to mail a report to me if a virus is found. This report lets me know if I need to help as its not been cleaned or if it needs following up to find and notify the source. The last mailbox is my private box and that one normally has messages from family, friends and the Virus Info List.

I also spend time looking at the major anti-virus web sites, including *Virus Bulletin*, to see if there is any new information that needs passing on to staff and students. This can be done in a couple of ways, in the Firstclass conference or via http://antivirus.open.ac.uk – the Technology Department's anti-virus web site.

Which Viruses Do I See?

The two different environments of staff and students provide a varied range of viruses. Since January 1999 the viruses seen by staff at the *OU* have, in the majority, been macro viruses plus a lot of Trojans and joke programs like Geschenk. The range that has been reported by the students shows more diversity with Boot Sector, File infectors and Macro viruses taking equal shares.

CIH has shown up a few times this year with five reports on the 26 April. The problem is not getting the software to the users but getting them to use it. How can you get a home user to download and install it on their own machine, used for work and fun! The only method I have in this type of environment is the conference and the odd article in the student newspaper, *Sesame*. We can only try to educate and when something like Melissa appears in the newspaper, it helps, but hopefully not too often as this can have the opposite effect of breeding complacency in the users.

Time for Coffee!

After I have done all the checks and answered the queries in some way I head for the coffee machine because after this I start my other work in Technology. This ranges from web site design, IT support, Network support, summer school course production, software support, and access database maintenance plus anything else PC-orientated that Technology wants done!

Response Times

If a student is passed to me via either our student helpdesk or a message on Firstclass, I try and sort the problem within 24 hours or as long as it takes to get a set of disks to them. If it is a member of staff the response is immediate – we try to solve all on-site problems as soon as possible.

Take the case of Melissa – I had three messages in my mail box on that fateful Monday morning, two from unknown sources and one from *CIAC*. As soon as I saw that message I went on-line to *Sophos*, *Vet* and *NAI* for confirmation and downloaded the update files. I do not know if I suspected something or it was just that the wife was starting work early that day but I was in work by 8am instead of my usual 8.30ish. By 8.30 the updates and information about Melissa were posted for the students, by 8.45 the updates for the staff were posted and by mid-day Technology machines had the updates installed.

These updates are also used by the rest of the University and the relevant areas responded in the same way, proving that even a diverse operation like *The Open University* can update staff and students efficiently. Even so, this event is making us look at the internal strategy again to see if we can improve response times.

I think that I have a different problem from most supporting staff and students in virus protection and I hope that this gives an insight in how we have tackled the problem with no students on site. We are now trying to get to grips with the larger problem of educating all 200,000 students rather than just those on-line. Interesting logistics problem! Let's not forget that we also have the job of looking after the staff of the University– without them, the students would not be able to study!

All virus reports go via either Roger Moore in our Academic Computing Department or myself and between us we have a database where we log these reports. This gives us an idea of the viruses seen by staff and students and shows how students at home see a different set of viruses from the set seen by members of staff. This information, once verified, is passed to *VB* for inclusion in the Prevalence Table once a month.

Final Word

At VB'98, Vesselin Bontchev stated that people like myself and plenty of others in the same line of work cannot be trusted to pass on accurate information to *The WildList Organization*. I do not profess to be an expert, but Roger Moore and I are the *OU's* resident knowledge base. We both get our information from the real 'Experts' and use it to help and protect both staff and students. After more than seven years I feel it's time these experts looked at teams like ours and worked with us to educate and disseminate good virus protection practice rather than still trying to isolate us. Two papers at VB'99 are centered on the WildList – I wonder if the idea of including organizations like ours has been shelved? Time will tell.

FEATURE 1

Virus Writers – Part 2

Sarah Gordon
IBM Research

I will put off my discussion of why virus writers write viruses until Part 3, next month. This article will examine the question ‘How have they changed?’. If you’ve been counting, this is question number five.

In October 1993, the number of virus writers who were actively contributing to the problem of computer viruses found in the wild comprised a relatively small percentage of the global computing population. The number of their viruses actually causing many problems was quite small too, especially considering the number of viruses known to exist – the number of computer viruses which were reported spreading in the wild was 71 [13], with a total virus count of about 3200 [14]. During this time, there were virus writers known and unknown; working on their own, and working in groups like Phalcon/Skism, RABID, NuKe, Trident and SLAM. They used handles like Dark Angel, Attitude Adjuster and Aristotle. They wrote viruses, placing them on publicly accessible BBSs, FTP sites, and WWW sites, and they kept some to themselves [15].

In some cases, they sent viruses only to anti-virus researchers, because while they wanted to show they could write proof of concept viruses, they did not want to release them to the general public. They wrote viruses that were not released in any way into cyberspace (for lack of a better term), and never caused anyone any problem (other than necessitating their inclusion in scanners ‘just in case’); they wrote viruses that they did release into cyberspace, causing all sorts of problems. They made source code available, and they kept code ‘just for their private individual use’ or ‘just for use within their own group’. They dedicated their viruses to various people, they used some viruses to promote their own groups or identities, and they left some viruses completely anonymous. They attended secondary schools and Universities, and they were professionally employed [16]. They began to beta-test viruses [17].

In May 1999, there are approximately 150 viruses found in the wild [18], with approximately 30,000 known to exist. Some virus writers are pretty well known, signing their creations, while some prefer to do their deeds in secret. Some labour alone, while others work in groups like 29a, SLAM (all-new, all-revised, and not related in any way to the original), The Codebreakers, and The NoMercy Virus Team. They use names like DarkMan, VicodinES and Knowdeth. Some put their viruses up on FTP or WWW sites; some prefer to keep them for themselves. Some restrict their distribution to within their own groups. In some cases, they send viruses only to AV researchers. Some virus writers today release their viruses to unsuspecting users; others do not actively release them. Some make code available, while others prefer to keep it to themselves. Some viruses are dedicated to individuals or causes; other viruses are used to self-promote. Some remain anonymously authored. Virus writers attend secondary schools and Universities. Some are professionally employed. Beta-testing of viruses is pretty common. Sound familiar?

New Bottles, Old Wine?

Some people claim interesting ‘new’ ideas have come out of the virus writing community. Has the ‘creativity’ of virus writers actually started to take on a whole new face? The answer, as is so common when analysing virus writers and their behaviours, is both yes and no. One purportedly new idea is something called (in its current incarnation), Project Zero. It was designed as an ‘experiment’ which should show what would happen if nobody in the VX community released viruses to the public any more for an arbitrary time period of, for example, one year [19].

While this may seem noble, one goal of such a project could be to lull the anti-virus developers and users into a false sense of security. Despite its emergence as a ‘novel idea’, the same idea was tossed around by NuKE affiliates in the old days [20]. Be it vortex or vacuum, the idea is the same, just dressed up in millennium garb.

Another ‘new’ idea is that viruses are actually ‘evolutionary programs’. In particular, several virus writers have recently mentioned to me [21] their belief that replicating code could be used to explore various concepts of artificial life. This is certainly true, but not a new idea; it was explored long ago in [22, 23], to cite just two examples. Additionally, these types of experiments in authentic artificial life concepts are worlds apart from ‘virus writing’ and should probably not be mentioned in the same breath.

Then there is the idea of viruses that could be good entities, also frequently cited as a ‘new’ idea – discussed several years ago in [24, 25]. Padgett Peterson, well-known anti-virus and general security expert says it best: ‘I have never seen a virus do anything that is not easier and more reliable to do without a virus (except be a virus, of course)’ [26].

But Wait, this is New! Really!

One interesting and actually somewhat new idea which has come to light recently may show a slight change in the *modus operandi* of the virus writing community. In the early months of 1999, we saw alleged virus writers and distributors attempting to spread confusion by going ‘public’ on the Internet, registering such domains as datafellowes.com and vgrep.com.

The Codebreakers Internet site, which vanished abruptly in the midst of the Melissa investigation, was reincarnated as codebreakers.net on a site hosted by a large web hosting company in Florida. The site was no amateur-looking hodge-podge. It was particularly well done, with excellent graphics and a 'research' feel to it. It was registered to someone claiming to represent 'DataFellows, Ltd' [27]. At this time, the site appears to have been discontinued, for unspecified reasons. Contact information for the domain refers to an email address located at a different web hosting company in Cupertino, CA. According to the person we asked at *Data Fellows*, Finland, neither the site nor these contact details had anything to do with the 'real' company.

At the same time, www.datafellowes.com appeared on the Internet, hosted by the same Florida-based company as codebreakers.net. As if registering a domain which is an obvious misspelling of an existing and well-known anti-virus software manufacturer were not enough, there have also been reports of misleading email associated with this domain. Several weeks ago, I received an email appearing to be from Mikko Hypponnen, an anti-virus researcher working for *Data Fellows*. The mail requested quite a few viruses. As Mikko is a *CARO* member, such a request seemed unlikely at best, and so I gave it extra scrutiny. Closer inspection showed that the message reply would have gone to datafellowes.com, not DataFellows.com.

Several other prominent anti-virus researchers also received similar requests. Who exactly was the mystery mail sender? I do not know. Clearly, had I complied with the request, I would have been sending viruses to someone who was not the real Hypponnen. Again, at the time of writing, the www.datafellowes.com site does not appear to be operational. Neither the company which hosted the site, nor *Data Fellows*, was at liberty to discuss details of the incident due to police involvement. Another example involves VGREP, a popular utility produced by *Sophos* and available at www.virusbtn.com. It provides a quick and easy reference for virus names. Imagine my surprise when I spotted 'Vgrep Anti-Virus Inc.' using vgrep@hotmail.com as an email address. Is this related to the Codebreakers and Datafellowes events? Only time will tell.

It seems that the 'bad guys' are attempting to confuse the issue by a troublesome (but not particularly creative) manipulation of procedure. The question remains – is this 'new'? Setting up BBSs which appeared to be 'legitimate research facilities' was a favourite ploy of some early virus writers [28]. The mid-1990s saw the same sorts of attacks using email when virus writers pretended to be everyone from Dark Avenger to well-known anti-virus researcher Frans Veldman, and everything in-between. Confusion is the name of the game. So, while the Internet provides some novel twists to the chase, the overall ploy is unchanged; the 'robbers' are pretending to be the 'cops'.

The operational characteristics and demographics of virus writers have undergone some subtle shifts, which began several years ago [29]. While geographic hot zones do pop up from time to time, the advent of cheap connectivity for many has resulted in more global alliances not centred around a particular BBS; the 'Net, as it were, in action. Once relatively regionalized [30], groups that do exist seem more geographically diverse; The Codebreakers group reportedly has seven members from Europe (Austria and Germany), three from the US and one from Australia. Where there are some strongly regionalized groups [31], these regionalizations seem based on language limitations.

Some virus writers are more willing to discuss issues now. This may be partly due to the general acceptability of 'counter-culture' ideas on the Internet *per se*, or the supposedly increased anonymity afforded by various forms of Internet communication [32]. There *is* more willingness to debate publicly – at least on the part of some virus writers and those who favour public availability of viruses. Next month, we examine motivations and justifications. Understanding why can provide some insights which will help us take action that can slow down the viral glut.

13. Richardson, K. 1994. Computer Viruses – The Breadth of the Problem. *Sophos Technical Report.*

14. The WildList. 1993. November. www.wildlist.org.

15. Gordon, S. 1994. Technologically Enabled Crime: Shifting Paradigms for the Year 2000. *Computers & Security.* Elsevier Science Publications .

16. Gordon, S. 1994. The Generic Virus Writer. *From the Proceedings of The 4th International Virus Bulletin Conference.* Jersey. Channel Islands.

17. Gordon, S. 1996. The Generic Virus Writer II. *From the Proceedings of The 6th International Virus Bulletin Conference*, Brighton, UK.

18. The WildList. 1999. February. www.wildlist.org.

19. Private Communication, 1995. Used with permission.

20. Private Communication, 1999. Used with permission.

21. Private Communication, 1999. Used with permission.

22. Dibbell, J. 1995. Viruses are Good For You. *WIRED.*

23. Myers, S. 1995. Computer Viruses: The Infection Spreads to Japan. *Computing Japan*.

24. Cohen, F. 1991. Trends in Computer Virus Research. ASP Press.

25. Stojakovic-Celustka, S. 1994. The Legend – Fred Cohen. *Alive. Volume 1, Issue 1.*

26. Peterson, P. 1999. Used with permission.

27. Internic. 1999. WHOIS. Registrant: DataFellows Ltd. (DATAFELLOWES-DOM)

28. Gordon, S. 1993. Virus Exchange BBS: A Legal Crime? *From the Proceedings of The Use and Abuse of Computer Networks: Ethical, Legal and Technological Aspects.* American Association for the Advancement of Science. National Conference of Lawyers and Scientists.

29. In (17).

30. In (28).

31. Gordon, S. 1999. Viruses in the Information Age. A presentation pre-print for The BlackHat Briefings. National Computer Security Center and Secure Computing. Las Vegas, Nevada.

32. Ahern, T. & Durrington, V. Effects of anonymity and group saliency on participation and interaction in a computer-mediated small-group discussion. *Journal of Research on Computing in Education*. Vol. 28, Issue 2.

FEATURE 2

pOLEmorphism

Andy Nikishin & Mike Pavluschick
Kaspersky Lab

Déjà vu was our first impression when we started examining the polymorphism of OLE viruses. We have seen this all before. Before we examine macro virus polymorphism, let us look back and see how it started with file viruses.

About ten years ago, there were simple viruses like Vienna or Yankee (though it must be said, the latter is rather a complex example which took a couple of days to analyse). These viruses were not encrypted. Then came the age of encrypted viruses like Cascade, Proud and Words.

The first polymorphic virus, called Chameleon, became known in the early 1990s. The problem with polymorphic viruses became really serious a year after that, in April 1991, with the worldwide epidemic of Tequila (the analysis of which took more than a couple of days!).

Finally, in early 1992, we saw the first polymorphic engine – MtE. Suddenly you could get a polymorphic mutant virus from a conventional non-encrypting virus by simply linking object modules – the polymorphic OBJ file and the virus OBJ file – together. All virus researchers were surprised and shocked, anticipating lots of new polymorphic viruses and wondering how to detect them. Now all modern anti-viral scanners can detect strong polymorphic viruses.

Virus construction kits appeared alongside polymorphic viruses. These kits facilitated the generation of virus source texts (in form or assembler), object modules and even the infected files themselves. Needless to say, generated viruses were rather different, but we consider that viruses made in this way can be classed as legitimate polymorphs.

What is Polymorphism?

According to Eugene Kaspersky, viruses are called polymorphic if they cannot be detected using so-called virus masks – parts of nonchanging virus-specific code. (It also applies to those which can be detected this way albeit with great difficulty.) This is achieved in two ways – either by encrypting the main code of the virus' nonconstant key with random sets of decryption commands, or by changing the executable virus code.

There also exist other, rather more exotic examples of polymorphism. A good example of this is the DOS virus Bomber – it is not encrypted, but the sequence of instructions which pass control to the body of the virus is itself completely polymorphic. Polymorphic viruses exist in all kinds of forms, from boot and file DOS viruses to *Windows* and even macro viruses.

Part 1 – The Past: Macros

Windows 95 was released in August 1995 and the first macro virus was discovered. It was WM/Concept. Nobody paid serious attention to that fact, as a result of which, virtually all the anti-virus companies were not ready for what happened next. There was a a macro virus epidemic and these companies started to work out quick but inadequate steps in order to put an end to it. The first macro detection engines built on the infected macro's name to seek for macro viruses. Virus makers took advantage of this weakness in macro scanners to create the first semi polymorphic macro virus – WM/Outlaw.

Its technique is polymorphic in that it changes its macro names on every infection and stores them in WIN.INI. If *Word* is restarted, the virus reads the names of the macro from WIN.INI. The names generated are just a character plus a number, e.g. A326 or RT898763. The reason that it uses characters at the beginning of the name is that *Word* can only use macros that begin with one. The polymorphic mechanism of this virus was extremely simple (but its source code rather big). The virus gets the current time and depending on the current hour generates the first letter for a future macro name, the rest of it the virus generates using *Word's* random generator. Outlaw caused some detection trouble to those anti-virus companies which did not employ a good macro engine.

The First Real Polymorphic OLE2 Viruses

WM/FutureN is generally considered to be the very first polymorphic *Word* macro virus. It generates random names and uses WordBasic's EditReplace command to change the original name to generated ones. During its spread the virus changes the name of its macro and that of only one of its arguments, but it still illustrates how to write *Word 95* polymorphic viruses. Using such technologies several polymorphic macro viruses were created, for example WM/MiniMorph – a very small virus which does not have arguments with a constant name. Then came viruses which inserted random comments and labels into their code (WM/PolyPoster, WM/Junk) and finally WM/UglyKid.

The latter virus (see *VB*, October 1997, p.8) uses quite a complex polymorphic engine. Different infected files have variable sets of commands in the virus' macros. Depending on a random counter the virus inserts garbage code after every line of significant code, which gets random data or the current time or inserts a comment string or leaves this line blank. Mutating this way, the virus can overgrow its size. To avoid this it creates a new macro instead of copying the old one. It uses quite a complex method to hide its main code in documents and templates – the main virus code is placed in the AutoText area and the virus' macros just read it from there, copy the text to the macros area and execute it. This was the first known virus to hide itself in this way which is very difficult to detect.

The *Word* macro virus K302 has an unusual structure – the text of the code's main infection routine is commented. The virus creates a temporary macro, copies its text to it, removes comments, executes and deletes it. This is good way to fool heuristic analysers. This virus also has a polymorphic routine, which modifies the decoding ('unremming') routine. Modification is concluded by adding garbage code not only to functions but also to some collation strings and by the initializiation of some variables, even calling some WordBasic functions. This virus also adds a number of 'goto' statements that go to the next line.

W97M/Slow has a rather strong polymorphic engine and contains all of the features described above. On each infection the virus' polymorphic engine renames internal virus variables to randomly selected names of randomly selected lengths. As a result there are no constant search strings with which to identify the virus.

Fortunately there were no tools to modify the source code in *Excel 95* and *Access* and we do not have any polymorphic viruses for these applications. Do you see the similarities with file viruses? The first macro viruses were so simple, then they became weak polymorphs and finally they were transmuted into strong polymorphic beasts. That was our past. Now it is time to move on to the present.

Part 2 – The Present: MS Office 97

The release of *Office 97* saw a new era. Visual Basic 5 was accepted as the integrated standard programming language. *Office's* application gave software developers a lot of new abilities, but it gave them to virus writers too. The universal object model of VBA lets applications interact with each other. Besides, the single syntax of VBA5 for all *Office* programs means that one macro code can be written for all applications. Virus makers took full advantage of these features to create new cross-platform viruses such as O97M/Triplicate (see *Virus Bulletin*, February 1999, p.4) and O97M/Cross (June 1998, p.11).

One more associated feature worth a mention is the useful and easy procedure which modifies the source code of macro modules during run time. Taken together it might appear that virus writers were handed a trump card but in spite of all these features they very seldom use non-traditional methods to write polymorphic routines in viruses. They still use tricks taken from *Word 7* viruses, fortunately for all of us. Most polymorphic macro viruses are incredibly similar because they use the same polymorph procedures. Here are a few of the most 'popular' ones.

VAMP v1.0 stands for Vic's Advanced Macro Poly. This polymorphic macro engine is used by many others such as W97M/Bench and W97M/Pri. The engine replaces variable names in the code with randomly generated ones.

APMRS uses another polymorphic macro engine, which inserts a few random numbers into the virus body, generated every time it is run. As a result, its size increases with every run. This virus may also cause detection problems because of the huge volume of garbage code. This engine is used by W97M/DMVCK1.C, W97M/Splash and others. Every line of the *Word 97* macro virus Sin has its own label. A polymorphic engine changes label names and lengths randomly. It is quite slow and makes unworkable code if any line of the original code has a colon in first 20 characters. There is a modification of this engine, which inserts garbage code like 'YYY = LLLL' in certain places.

W97M/SWLABS99.B uses another feature – the Visual Basic interpreter. If the string of virus source code ends with '_', the next string of code is a continuation of previous one. The virus gets its own code lines and splits them into several groups depending on a random counter. The code is not changed too much, but becomes quite unreadable and obstructs analysis of the virus.

Though most viruses use similar polymorphic engines, there are some very unusual and original ones that use non-standard solutions, for example W97M/Walker.B. Every line of code of this virus starts with a digital label and ends with a 'Goto' statement pointed to the next line. The first line of code is a jump to the first meaningful code line. The virus mixes its code lines in random order, but still works.

There is one more aspect of polymorphism in macro viruses – manual polymorphism. It is very easy to obtain the source code of a macro virus. Therefore, anyone can modify the original code. Everyone is familiar with the examples of this – W97M/Wazzu, XM/Laroux and W97M/Concept.

Virus Constructor Kits

A virus constructor does not constitute a virus by itself. Virus constructors are special tools which generate new viruses without the use of programming. 'VicodinES Macro.Poppy Construction Kit' (or VMPCK) for *Word 97* is the most notorious constructor. It can create viruses based on four fundamental parameters – infection strategy (14 variants), stealth procedures (6 variants), payloads and triggers (10 variants) and extra code (9 variants). This kind of 'badware' may cause a headache for virus researchers.

Part 3 – The Future: Office 2000

It is always a little daunting to predict the future – what if predictions come true? When we discussed this article we tried to dream up a future for macro viruses and we were afraid of our ideas.

It is not too difficult to make links with the past. Just imagine – by using VBA5 features a virus is able to modify its code beyond recognition during run-time, which can make it very hard to detect. Using polymorphic engines together with encrypted code may result in… time to improve our scanner!

TUTORIAL

When Barriers Break Down

William E Jones & Christine Orshesky

Traditional barriers, which included anti-virus protection installed on each desktop or the standalone systems used to scan all diskettes as they came into or out of a facility, are no longer sufficient. We have been forced to move from protecting only the desktop to suites of anti-virus products that provide multi-tier protection at all points of entry to a network such as email servers and firewalls.

Many administrators and organizations have experienced times when their favourite anti-virus or other security product did not measure up to the newest threat and the ‘malicious’ item made its way past their barriers and into their networks, causing them to respond instead of defend. Macro viruses like Marker and Melissa clearly demonstrated to many organizations the need for a more diverse response mechanism.

Moreover, it is becoming imperative that every company establish a posture or strategy that will actively defend, manage, and respond effectively to malicious logic incidents – namely a ‘response in layers’ approach including much more than just anti-virus products. Many of the protection strategies and incident management techniques outlined below are drawn from actual incident response initiatives and case studies.

Attack in Progress – Recognizing the Symptoms

To determine if something malicious has made it through one must know what is protected behind the barriers, when it changes, and what the changes may mean. Traditional processes like configuration management, auditing, and user awareness are all good ways to be able to identify the symptoms of malicious code – in addition to any installed anti-virus product that is running on a given platform.

Configuration Management: knowing your system content and configuration is essential to enabling you to detect when something has changed. Such things as configuration control and integrity checking can provide visibility into changes in a system or network.

User Awareness: you can have eyes and ears at every level of the system. When users are provided with good guidance and clear reporting procedures, they are able and willing to report suspicious activities and can sometimes be your early warning system.

Review of Auditing Information: turn on any auditing that is available. This will provide additional information about unauthorized access and file changes that can identify a possible attack. Auditing can also help to piece together what occurred after an attack is identified. Capture enough audit data to make it meaningful but not so much that the system performance is severely impacted.

When reviewing audit data, look for unusual port usage, unauthorized file access, file permission changes, and connectivity with suspicious sites or organizations, particularly where file transfers are seen.

Containing the Breach

When malicious logic (or indeed an intruder) has compromised a system or network and thwarted the protective barriers, a concerted effort must be made to contain the spread of the virus or other malicious code. The actual mechanisms to isolate the affected system(s) will vary, but disconnecting the affected area from external (or internal) contact via a network or user is generally the best approach.

Disconnect all affected systems from the network – thus, you will prevent the malicious code from spreading. This is particularly useful if the virus is spread via email. In some cases you also prevent it from performing its payload, which may include sending information via FTP, or from pinging a remote site.

Block adversarial IP addresses or close ports – depending on the nature of the malicious logic, the payload may include uploading information via ftp to a remote location, potentially a hostile or competitive organization. In these cases, disabling the use of certain ports can help to contain the virus and to prevent you from advertising to an adversary that you are indeed infected.

Investigating the Incident

Once you have quarantined the affected systems and contained the breach, an investigation as to the source and type of breach and the potential recovery methods must be completed as quickly as possible. An analysis of the audit data and an inspection of the system(s) is necessary to determine the source and scope of the breach.

Analyse logs – damage assessment is usually done by reviewing the audit logs from the system or network for changes in configurations, unknown entries, unauthorized file access, unsuccessful password attempts, and suspicious network activity. Careful analysis may provide a clearer picture of how the malicious logic entered the system, what it did once it arrived, and where it may have travelled (some relatively recent viruses have carried a log along with them, providing a kind of travel itinerary).

Inspect the system(s) – looking for Trojan horse programs or other forms of malicious code is crucial to the successful recovery of the system and can shed light on the extent of the damage or infection. An active configuration management process that manages the standardization of system configuration and the verification of the integrity of the system facilitates this. Not only should you look for the immediately obvious items, look for backdoors that may have been inserted during the system compromise, such as automated scripts which upload potentially sensitive data to a remote location.

Planning the Defensive Attack

Preparing for recovery and prevention of recurrence is essential. You already know that you have been infected or attacked by a certain virus or other form of malicious logic. Contact the vendor – if you have not already done so, contacting your security product vendors, particularly your anti-virus vendor in the case of a new/unknown virus is essential. A patch or fix may allow you to recover your systems with the least effort and minimal data loss and should be thoroughly explored before more destructive methods, such as file deletion or drive formatting.

Formulate the plan – with the information from the investigation and the possibility of a patch or fix, it is time to assemble the system administrators and other key system and network personnel affected by the incident, such as email administrators. This team will develop the strategic and tactical approach to recovering from the incident as well as preventing the incident from recurring, if possible. This approach should be in the form of easy to follow steps, which, if necessary, can be repeated at a later time.

The Clean-Up Effort

Performing sweeps of the systems with anti-virus software can effectively stop the continued spread of known viruses and malicious codes from propagating throughout the system or networks. It can also provide a relatively effective method for recovering infected files when a clean backup is unavailable. In some cases, complete recovery may include labour intensive steps, depending on the payload and complexity of the attack. In addition, complete recovery will also include the verification of the eradication effort, such as rescanning a system once a virus is removed.

Notifying Others of the Breach

There are three main parties who should be notified:

System Administrators – informing the administrators of systems or networks connected to yours, as well as potential sources and recipients of the virus will aid in the containment of the infection. It will also allow other system administrators to be on guard for suspicious behaviour on systems under their control.

Management – it is very important to keep management informed. If there is a need for a legal investigation, management needs to be aware of the circumstances and potential impact of such an investigation. Another important aspect of this is management's ability to provide concurrence and enforce compliance with recommended actions, easing the 'pain' of performing some unpopular and adverse actions, such as disconnecting services or users from the network during an incident.

Users – as mentioned previously, they are the eyes and ears for the system. Unfortunately they are also the hands that introduce malicious logic into the system. Keeping users informed can aid in the containment. It is not imperative nor should it be encouraged to provide them with details. However, information on the general ways in which the system can be (or has been) compromised and the impact it may have (or had) on their data can help to heighten their awareness and help bring the safe computing messages directly into the user's reality.

Documenting the incident's processes and procedures can be valuable in preventing future compromises. Documentation can help to alleviate hasty decisions and provide a step by step account of the recovery effort for future reference. It is also important to note the source and scope of the incident. This provides valuable insight into the incident trends and the effectiveness of the protections in place.

Planning for the Next Attack

Contingency planning is one way to incorporate lessons learned into our defensive posture. Administrators and technicians must know the limitations and vulnerabilities of their systems – which port or router is most susceptible to which type of attack, or at what point in the system architecture anti-virus protection can be most effective. Network systems should be protected using defensive layers – routers, firewalls, anti-virus software, security scanners, and integrity checkers. Systems should be scanned at scheduled times and networks assessed to identify vulnerabilities before they can be exploited. All known vulnerabilities should be documented along with the specific risk. This may mean installing vendor patches or reconfiguring networks to close specific holes.

Summary

As we have seen in many of the more recent incidents, malicious logic threats have the ability not only to disrupt and corrupt data, but also to send data outside an organization, potentially exposing the data (or the company) to unfriendly recipients. With increased capabilities of viruses, Trojan horses, and other forms of malware, it is no longer enough to install an anti-virus product on particular platforms and consider yourself protected. When the traditional barriers break down, a cohesive and effective process must be poised to manage and respond to the resulting incident. Most elements of an effective response cannot be bought off the shelf. These elements are knowledge, experience, and skill – it takes preparation and time to incorporate them into your working environment. It takes more than anti-virus products and a few technicians to prevent, manage, and recover from malicious logic attacks.

PRODUCT REVIEW 1

DialogueScience AntiVirus Kit v3.0

The spotlight this month falls upon a Russian product as we take a look at *DialogueScience's AntiVirus Kit* (*DSAV*). The complete package provides the user with a file integrity checking module, an anti-virus package, a mail scanning module (optional) and a card security module (optional). In this review, we investigate the first two components of this package – *Advanced Diskinfoscope* for *Windows 9x*/*NT* and *DrWeb32* v4.10. Are the programs really two modules of a complete protection kit, or merely two standalone products bundled together by the marketing team?

Installation

For the time being at least *DSAV* is only distributed in electronic form and so 'the fully boxed product' which we normally require for standalone reviews was not available. Instead, a CD was supplied, containing the programs and the on-line documentation.

The product was installed onto *Windows 95*, *98* and *NT* platforms. The CD-ROM autoruns when inserted into the PC, and following the *DSAV* splash screen the installation routine begins under the fatherly guidance of *InstallShield*. Due to the electronic distribution of this product, the lack of a manual came as no surprise. What was a surprise however was the lack of any readme file prior to the installation process. It may be that the product has been designed to be used with on-line help only, but as pointed out within those help files – 'it is recommended that you do not install *ADinf32* until you have scanned your hard disk with *DrWeb32* and a disk utility package'. A touch of chicken and egg syndrome since this message cannot be read until you have already installed the product.

The usual choice of typical, compact or custom installations are offered by *InstallShield*, following which a summary of the components the user has selected to install is shown. In choosing a custom install, options to install *ADinf32*, *DrWeb32*, and *SpIDer Guard* for *Windows* are provided.

Once the chosen options are confirmed, file copying takes place. Subsequently the user is prompted to reboot the system when installing on an *NT* machine, but not on *Windows 9x* machines, despite the fact that changes to system files have been made in all cases. Upon rebooting, two icons are placed on the desktop to load either *DrWeb32* or *ADinf32*. Alternatively, each of the programs can be loaded from the *Windows* Start Menu.

What is Advanced Diskinfoscope?

Strictly speaking, file integrity checkers are not pure anti-virus products. However, a product capable of monitoring the size, date attributes and (in some cases) content of files is certainly a logical weapon to use in the anti-virus war. It is with this in mind that *Dialogue Science* have integrated *Advanced Diskinfoscope* into *DSAV*. The product is supplied in two versions, a 32-bit version (*ADinf32*) for *Windows 9x* and *NT*, and a 16-bit version (*ADinf*) for use in DOS and *Windows 3.xx* environments.

The help files for *ADinf32* are well set out and reasonably thorough. The first few pages present a brief description of the various types of computer virus, and offer some general computing advice where the importance of learning how to manage files, perform frequent backups and correctly use anti-virus software is stressed. Following this, the use of *ADinf32* is described in detail with frequent screen shots to help familiarise the user with the product.

The principles of operation of *ADinf32* are reasonably straightforward. Upon disk scanning, information concerning the logical drives, boot sector, bad clusters and file tree is recorded. Additionally, a form of checksumming is used to verify the integrity of files – a Cyclic Redundancy Check (CRC) of each is performed. All this information is stored in data files referred to in *DSAV* as "diskinfo tables".

When first executed, *ADinf32* has to scan the PC in order to build these files. Once created, they act as references to which fresh data from subsequent scans can be compared. As well as the standard disk scan, *ADinf32* also provides a facility to search for stealth viruses. For this it compares the information returned by the OS to that it reads direct from the BIOS (*Win9x*) or physical device (*WinNT*) – any differences suggest of a suspected stealth virus infection.

DSAV uses different CRC methods depending upon the format of the scanned file. Of particular relevance to current computer virus threats, *ADinf32* parses the macro components from *Word* and *Excel* files and includes only this information in the file CRC. Thus only changes to the macro content of such a file will actually trigger *ADinf32* into thinking that it has changed – additions to the content of the document or spreadsheet are ignored.

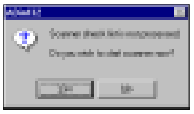
Using Advanced Diskinfoscope

Without a hard copy of any user manual to hand, there are two ways of approaching *ADinf32* – either peruse the help files religiously, noting down the important points, or just point, click and use the on-line help when necessary. Initially, the latter approach was chosen and the ease with which the program was used bears testimony to the logical and user friendly interface.

As with any ‘non-essential’ utility program, if *ADinf32* encroaches too much into the user’s normal routine, it would not be used. Pleasingly therefore the overhead in running *ADinf32* is quite low. When first executed, it prompts the user to create the diskinfo tables for each of the fixed drives. This process is reasonably quick (approximately 15MB/sec of data was catalogued during testing). Subsequent disk scans were performed at a similar rate, and so if scheduled to run at *Windows* startup they would be finished in the time taken to make the morning cup of tea. Scanning for stealth viruses was slower however, with scan rates of approximately 5 MB/sec being achieved.

The concept of profiles is used to configure and manage *ADinf32*. Besides the default and boot-up profiles that are part of the package, additional customised profiles can be created in order to automate integrity checks of the system. To facilitate this process, it is possible to copy one of the existing profiles, make the required changes and save it as a new profile. Information as to what drives and files are scanned, and what action should be taken if any changes are detected is stored within each of the profiles. Thus it is possible to configure *ADinf32* to scan certain file types or specific files/folders at each (or just the first) *Windows* startup.

If changes are detected during a disk scan, the user is prompted to view the results in an *Explorer*-esque window. From this window, right-clicking any of the displayed files enables options to Scan for viruses (using *DrWeb32*), View properties, or apply Special marks (e.g. hide changes, declare file as stable). A detailed summary of all the changes is also written to a log file (ADINF32.LOG), and a list of the files in question are retained in ADINF.LST. Upon quitting *ADinf32* the user is asked whether the diskinfo table should be updated to reflect the changes. For changes that *ADinf32* considers non-suspicious, the focus is on the ‘Update’ button. Following this, the user is prompted to scan the non-suspicious files for viruses – exactly which anti-virus scanner is used (be it *Windows* or command-line based) can be configured within the profile setup. The scanner can be set to start automatically immediately after quitting *ADinf32* (with no user prompt).



Certain changes are considered suspicious by *ADinf32*. For example changes to the boot sectors (master or DOS), changes to files that have been declared as ‘stable’, or the existence of peculiar date/time file stamps all trigger it into suspecting a virus infection. Additionally, changes to the macro content of *Word* and *Excel* files are considered suspicious. Following a scan in which suspicious changes have been detected, the user is warned, and after having viewed the results prompted (logically) not to update the detected changes to the diskinfo tables. Subsequently however, a peculiarity in the action that *ADinf32* undertakes was noticed. Above it was noted that upon detection of freshly created files the user is prompted to scan them. Surprisingly though, this same action is not recommended when the detected changes to the file system are considered suspicious. Surely a more satisfactory integration between the integrity checker and the scanner would be to enforce the scanning of suspected objects?

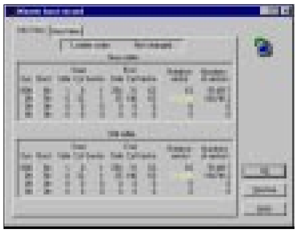
Virus ‘Detection’ with Advanced Diskinfoscope

The use of *ADinf32* as a protective measure against viral infection was then tested. Infected *Word* documents were copied into the My Documents directory and the diskinfo tables were updated to include these infected files. Subsequently, goat files in the same directory were infected from these documents. *ADinf32* successfully detected that the macro content of the goat files and the NORMAL.DOT template had changed, and warned that this was potentially due to a macro virus infection.

Thus far, mention has only been made of *Word* and *Excel* files – this unfortunately mirrors the situation as far as *Dialogue Science* are concerned, since at the time of writing *ADinf32* does not cope successfully with *PowerPoint* and *Access* file formats. Modifying the configuration profile to associate PPT and MDB file extensions with macros made the situation worse, since the increase in total file size upon macro infection is ignored as *ADinf32* attempts to parse the macro content.

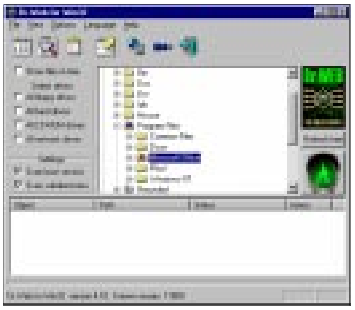
As part of their replication cycle, some macros deposit files on to the hard disk – i.e. W97M/Ethan.A and {Win32/W97M}/Beast (see p.6 of this issue). *ADinf32* provides a useful tool for monitoring such activity, detecting these dropped files which may otherwise get missed by anti-virus scanners especially if they possess odd file extensions.

Boot virus infections were also successfully detected by *ADinf32*. Details concerning the boot sectors before and after the change are displayed, with the changes highlighted. The user has the option of restoring the original configuration, although it would be nice to have the option of scanning the boot sectors for viruses prior to this decision, since the change may be due to a legitimate reason.



DrWeb32 anti-virus scanner

Lovers of stylish logos and dramatic splash screens will not gain satisfaction from *DrWeb32*. The splash screen displayed upon loading is more reminiscent of early *Windows 3.xx* products. The program is easily controlled either by using the drop down menus, or from the buttons on the toolbar. These buttons enable the user to view folders to scan, view scan results, show scan statistics, clear scan



results, obtain updates (via a TCP/IP connection), configure setup options, or exit the program altogether.

A slight lack of attention to detail is apparent in the on-line help files supplied with the submitted product. They seemed to have been designed for a slightly older version of *DrWeb32*, since some of the screen shots seem to be missing certain options that have more recently been added.

Using DrWeb32

Fairly standard configuration options are offered by *DrWeb32*, but there is no concept of user profiles to enable specific tasks to be set up and then repeatedly performed by a single click. Instead changes to the configuration must be made manually prior to each scan.

By default, files are scanned 'By format', although this can be changed to 'All files', 'Selected types' (i.e. by extension) or 'User masks'. Archives and packed executables are included in the scan, and there is an option to scan 'E-mail files' (although this is greyed out unless the Mail module of *DSAV* is also installed). Currently the product is only designed for use on workstations and so infection reports are only written to a log file and to the screen. Presumably as the product is developed to provide network functionality, remote reporting options will appear.

The action *DrWeb32* takes upon finding an infection depends on whether the file is considered infected, incurable or suspicious. Options to report only, attempt cure, delete, rename or move the files in question, are provided for each of these three conditions.

As is becoming increasingly popular in anti-virus packages, a facility to obtain updates from a remote site is provided, and assuming the user has a valid username and password they can be obtained (weekly or monthly) on a day of your choice from the *DialogueScience* FTP site automatically.

In terms of scanning speed *DrWeb32* has changed little from that reported in recent Comparative Reviews, lying on the slower end of the scale relative to other anti-virus scanners. If using the *DSAV* package however the scanning speed of the anti-virus module is less important than that of the integrity checker, since only selected new or modified files will regularly require scanning.

SpIDer Guard provides the on-access component of anti-virus protection. Following installation and a system rebbot, it is automatically started but can also be loaded from the Start Menu. An icon in the taskbar confirms that *SpIDer Guard* is loaded, and double clicking the icon gives access to the configuration options. These are very straightforward and reminiscent of those for *DrWeb32*.

Detection Rates

Throughout recent comparative reviews, *DrWeb32* has shown itself to be up there with the big names in terms of detection rates, setting an example to some major products. This time around, detection rates were excellent again. For both on-demand and on-access scanning, 100% detection against the ItW (file and boot) and Polymorphic test-sets was matched by 99.9% detection in the Standard and Macro sets. Templates infected with W97M/Boom.A.De and W97M/ZMK.F, and a Win32/Ska infected DLL file accounted for the misses during on-demand scanning. *SpIDer Guard* missed these samples also, as well as MDB files infected with A97M/AccessiV (A and B variants).

Conclusions

As with other similarly packaged programs that are available, the question of integration between the modular components is of particular interest. In *DSAV*'s case, the results suggest that integration has only been weakly implemented thus far. Notably, the package could be greatly improved by altering the way in which 'changes' detected by *ADinf32* are handled. The issue of updating the data files (diskinfo tables) of an integrity checker with recent changes to the system is of fundamental importance to its successful operation. After all, updating the data files to include potentially infective files (essentially validating them) completely undermines such a program. It is interesting therefore that upon detecting (non-suspicious) changes to the system *ADinf32* prompts the user to update the diskinfo tables prior to scanning the files in question with *DrWeb32*.

Another area which needs attention is the handling of *PowerPoint* and *Access* files by the *ADinf32* module. O97M/Triplicate.D made its première on the WildList last month and with the current prevalence of macro viruses this is a matter of some importance.

The developers have informed *VB* that extensions to the package to include a 32-bit *ADinf Cure* module (to aid file recovery), *Windows NTFS* support and network administration are under way. Coupled with the resolution of the few problems uncovered during testing, such additions will no doubt fortify what is a promising product.

Technical Details

Product: *DialogueScience AntiVirus Kit.*

Developer: *DialogueScience Inc*, 40 Vavilova St., Moscow, 117786, Russia; Tel +7 095 9382970, email sales@dials.ru, WWW http://www.dials.ru/.

Availability: *Windows 9x/NT,* 16MB RAM & 10MB disk space.

Version Evaluated: 3.0.

Price: Annual subscription (inclusive of full updates) – $29 for *ADinf32*, $49 for *DrWeb32*. Contact distributor for multiple or site licence details.

Hardware Used: 166MHz Pentium-MMX with 64MB of RAM, 4 GB hard disk, CD-ROM drive and 3.5-inch floppy, running *Windows 95, 98* and *NT.*

[1]**Virus Test-sets:** Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/Win98/199905/test_sets.html.

PRODUCT REVIEW 2

eSafe Protect Enterprise

Martyn Perry

eSafe Protect (*ESP*) provides various elements of protection aimed at Internet users, with virus protection being one aspect in the whole range of services. From previous experiences with multi-function products, the quality of the virus detection tends to suffer at the expense of other modules. Let us see if that is true this time.

The licence is limited to installation on a single computer. For multi-server deployment or deployment of scanners onto PCs, additional licences need to be purchased.

Presentation and Installation

Normally, this product comes on CD but the test version was installed from a downloaded self-extracting file. The first thing that happens when it begins the setup is to offer to check for the latest version via the Internet.

A choice of network is presented, namely *Windows NT*, *Novell NetWare* or other. Choosing the *NT* route led to a request for confirmation that the computer was an *NT* Server on the designated domain. Following on, the default destination directory can be chosen or you can opt for a user defined location.

The next prompt asks whether the installation will be either for evaluation purposes or a full registration. The latter requires a 17 character code (supplied with the packaging) to be entered. If only a 30-day evaluation is required, the installation can continue.

There follows a sequence of file copying and then an option to install an on-line alert facility. This sends a detection message to a specified email address. For this facility to operate, an email address and SMTP server need to be designated. This option was not used on the test configuration. The next option is to obtain automatic downloads of updates. There follows a warning that ‘sandboxes’ for some Internet-enabled applications are placed in learn mode. This lasts for 14 days and monitors the typical usage patterns.

eSafe Protect Gateway

The *eSafe Protect* Gateway provides content filtering for different types of connections made though firewalls. This includes the scanning and cleaning of SMTP, HTTP, and FTP connections. There is also a facility called a Sandbox which can be used to configure the access to directories. Access limitation can be defined manually. Alternatively, a learn mode can be employed to monitor normal usage so that a profile can be drawn up of the access needed for day to day activity.

Blocking rules can be defined for files and alerts issued if the rules are breached. Files which pass through can be virus scanned using the integral virus scanner. The anti-virus module is subdivided into three components, namely On-demand scanner, On-access scanner, and Environment. It is possible both to strip macros from their text components and remove cookie, scripts and applet tags from some HTML pages. Archive files are inflated and then scanned. If an archive is compressed more than once, *ESP* Gateway will keep inflating these recursive archives until the core files are reached.

Manual/On-demand Scan

The configuration options for on-demand scanning are not available on the console. They have to be set up on the workstations. This seems to miss the trick of being able to pre-configure the majority of workstations and issue the settings when scans are deployed. The administrator display message states ‘The on-demand scanner does not affect normal operation or present a security risk. Therefore, it is not configured by the administrator.’

The default set of files to include in a scan are COM, EXE, DO?, XL?, VXD and DLL. Scan methods are Standard scan, Smartscan, Remove integrity files, and Scan and analyse. The choices of scannable items are Archive files, Macro files, All files or File extensions. There is an additional option to activate a progress display.

For reporting purposes, a file can be defined (default C:\ESPLAN\DEFAULT.REP). There are options to overwrite an existing file, append to it or limit the appended data up to a maximum file size. The level of detail in the report information can also be selected. There is a separate choice of responses depending on the infection type. These types are Removable virus, Non-removable virus or File modified. For the first option you can Ask user, Notify user, Delete file, Remove virus. For Non-removable virus the choices are reduced to Ask user, Notify user, Delete file. Finally, the File modified choices are Ask user, Notify user, Recalculate file. In all instances there is the option to write to the alert file.

On-access Scan

The configuration for this screen consists of two tabs, Operation modes and Scanning activities. The scanner can be configured to run silently (i.e. not displaying virus warnings) or not, to use a Standard scan or SmartScan and to determine which file types are to be scanned. A Standard scan looks at files for known viruses but does not create any integrity files. The SmartScan uses integrity files to detect unknown viruses and determine whether to scan for known viruses.

Integrity files contain checksum values for files stored in a particular directory. If the directory does not have an integrity file, then the scanner scans for known viruses and creates one in that directory. If the directory does have one, the scanner compares the file against the contents of the integrity file. If they are not consistent, the file is scanned and the integrity file updated. If they are, the scanner does not scan for viruses. Integrity files are named VS.VSN and stored with a hidden attribute. The whole point of the exercise is to try to improve performance by reducing the amount of virus scanning required.

The default set of files to include in an on-access scan are COM, EXE, DO?, XL?, VXD, SCR, with SCR replacing DLL in the default set. File settings can be configured for file creation, file read and file execute. However, the reaction to an infection is fixed and depends on the selection made. File creation's action is to delete the file. With file read and file execution access is denied. There are three groups of settings:

1. Recommended:

   ✔ file create action – delete file
   ✗ file read – access denied
   ✔ file execution – access denied

2. Custom:

   Any combination of the above options is allowed. To change a selection, right click to toggle the choice.

3. Disengage scanner:

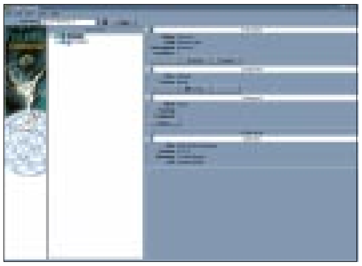
   ✗ file create action – delete file
   ✗ file read – access denied
   ✗ file execution – access denied

Scheduled Scan

The scheduler can be defined to run unscheduled, or scheduled once, every hour, every day, every week, every month. The start time can be configured separately for time, day and date. The scheduler is integrated with the on-demand scanner (unscheduled) and uses the same settings.

Administration

The setup for administration can have a password to prevent unauthorized change. There is an integral virus information list – the test version was dated 12 May 1999. The virus portion of the module is handled in two parts. Server administration is controlled by *eSafe Console*, which also provides monitoring facilities. *eSafe Protect* provides the configuration options for the entire product. The client programs provide support under *Windows 95*/*98*, DOS and *Windows NT*.



Under the environment option which is called from the Administrator program, it is possible to change the SmartScan integrity file name [VS.VSN], the path of the alert filename [C:\ESPLAN\VS95NT.LOG] and copy infected viruses to a quarantine directory [C:\VIRUSES]. There are further options to add a bespoke message for virus notification and the option to sound an alarm.

In addition there is an option to create an 'ignore' list. The default selection includes: VSCHK.COM, CONFIG.DOS, SUHDLOG.DAT, ASMREC.MEM, BLOCK.MEM, VIRUS.MEM, VIRUS.PGM, and WIN386.SWP.

Updates

Updates are provided for the duration of the licence by downloading from a secure *eSafe* site. The *ESP* Gateway can be configured to download and install updates automatically on a regular basis. The update facility is available under the environment section and can be obtained from a downloaded file, a floppy, or a directory on the LAN.

Scanning Overhead

To measure the extra work performed in detecting a virus, a diskette comprising 26 EXE and 17 COM files was scanned. The scan was repeated with the files infected with the Natas.4744 virus. It took 36 minutes and 30 seconds to scan 5,500 clean files, with no false positives. During the Clean set scan there were three instances when the scanner reported that it had skipped some files.

Detection Rates

The *eSafe Protect* scanner was tested against the traditional *Virus Bulletin* test-sets – In the Wild, Standard, Polymorphic, Macro and Boot Sector. All the tests were conducted using the default scanner file extensions supplied. The scan action option was selected in order to delete the infected files and the residual file count used to determine the resulting detection rate.

eSafe Protect successfully detected all of the ItW boot sector viruses, but results were poor against the ItW File set, seven samples of TPVO.3783.A remaining undetected. Three viruses in the Polymorphic test-set caused problems for *ESP*, namely DSCE.Demo, Girafe.TPE and one sample of Natas.4744. Meanwhile, against the Standard test-set samples of Win32/Ska, Win32/Redemption, Argyle, Win95/Boza.D, CMOS.3622, Cruncher, DBF.1046, DNA.1206, Greets.3000, Win95/Navrhar, Pizelun, and TPVO.3783.B were all missed.

Results against the Macro test-set were equally disappointing. Misses were recorded against the full range of *Office* viruses, including the polymorphic X97M/Soldier.A, PP97M/Vic.A, PP97M/Shaper.A, XM/SpellChecker.A, and eleven variants of Laroux to name but a few. Re-running the tests with ‘All files’ selected did little to change the picture, the only difference being that the samples of O97M/Triplicate without extensions (all four variants) were detected this time round.

Real-time Scanning Overhead

To determine the impact of the scanner on the workstation when it is running, the following test was executed. 200 files of 21 MB (a mixture of DOC, DOT, XLS, XLT, XLA, EXE and COM files to reflect typical file types being moved) were copied from one folder to another using XCOPY. The folders used for the source and target were excluded from the virus scan so as to avoid the risk of a file being scanned while waiting to be copied.

The default setting of Maximum Boost for Foreground Application was used throughout for consistency. Timing tests were run with Standard scan rather than SmartScan to avoid the overhead of generating the integrity files.

Due to the different processes which occur within the server, the time tests were run ten times for each setting and an average taken. The tests were as follows:

- Program not loaded: establishes the baseline time for copying the files on the server.
- Program installed, scanning files being created: tests the impact of the application scanning files as they are created on the server.
- Program installed, reading files only: tests the impact of the application scanning files as they are being read.
- Program installed, scanning both creating and reading files: tests the impact of the application scanning files when being created and read.
- Program installed, scanning both creating and reading files *and* running manual scan: tests the additional load of an application accessing files.
- Deselect virus scan, Sandbox, and Firewall.
- Program unloaded: run after the server tests to check how well the server is returned to its former state.

The timing results proved difficult to assess. When the creation and reading tests were run they gave low overhead compared with the baseline. When the virus detection was turned off along with the Sandbox and Firewall options, the speed test was actually faster than the baseline. A similar result was reported in the March Comparative Review. Perhaps the marketing spin should be ‘load *eSafe* but don’t run it and it will improve your server’s performance!!’ This is the first set of tests using *Windows NT v4.0* Service Pack 5 and it appears that the base setting is slower than under Service Pack 3.

Summary

Although *eSafe Protect* comprises a number of facilities for Internet protection, the main emphasis of this review was the anti-virus component of the product. The concern about detection rate at the beginning of the article seems well founded. Although the detection rates were on a par with previous results from the March Comparative Review for this engine, they still appear to be on the low side compared with other scanners.

Although focused on the anti-virus aspect of the product, it is interesting to see this type of product evolving to counter the new threats posed by the Internet. If the detection rates can be improved, then this could be a useful tool for a site administrator, particularly in schools.

eSafe Protect Enterprise for Windows NT

Detection Results

Test-set[1]	Viruses Detected	Score
In the Wild Boot	44/44	100.0%
In the Wild File	519/526	98.7%
Standard	1217/1265	95.9%
Polymorphic	13968/14444	96.7%
Macro	2526/2773	91.1%

Overhead of On-access Scanning:

The tests show the time (in seconds) taken to copy 200 COM and EXE files (21 MB). Each test was repeated ten times, and an average taken.

	Time	Overhead
Not loaded	22.0	–
Loaded, creation	22.2	0.87%
Loaded, reading	22.5	1.97%
Loaded, creation, reading	22.1	0.38%
— + — + manual scan	24.0	8.93%

Technical Details

Product: *eSafe Protect Enterprise.*

Developer: (International) *Aladdin Knowledge Systems Ltd*, 15 Beit Oved Street, PO Box 11141, Tel Aviv 61110, Israel. Tel +972 363 62222, Fax +972 353 75796, (UK) *Aladdin Knowledge Systems UK Ltd*, 1 William Street, Windsor, Berkshire, SL4 1BB, UK. WWW http://www.aks.com/ and http://www.esafe.com/, email esafe.sales@aks.com or info@esafe.co.uk.

Price: $10 to $40 per seat, dependent upon quantities.

Hardware Used: Workstation: *Compaq* Prolinea 590, 80 MB of RAM, 2 GB hard disk, running *NT Server v4.0 (SP5).*

[1] **Virus Test-sets:** Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/Win98/199905/test_sets.html.

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.

END NOTES AND NEWS

***Virus Bulletin's* recently reinstated Letters page** is your chance to air your grievances, seek advice or just make yourself heard on anti-virus-related issues. If you fancy appearing in the pages of *VB*, email us your thoughts; editorial@virusbtn.com.

Claiming 'virus protection is an integral part of the PC and should not have to be purchased separately', ***Computer Associates International* (*CAI*) has staked its claim in the free anti-virus software arena**. The recently released *InoculateIT Personal Edition* (*InoculateIT PE*) for *Windows 95*/*98* and *NT* workstation – a re-badged version of *Vet Anti-Virus* – is the product on offer. Aimed at the single user, the readme file recommends those looking to 'install anti-virus protection to many computers in a business… evaluate other anti-virus products in the InoculateIT range'. Updates are provided via a Live Update applet and the web. Registered users can request technical support by email, but a disclaimer on the web-form offering this claims such support, initially, is only available to North American customers. Apart from the *CAI* splash screens, *InoculateIT PE* should be familiar to users of *Vet Anti-Virus*. This is the first free, fully-featured scanner offered for *NT* and *Windows 9x* with unlimited updates, on-demand and on-access components plus technical support – its impact on the home user/small office market will be watched keenly. Details can be obtained at the web site http://www.cai.com/antivirus/personal/.

At this year's **Secure Computing Award ceremony** in London *Network Associates* received the Academy Award for Best Anti-virus Solution for *VirusScan v4.0* and the Special Award for Best Security Software for *PGP v6.0*. *Symantec* won the Reader's Trust Award for Best Anti-virus Product for *Norton AntiVirus v5.0*.

Information Systems Auditor is a new UK monthly publication which covers all aspects of computer security including Y2K, risk assessment, legality issues, fraud and viruses. A limited free trial offer is available; Tel +44 1993 824130 or email sales@intnews.com.

Registrations are now being taken for **VB'99, to be held in Vancouver, Canada from 30 September–1 October.** For your copy of the conference brochure, or registration information and a delegate pack, contact Jo Peck at *VB*; Tel +44 1235 555139, fax +44 1235 531889 or email joanne.peck@virusbtn.com. Further details are on the *Virus Bulletin* web site at http://www.virusbtn.com/.

Workshops and courses run by *Sophos* in July 1999 include: Advanced Internet Security on 6 July, Implementing *Windows NT* Security on 7–8 July and Practical Anti-Virus on 14–15 July. All the sessions will take place at the organization's training suite in Abingdon, UK. For details on late bookings for June or to reserve your place in July, contact Karen Richardson; Tel +44 1235 544015, fax +44 1235 559935, or visit http://www.sophos.com/.

Central Command and *Kaspersky Lab* annnounce the release of the first integrated anti-virus protection for *Microsoft Office 2000*. ***AntiViral Toolkit Pro for Microsoft Office 2000*** offers protection against viruses in *Word*, *Excel*, *Access* and *PowerPoint* as well as via the Internet. A special edition of this product may be downloaded for evaluation from http://www.avp.com. For more information contact Renee Barnhardt; Tel +1 330 7232062 or email renee@avp.com.

Reflex Magnetics has launched ***Reflex Disknet Data Security for Windows NT v1.7***, which includes a Program Security Guard (PSG), a powerful software module which claims to protect against viruses like Melissa. Additional new features include the Reflex Macro Interceptor and Boot sector protection. It is available now at £249 +VAT for the Administrator and £28 +VAT for the Client version. For details contact Phillip Benge; Tel +44 171 3726666,fax +44 171 3722507 or email benge@enterprise.net.

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.

The call for papers for the ninth annual *EICAR* conference has gone out. The deadline for submission is 1 July 1999. **The First European Anti-Malware Conference takes place in Brussels, Belgium from 4–7 March 2000.** A broad range of topics for papers include malicious code, unwanted side-effects or malfunction, network security, the information age and society, cryptography, privacy and anonymity, new media and e-commerce. For more information visit the web site http://www.eicar.dk/.