ISSN 0956-9979 AUGUST 2000

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

Editor: **Francesca Thorneloe**

Technical Consultant: **Matt Ham**

Technical Editor: **Jakub Kaminski**

Consulting Editors:

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

IN THIS ISSUE:

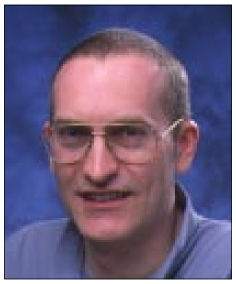
• **Pack-a-Mac:** In the coming months *VB* will be initiating Comparative Reviews for Macintosh products. Full testing protocols, proposed schedules and submission criteria are outlined on p.22.

• **The numbers game:** Denis Zenkin is a self-confessed virus prevalence table junkie. He shares his findings over the past few years in his Feature on p.10.

• **Happy new network?** Millennium fever is well and truly over but what is the reality about moving your corporate network over to *Win2K*? Péter Ször investigates on p.8.

CONTENTS

COMMENT



“ AV companies need more initiatives to promote closer working practices ”

Divided We Fall

The anti-virus industry has a history of good initiatives and co-operation between companies. Two prime examples have been the *CARO* (*Computer Anti-virus Research Organization*) schemes by which viruses can be given a standard name, which reduces customer confusion; and also, the sending of samples to other AV companies, which helps to keep everyone up to date.

However, like Joe Wells (see *VB*, June 2000, p.2), I too have found the industry lacking, but from the perspective of looking forward in order to meet the emerging demands and pressures of a truly global economy where everything happens yesterday. What we are seeing now is an explosion of growth whereby when something happens on one side of the world, it almost immediately spreads to the other side of the world. Are the existing schemes good enough to cope with the speed of modern-day AV warfare?

A recent initiative by the *WLO* (*WildList Organization*) to help counter the problems caused by the speed of mass-mailing viruses has caused uproar amongst some people within the AV industry. REVS (Rapid Exchange of Virus Samples) is designed to get urgent samples of rapidly spreading viruses to the virus analysts in each of the member AV companies, in a secure and controlled way. During the recent LoveLetter outbreak it more than came into its own. We all saw the variants as each member company saw them and we could upgrade our detection as it became necessary. What could possibly be wrong with implementing such a system?

Well, some people think that the existing structures for sending urgent samples are good enough. However, those structures rely on individuals being at their desks day and night in order both to send and receive these urgent samples to and from other companies. Heaven forbid that people should want to sleep or go on holiday!

Another response is that we have *CARO*, that body of senior members of the AV community, always on hand to send and receive samples, who never take a break, and of course every AV company has at least one member. What a load of rubbish. Very few AV companies have *CARO* members. Some companies have *CARO* members who are no longer part of day-to-day analysis and virus detection. I am sure there are plenty of AV companies out there serving their own country or region who have probably neither heard of *CARO* nor have much chance of getting into *CARO* and receiving any urgent samples. What happens when all those *CARO* members meet at the *VB* conference in Florida in September? Do virus writers take a holiday because their ‘adversaries’ are away? Of course they don’t.

Now, I believe these structures are good and have served the industry well but they are not particularly helping us to protect customers from the latest and fastest-spreading virus. Let’s work with the AV company that only serves Ecuador and has no desire to expand past those borders. Let’s take their rapidly spreading samples, which will probably be in your country by the time you have read this article, and protect our customers from the virus – and when we see a possible major outbreak in the making, we send them copies of it so that they can protect their customers.

For many companies it seems that shareholders must take precedence over customers, so that means making maximum opportunity of any situation to promote their own product. Fair enough, a profit needs to be made, but at the expense of not protecting customers? I don’t think so.

If I were a customer looking for an AV product, I would be asking the supplier to be part of any initiative which allows him to get virus samples speedily and securely. Customers want their desktops protected now! I believe that AV companies need more initiatives to promote closer working practices and that the many excellent structures that exist need to be improved to cope with the greater needs of our new e-society. So, ask your AV vendor: are you a member of REVS?

Stuart Taylor, Sophos Plc, UK

NEWS

What a CAD!

Source code for the first *AutoCAD* VBA macro virus was posted on a public Web site in mid-July. Its writer wishes it to be known as Star, but the AV research community had not decided on a name at the time of writing. The appearance of such a virus is not an unexpected development, as *Autodesk* was an early VBA licensee. Due to the timing of *Microsoft's* purchase of *Visio* relative to the release of V5M/Radiant (see *VB*, March 2000, p.6), this is truly the first VBA virus for a non-*Microsoft* application.

The code uses a common VBA class object infection mechanism – locating an object holding its code, copying that code to a string, then injecting it into an uninfected VBA module via the InsertLines method. One of the situations in which it replicates recursively is when a drawing with an infected embedded project is closed while another drawing with an embedded project is open. Bugs in the code mean that several other conditions must be met, each reducing the chances of the virus being a real threat – *AutoCAD* users need not be unduly worried. They should, however, ensure *AutoCAD's* macro warning option is enabled, watch for their scanner adding detection of this virus and be aware this may require them to add several file extensions to their scanner's 'files to scan' list ▮

Mac's Back

The Mac Virus Web site, highly regarded by Mac news and technical information resources as a source of virus news and reference material, was sadly missed when it closed down in September 1999. Now David Harley, the reference copy of whose 'Viruses and the Macintosh' FAQ was hosted there, has taken over the hosting and maintenance of the site at http://www.sherpasoft.org.uk/MacVirus (although macvirus.com still works). All enquiries should be made to harley@macvirus.org.uk ▮

Surveying the Damage

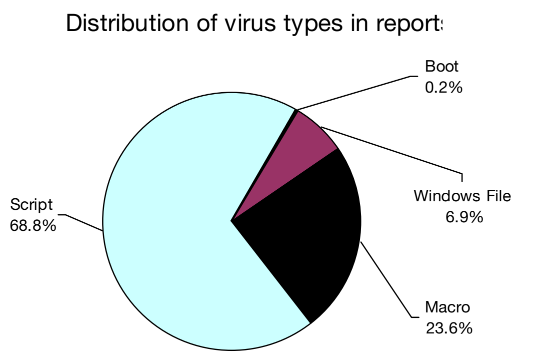
Yui Kee Computing Ltd interviewed 283 home users at the Computer 2000 Exhibition from 4–7 May 2000 at the same time as conducting a corporate survey which questioned data security staff at 125 Hong Kong companies, ranging from those with fewer than 10 PCs to those with over 1000. Both corporates and home users ranked viruses as the highest security risk. 62% of the corporates questioned reported having experienced a virus attack in the last year compared to 12% of their domestic counterparts. This marks a change in the traditional perceptions of staff bringing in viruses from their home PCs. Not surprisingly, LoveLetter was the culprit named by most of the business users, while home users still considered CIH still to be their biggest problem ▮

Prevalence Table – June 2000

Virus	Type	Incidents	Reports
Stages	Script	1095	51.1%
Kak	Script	277	12.9%
Marker	Macro	156	7.3%
Laroux	Macro	90	4.2%
LoveLetter	Script	88	4.1%
Win32/Ska	File	82	3.8%
Class	Macro	42	2.0%
Win32/Pretty	File	39	1.8%
Ethan	Macro	38	1.8%
Thus	Macro	36	1.7%
Melissa	Macro	24	1.1%
Yawn	Macro	15	0.7%
Netlog	Script	13	0.6%
Win32/Fix	File	13	0.6%
Tristate	Macro	12	0.6%
Divi	Macro	11	0.5%
Prilissa	Macro	9	0.4%
Walker	Macro	9	0.4%
Eight941	Macro	8	0.4%
Myna	Macro	8	0.4%
Smack	Macro	8	0.4%
Win32/Funlove	File	6	0.3%
Total		2144	100%

[1] The Prevalence Table includes a total of 65 reports across 27 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

In order to avoid a distortion of the figures, data for the 'self-reporting' W97M/ColdApe virus (totalling 682 reports in June) have been omitted from the table this month.



LETTERS

Dear Virus Bulletin

Legal Precedent

Whilst I found Mich Kabay's Comment on the subject of viruses as speech in the July 2000 edition interesting, it is nevertheless at odds with the real world in several ways.

To recap briefly, Mich suggests that virus source code should not be considered a form of speech. This view, if it were to be widely held, would have considerable ramifications here in the US – the First Amendment guarantees Americans their right to free speech, and must surely be both one of the most jealously guarded, and one of the most widely misused articles of government anywhere in the world. Any possibility of infringing the First Amendment sends shivers down the metaphorical spines of US legislative bodies and sends the legal process into paroxysms of confusion.

There is already a legal precedent for regarding source code as subject to the protections offered by the First Amendment. On 4 April 2000, the US Circuit Appeals Court ruled, in the case of Junger vs Daley, that the plaintiff (Junger) could post his cryptographic source code to the Internet. I quote from the ruling: '… computer source code, though unintelligible to many, is the preferred method of communication amongst computer programmers. Because computer source code is an expressive means for the exchange of information and ideas about computer programming, we hold that it is protected by the First Amendment.'

The ruling has far-reaching implications, and will no doubt be the cause of much future legal disagreement, thus guaranteeing employment for lawyers for years to come. Nonetheless, unless there is some legal difference between virus source code and crypto source code, the current legal precedent in the US appears clearly to disagree with Mich's opinion. Whilst I have recently moved, and now live in the United States, I am not an American. Personally, I cannot decide whether or not virus source code should be regarded as a form of speech; this letter is merely a statement of the current realities, not my personal opinion. In addition, I am not a lawyer …

Ian Whalley
IBM TJ Watson Research Centre
USA

Pleading the First

While it's interesting that Mich doesn't think computer programs should be considered 'speech', the fact is that in the United States – from where he posts his Commentary – they sure seem to be! I direct readers to various academic and court publications referencing Bernstein vs United States Department of State. However, this is not the end of the world!

The real question would seem to be not 'are computer programs speech?' (it's already been established that they are), but rather 'are all *forms* of programs *protected* speech?'. After all, not all speech is constitutionally protected speech – as many court cases throughout history have clearly demonstrated – and whether or not certain *types* of program (i.e. viruses) will eventually be found to be 'protected' or 'non-protected' speech is another matter!

Interestingly, the findings in the United States vs Freeman case seem to demonstrate that a program which is instrumental in and intertwined with the performance of a criminal activity does *not* retain First Amendment protection. Is infecting the computers of unsuspecting users a criminal activity? You bet it can be! So, the Freeman example would certainly seem to be applicable as well in the case of some specific viruses. I look forward to the first prosecution that draws upon this established precedent.

The reader should note, however, that these findings apply to a *specific* program, not a specific *functionality* of program. It has been well established by legal precedent that functionality of speech (including computer programs) does not determine the amount of 'protection' it is afforded. Indeed, the functionality may *be* the expression.

The code in the book Mich mentioned, like the 'instructions' of other wrong things in many books published worldwide, could not jump off the page and type itself in and execute itself just as other 'instructions' given in books cannot do.

Just as guides for abortion, euthanasia, suicide, adultery, revenge and more have been admitted to the realm of First Amendment protection, it's likely that the publication of self-replicating code *in general* will be admitted to this realm if push comes to shove. If not, and it becomes illegal, I don't expect it will make a lot of difference in the long run anyway, other than to drive it all back underground where it all started.

Now, what is done with the published information is a whole other matter – and that is really where the legal focus needs to be! (The moral focus is another matter again, and is actually the one where the most change in all of these areas is likely to be affected.)

Finally, Mich asks why should we accept an excessively broad definition of speech that includes self-replicating code that does xyz. Well, apart from the fact that it's already been established that computer programs are speech, and that functionality does not determine the amount of First Amendment protection, it's really very simple: the right to speech, even speech we don't like, is very precious. Our concern is that if we can justify the suppression of information as 'undesirable', or 'potentially dangerous', is it that much further a jump to the suppression of other information some others may not like?

The bottom line is I don't know *anyone* who believes it is wrong to punish the authors of self-replicating computer programs 'when and if those programs cause criminal or civil damages'. Punishing those responsible for a criminal or civil wrong is to be highly sought after. Punishing someone who plays a part in the wrong is also sought after.

This may or may not include punishing a person who writes a self-replicating computer program – but to suggest making it illegal to write a self-replicating program is capricious at best.

Sarah Gordon
Independent Consultant: http://www.badguys.org
USA

The Prosecution Rests

In July's issue of *VB*, the opinions of Michel Kabay were stated as being 'controversial' – this is a point I think needs to be clarified. I don't believe that his opinions are anything but fair and well-founded.

At this point, I should indicate that these are my personal thoughts on this, and not those of my employers.

Consider this: in most countries it is illegal for a member of the public to make a bomb – it is tantamount to terrorism. Why is this illegal? Simply because a bomb placed correctly can destroy much property which is very expensive to repair (it is also possible for a bomb to take human life, but this is not always the reason behind planting a bomb).

If we consider a virus such as LoveLetter, we also find that its effects came with a high dollar value both in system downtime and also man-hours taken to clean up after it. So where is the difference? The answer that a virus writer is likely to give is that a virus is, in essence, far different from a bomb and can't claim life – it is seen as their right to create viruses without hindrance from the law.

In truth a virus, by definition, causes unwanted side-effects on computer systems that cost money to sort out. To take it a step further, most viruses are written so as to hide themselves from unsuspecting computer users. By taking the points that a virus is designed to avoid detection and also to cause unwanted damage, it is clear that there is no legitimate reason for creating one.

Surely if a large company is hit by a virus and incurs large costs cleaning up after it, the organisation should have some recourse to the person who created it? After all, if Big Ben were to be destroyed by a bomb, I'm pretty certain the person who made the bomb would be brought to justice.

As for freedom of speech – is it infringement of people's civil liberties to be stopped from doing something that prevents millions of other people carrying out their jobs? The answer to that, I think, is a definite no.

John Bloodworth
NAI
UK

After the Love Has Gone

The life of the notorious VBS/LoveLetter worm can be divided into two stages: in the first incubation stage, it is unknown to virus scanners; in the second stage, when the updates come out, the virus dies out. It roughly multiplies its number in each life cycle.

AV companies believe that the best solution is to protect more PCs with virus scanners. Although this is a factor, there are others too. The LoveLetter lifecycle is approximately the frequency with which an average user reads their email. This cannot be reduced – users are not likely to read mail less frequently. The multiplication factor is the difference between the worm copies sent out from an infected PC and the number of 'absorbed' worm copies, which do not cause further infections.

The number of worm copies sent out can be decreased by introducing blocks in the mass-mailing capabilities of *Outlook*, exactly what *Microsoft's* questionable new security patch is doing.

Absorption comes from several sources. The worm is absorbed if the mail arrives on a PC that is not running an email client capable of activating it, or if there is virus protection that detects it. We cannot do much about the former – *Outlook* will remain the dominant email client.

Users could accept the rule of not running any attachments before thorough checking, but I am afraid that is not going to happen either. The latter component could be increased if virus scanners were able to block *new* viruses. There is a whole lot to do in this area.

If the virus multiplication factor is higher than 1, the virus will spread exponentially. Given typical users with typical address book sizes, this multiplication rate could be well over 10. Nuclear chain reactions are driven by the same equations as virus propagation. Fourteen years ago in an incident in Chernobyl, the multiplication factor was slightly above 1.2. You may have heard about its outcome.

The total number of viruses released has to be decreased by reducing the reaction time of the AV companies. With automated analysis and signature extraction this could be reduced to about 1 hour but there is not much hope of decreasing it further.

Gabor Szappanos
Computer and Automation Research Institute
Hungary

VIRUS ANALYSIS

What's New, PussyCats?

Costin Raiu
GeCAD Srl, Romania

'Multi-platform' is no longer a rare term when it comes to the description of new macro viruses. This technique started quite early in the history of modern macro viruses, with implementations able to jump between *Word* and *Excel*, and later *PowerPoint*. *Microsoft Project 98* was added to the list of vulnerable OSes next – P98M/Corner was the first to infect this classic platform which was already old in the winter of 1999.

Rumours of the first *Visio* macro virus started in early 2000, and were somewhat realized in the virus named 'Radiant', initially received by the AV community in source code form. Its actual existence in native form was initially denied because anti-virus researchers did not want to create the binary from source, given the ethics related to the creation of new viruses. Soon afterwards, a 'real' *Visio* virus was received in the form of the V5M/Unstable macro virus. A couple of days later, a binary of V5M/Radiant put in an appearance as well.

The remaining step was for someone to link all these new and old techniques in a massive multi-platform macro virus able to infect *Word*, *Excel*, *PowerPoint*, *Project 98*, *Visio* and maybe even more (anyone dare to think of *Access*?). Well, the virus authors have made one more step in this direction with {W97M,X97M,P98M,V9M}/Cats.A, as named by *CARO*.

The Virus

One of the reasons I mentioned that V5M/Radiant was initially received in the form of its source code, and that its binary document form was not created in the AV labs, is because Cats reached us in a similar manner. Many macro viruses use VBA methods to save their source to a text file on disk, then import it into a clean document in order to infect it. However, even if the Cats 'loader' or 'constructor' works in much the same way, we still cannot ignore the fact that, even with a helper script, this would be the same as if we were to copy and paste the virus source into a clean document in order to obtain the binary form. That is why I will attempt to describe this virus and still refrain from creating a binary copy. Computer anti-virus researchers are often accused of creating new viruses, so minimizing this possibility is an important part of our ethics.

Operation

The Cats virus from my collection resides in the form of a VBScript loader file which, in the sample set I received, is named MSOFFICE.EXE.VBS. The actual virus source which is *supposed* to be named EXPO.EXP is located in the root of the C: drive. Well, supposing that someone follows this path and drops EXPO.EXP to C:\, and then runs MSOFFICE.EXE.VBS – the virus constructor is executed. This small script, no longer than 455 bytes, will instantiate a COM object of the Word.Application type, wipe everything in its Normal template ThisDocument module, then use the VBA .AddFromFile method to insert the code from the C:\EXPO.EXP source file in the very same module.

From then on, the 'resident' virus will wait for someone to run *Word*, and execute the virus from the Normal template. There, the virus hooks the Document_Close event – thus, the first time a document is opened, then closed, the virus code is launched. Cats will start its replication code from here. In order to prevent possible errors from stopping the replication process, the virus will first take care to activate error handling and disable all the potential alerts displayed when a macro encounters a problem while running.

Then, it will perform a couple of checks in order to be sure the appropriate piece of code is run for each platform. In case it is run from the *Word* Normal template, our main case, the virus will enable the File/SendTo command to send *Word* documents as emails, not as document text which will, of course, lose its viral macros.

Other macro viruses, for example P98M/Corner, also do this to increase the odds of travelling and possibly infecting other systems via email. Actually, a lot of Cats' code looks similar to Corner's routines – either the Cats author wrote Corner, or at least he used it as a source of inspiration.

After that, the virus will simply check if the active document is already infected, and if not, it will copy itself into the ThisDocument class module of the above-mentioned active document. The copy process is performed using the standard VBA .InsertLines method.

We should also note that in all cases (well, actually not all of them, as you will see below) where the virus attempts to copy itself somewhere, it takes care to delete any previous macros from the target module. This will prevent problems caused by multiple infections or interactions with other viruses using the Document_Close (or Document_Open, or the respective *Excel*, *Project* or *Visio* routines) message handlers. Well, at least we do not have to worry about Cats 'sandwiches', except for the cases when Cats was the first to infect the module.

Like most viruses, Cats contains a self-check to prevent multiple infections – this is performed by checking if the second line of code is an apostrophe-style comment ('). However, as you will see below, the self-check fails when used in *MS Project*; on the other hand, it probably works fine for *Word*, *Excel* and *Visio*.

Following the virus source, the next step is to infect *Excel*. Cats creates a standard Excel.Application object, then a new, empty workbook. It infects the ThisWorkbook module with virus code, and saves it with the name Book1 in the *Excel* startup path. Due to the way *Excel* works, as soon as the *Excel* infection is finished, the *Excel* instance of the virus will be called. In *Excel*, the virus subroutine is called Workbook_Deactivate. The *Excel* image of the virus will skip the *Word* infection, the *Excel* (itself) infection, and will hit the *Visio* and *Project* parts.

The *Project* infection routine will only work if *Project* is already running. If this is indeed the case, Cats will copy itself into the *Project* global template and patch itself a little bit in order to become compliant with the *Project* VBA and *Project* VBA objects. A bug in the infection routine causes the virus to insert itself into the *Project* global template each time the *Project* infection routine is run, thus the virus is intended in its *Project* form. I can only suppose the author did not bother to verify the *Project* infection, or did not check it too thoroughly.

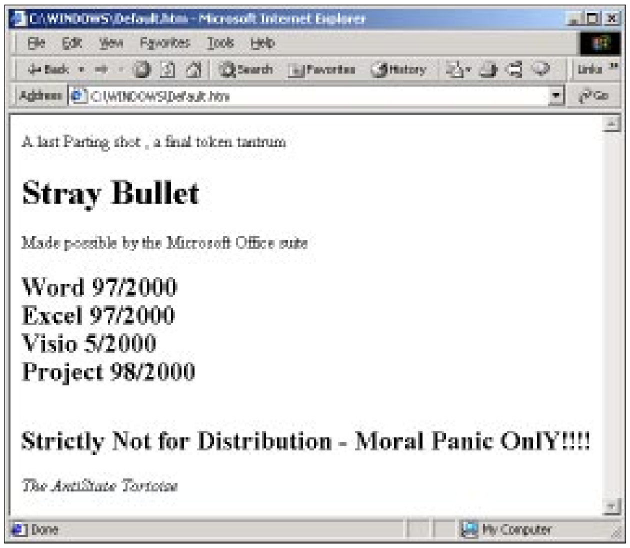
For *Visio*, the cross-platform jump works even if *Visio* is not currently running on the system – the virus creates a new object of the Visio.Application type, and copies itself into the ThisDocument module of the newly created *Visio* drawing. The drawing is eventually saved to the fixed path C:\PROGRAM FILES\VISIO\SOLUTIONS\BLANK DRAWING.VST which is supposed to be the base, standard *Visio* template. Let us note that Cats might have been intended for *Visio* as well; however, the virus creates a new *Visio* presentation which has no macros, and infects it. Since for *Project* the virus targets the global template, the missing 'cleanup' code prevents it from being a true, fully functional *Excel*, *Word*, *Visio* and *Project* infector.

It is interesting that this virus implements some sort of safety belt, obviously designed in order to resuscitate itself if, for some reason, it is deleted from the *Word* global template. In order to accomplish that, a small script, identical to the one we originally received (i.e. the virus 'loader'/'constructor') will be dropped to the fixed path C:\WINDOWS\START MENU\PROGRAMS\STARTUP \MSOFFICE.EXE.VBS where it will be run from each time *Windows* is started.

The associated (and mandatory!) EXPO.EXP file is dropped by the virus during the *Excel* cycle. This only happens if the virus is run from an infected workbook, and the global *Excel* template Book1 in the *Excel* startup directory is not yet present. We should note that the dropped EXPO.EXP file will probably contain a copy of the virus with the Document_Open handler, and not the usual (for the Normal template) Document_Close.

This is unlikely to cause any problems to the virus' replication cycle, but probably the author did not notice this small bug either, since infection of documents from the Normal template on Document_Open has more chance of being noticed than infection on close.

Next, the virus runs a payload subroutine which replaces the usual C:\Windows\DEFAULT.HTM file with one of its own, which looks like this:


C:\WINDOWS\Default.htm - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\WINDOWS\Default.htm
Done
My Computer

A last Parting shot , a final token tantrum

Stray Bullet

Made possible by the Microsoft Office suite

Word 97/2000
Excel 97/2000
Visio 5/2000
Project 98/2000

Strictly Not for Distribution - Moral Panic OnlY!!!!

The AntiState Tortoise

Conclusion

The Cats virus takes an interesting step down the macro virus development road. When I first saw O97M/Tristate, I anticipated that even if *PowerPoint* were not the main replication vector, it would still play an important strike-back role in the virus' life. Without disinfecting the virus in all the infected objects, the chances for it to reappear will still exist.

Indeed, for a while, users had problems because not all scanners were able to disinfect O97M/Tristate in infected *PowerPoint* presentations. With upcoming viruses able to infect even more platforms, 'complete' disinfection gets more complex. If we detect viruses in *Excel*, *Word*, *Visio* and *Project*, but we forget about the scripts designed to reload them, or we do not disinfect viruses in, let us say, *PowerPoint* presentations, users will again have to live through the same sad Tristate story.

{W97M,X97M,P98M,V9M}/Cats.A

Aliases:	Corner.B.
Type:	Multi-platform macro virus.
Payload:	See picture. Drops a custom file called C:\WINDOWS\DEFAULT.HTM.
Removal:	Use an updated anti-virus product able to handle it, and do not forget about the other files, namely C:\EXPO.EXP and C:\WINDOWS\START MENU\ PROGRAMS\STARTUP\MSOFFICE .EXE.VBS.

TECHNICAL FEATURE

Moving to Windows 2000

Péter Ször
SARC, USA

At the 1999 *Virus Bulletin* conference, Darren Kessner explained some of the new features of *Windows 2000* that could potentially be used for malicious purposes by virus writers. Some of the new features, such as the *Intellimirror* or *Microsoft Installer*, are still waiting to be discovered by virus writers.

Although the new features make it easy to create new virus types and spreading mechanisms, some of the old 32-bit *Windows* viruses failed to work on the release version of *Windows 2000*. Many US corporations are planning to upgrade to *Windows 2000* during September. Moving to *Windows 2000* does make for a few clear advantages. This is, of course, a very costly procedure, and most corporations would do well to wait carefully until the first Service Packs are available.

Windows 95 viruses

Most *Windows 95* viruses will fail to work on *Windows 2000* altogether. About 50% of all 32-bit *Windows* viruses are classified as '*Win95*', meaning that they only work properly on *Windows 95*/*98*. Most of the *Win95* viruses that have the potential to spread were developed with the use of VxD functions. Since the VxD driver model is not supported under *Windows 2000* (or under *Windows NT*), a whopping 50% of all 32-bit *Windows* viruses will not affect someone's *Windows NT*/*2000* system. A workstation upgrade from *Win9x* to *Win2K* would mean the end of such viruses as CIH.

Win32 viruses

Half of all 32-bit *Windows* viruses are actually classified as 'Win32'. These viruses are able to replicate under at least two major Win32 systems. *Windows 2000* is an *NT*-based system with a number of major enhancements. Thus, someone might believe that all viruses that worked under *Windows NT* would still work under *Windows 2000*.

Another common misconception is that *Windows 2000* is so special that virus writers need to write completely new viruses to support it. There were announcements from several anti-virus vendors related to the W2K/Installer virus. The virus was quickly labelled as 'the only virus able to work under *Windows 2000*'.

This is why it was interesting to see if there were any Win32 viruses that did not work on the new *Windows 2000* release version. It turns out that a significant 25% of all Win32 viruses (half of all 32-bit *Windows* viruses) cannot work on the release version of *Windows 2000*. This is due to minor incompatibility problems that appear in those viruses which were written in Assembly language.

KERNEL32.DLL Base Address

Several Win32 viruses search in the process address space at various locations for the loaded KERNEL32.DLL. Many Win32 viruses do not search the complete process address space but check the 'MZ', 'PE' sequence at the known KERNEL32.DLL base addresses.

The common location of the KERNEL32.DLL is at 0x77F0000 under *Windows NT*. *Windows 95* and *Windows 98* use a higher address since the DLL needs to be loaded into the shared memory space 0xBFF70000 for both versions. Some of the viruses concerned check for the loaded KERNEL32.DLL at that address in order to identify the addresses of all APIs they need to call.

Early *Windows 2000* betas (*Windows NT 5.0* at the time) used significantly different KERNEL32.DLL base addresses for almost every release. For instance, beta 1 used the address 0x77EF0000; that was changed to 0x77ED0000 in beta 3. The RC2 version used 0x77E80000. Not surprisingly, many viruses only work on the RC2 version of *Windows 2000* and fail to work on the release version. This is because in the final version the address was specified as 0x77E00000.

Viruses that check for the loaded DLL at one wrong location will fail to work. For example, W32/Cabanas does not pay attention to the moving DLL base address and fails to use that method. However, Cabanas uses more than one method to get the API addresses. If the actual host application has an import to the APIs GetModuleHandle() and GetProcAddress(), the virus can replicate to other files.

Viruses that use only one (wrong) method will fail. For instance, the W32/FunLove virus fails under the release version of *Win2K* although the virus writer paid attention to the new base address. The way W32/FunLove checks for the location of 'GetProcAddress()' is based on a string detection. Since the code changes, the virus is unable to locate the GetProcecAddress() API. FunLove also fails to use the W32/Bolzano trick of patching the NTOSKRN.EXE file because of the SFC (explained later). Almost 25% of all Win32 viruses fail to replicate because of similar problems. Obviously viruses that are created after the release version of *Win2K* are likely to work properly again.

System File Checker

Win2K introduced the System File Checker (SFC) feature. The SFC uses two directories under the WINNT folder, 'Driver Cache\I386' and 'SYSTEM32\DLLCACHE'. The

Driver Cache\I386 directory contains a CAB file called DRIVER.CAB. This file comprises an archive of all the *Microsoft* drivers for *Windows 2000* as well as other crucial system components.

The DLLCACHE directory contains other DLLs and executables such as NOTEPAD.EXE or CALC.EXE. The directory might not necessarily mirror all the applications. A limit is specified according to the drive's free disk space during basic installation of *Windows 2000*. For a large disk, however, the DLLCACHE mirrors almost every standard application and DLL.

The WINLOGON process is always loaded on *Windows NT*/*2000*. WINLOGON was selected to contain the SFC extension. When WINLOGON starts, it registers a directory 'change notification request' callback function of its own to the system, so that whenever the contents of certain directories change, WINLOGON receives notification. Then WINLOGON looks for the change.

It seems the SFC of the *Win2K* release version uses a catalogue of cryptographic signatures of all system files. In the case of a hash difference, the SFC will use either the DLLCACHE or the DRIVER.CAB file to replace the modified file silently.

WINLOGON will only generate message boxes if an original copy of the file is not available under the SFC directories. If this is the case, a message box pops up asking for a CD that contains the installed *Windows 2000* version. The changed files are copied from the CD and overwrite the replaced files automatically. The SFC compares the file contents in other ways. The version information seems to be a key element.

Closely related system component versions might be replaceable by presenting a new copy of the driver or executable in the SFC directory first. A newer version can overwrite the existing copy. Clearly, the SFC's primary purpose is not protection against viruses. However, it can be used even from the command line and it generates 'information logs' viewable with the Event Viewer.

Some viruses will fail to work completely because of *Windows 2000's* SFC feature. For instance, the W32/Kriz virus tries to create an infected copy of KERNEL32.DLL first in the SYSTEM32 directory. During boot time, that newly created file would replace the old one. However, the SFC will not let the modification remain. When the machine boots, the W32/Kriz-infected DLL file is gone and KERNEL32.DLL is not replaced. This is because the SFC pays special attention to kernel components and checks their integrity prior to anything else.

Actually, the SFC *does* let the modification happen. It does not try to prevent the modification in any way. Instead, after the modification occurs, it tries to replace the modified file with the old copy. This is a special way of protecting against installation software that replaces certain DLLs or EXE or SYS files with a different, incompatible version.

This problem is known as 'DLL hell'. The SFC was not developed against viruses, and virus writers might find ways to fool it via different methods. However, the SFC has a certain benefit against the majority of Win32 viruses that modify files in the WINNT folder whenever they have the necessary rights to do so. If the SFC-related components are protected with standard system security such as file protection against writes, only Kernel-mode *Windows 2000* viruses can challenge the SFC.

It is difficult to notice this automatic backup system. When a Win32 virus such as W32/MIX is executed under *Windows 2000*, a very noticeable disk activity will follow the execution of the virus. This is because the virus infects new executables and the SFC starts to get the modification notices and tries to replace the files with original copies. As long as all the copies of the original EXEs are available, *Windows 2000* will be able to work silently without displaying a single warning about the fact that the SFC was used (zero user administration).

If the task list is checked during this time, WINLOGON shows very high processor usage. This is due to the fact that WINLOGON copies the files from the mirror directories one after other. This means copying megabytes of data onto the disk. It is noticeable, although it would be better to have an optional warning message, say after the first ten modified executables were changed!

It is strongly recommended that System Administrators look into the Event logs very frequently. Personally, I believe that it would be better to put this feature under the 'Security Log' instead. The standard 'Information' message might not appear to be important but it could be a sign of virus infection. Since non-standard applications are not involved with the SFC, those executables that do not have cryptographic signatures are not protected in any way.

Some of the newer viruses around such as W2K/Installer, W32/Dengue and W32/CTX do not infect the executables if they are found to be SFC-protected. This is because virus writers can use the SfcIsFileProtected() API exported by SFC.DLL to check if a particular file is protected and thus avoid infecting it.

The SFC does not stop the spread of certain viruses and it is not a virus security feature. However, it makes the spread of regular PE file viruses less trivial. Virus writers will certainly try to switch off this feature of the system one way or another. In any case, it is strongly recommended to use the SFC regularly.

Conclusion

Real HLL-written Win32 applications will work perfectly well on *Windows 2000* without any major problems. Thus, *Win2K* will not prevent the W32/ExploreZip worm kind of episode. Moreover, *Win2K* supports VBS by default and therefore some of the newer viruses might 'appreciate' the *Win2K* environment more than *Windows 95* or *Windows NT*.

FEATURE

The Number of the Beasts

Denis Zenkin
Kaspersky Lab, Russia

As a long-time, regular reader of *Virus Bulletin* I always peruse all the articles and reviews published in the magazine carefully. One of the most interesting parts is the monthly published virus prevalence table. I am very happy to have such statistics. The Prevalence Table is very useful, giving users a snapshot of what is really going on in the virus world. Many people are too lazy (sorry, I should say busy) to look into Joe Wells' WildList to see the most dangerous threats. If they did, they would not see the prevalence of different viruses. On the other hand, *VB's* Prevalence Table is a must for everyone who deals with computers, no matter whether they are a network administrator or a home user.

As an avid *VB* Prevalence Table fan, I decided to see what is behind them and the results are extremely interesting. A summary of all the virus statistics published in *Virus Bulletin* reveals the top ten viruses in history, the top viruses by type and the viruses of the year. This enables one to draw a general chart for the number of infections since 1995, compile an aggregate virus prevalence table for the whole period and prepare a pie-chart of the most prevalent virus types. Finally, we can take a look at the most dangerous systems for computing as regards the number of viruses that could affect them.

Top Ten Viruses in History

W97M/ColdApe is a very sensitive case. As you have probably noticed, the data for this virus was omitted from the Prevalence Table in the February issue and ColdApe was rated as self-reporting.

	Name	Type	No. of Incidents	Percentage
1	ColdApe	Macro	8856	15.3%
2	Cap	Macro	3893	6.7%
3	Class	Macro	3847	6.6%
4	Ethan	Macro	3512	6.1%
5	Win32/Ska	File	3462	6.0%
6	Laroux	Macro	2548	4.4%
7	Marker	Macro	2423	4.2%
8	Win95/CIH	File	2172	3.8%
9	Concept	Macro	2007	3.5%
10	Form	Boot	1517	2.6%

Before this, all ColdApe reports were counted as 'true'. Just imagine, every day an infected computer sent out a message to Nick FitzGerald (the virus's original payload). Each time was counted as a new incident, but actually this is not true. In only one year (1999) it registered 8,622 times, which is unbelievable! This is the reason why it tops our 'Top Ten Viruses' list.

Top Viruses by Virus Types

There is no need to introduce the nominees. All of them are 'well known' due to the great financial loss they caused.

Nomination	Place	Name	Incidents	Percentage
Top macro virus	1	ColdApe	8856	15.3%
Top file virus	8	Win95/CIH	2172	3.8%
Top boot virus	10	Form	1517	2.6%
Top script virus	26	VBS/Kak	660	1.1%
Top worm	5	Win32/Ska	3376	6.0%

Once again, we should be careful when we analyse W97M/ColdApe. If we omit the data for this virus, then the real 'winner' would be the WM/Cap macro virus. With regards to the top script virus, I should mention that things will change. In only one month, VBS/LoveLetter scored 654 incidents, while JS/Kak has been reported 660 times since its discovery in late 1999. Due to a great number of variations, next month we expect LoveLetter to take the lead. It even has the chance to become 'Virus of the Year'.

Viruses of the Year

This table shows how fast things are changing. For many years since the first PC virus was discovered, boot viruses were always the type that spread the most widely.

Year	Name	Incidents (for year)	Percentage (for year)
1995	Form	328	13.3%
1996	Concept	762	15.9%
1997	Cap	694	14.7%
1998	Cap	953	16.8%
1999	ColdApe	8622	25.5%
2000*	Win32/Ska	778	12.1%

After 1995, when the first macro virus – WM/Concept – appeared, it occupied the top position for four years. Only in 2000 (*shown up to May) did worm-style viruses, due to their exceptional mass-mailing abilities, top the list. In 1999, if we omit the 'self-reporting' ColdApe, the top virus would be W97M/Class, with 3216 incidents and 12.6% of the total reports. Nowadays, the Internet has become the main virus propagation source. Thus, to become widespread, the virus requires special worm-style spreading abilities via email, IRC channels and so on. Many more viruses of this type are appearing. For the most part they are still macro, script or file viruses, but they feature new propagation technology.

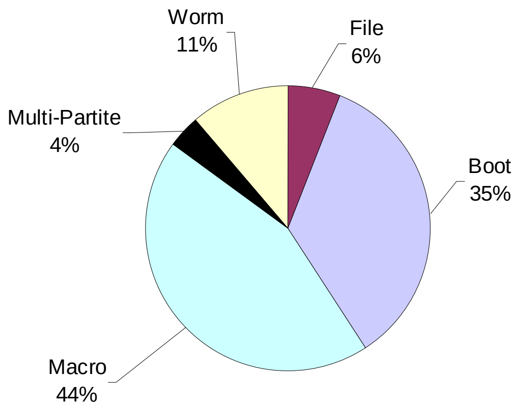


Figure 1: Most prevalent virus types between 1995 and May 2000.

Despite numerous rivals, macro viruses are still number one in the world's virus charts. However, modern trends demonstrate that more and more of them are moving into the worm group and, with each year, the macro part of the pie will become smaller and smaller while the worm section will grow steadily and quickly.

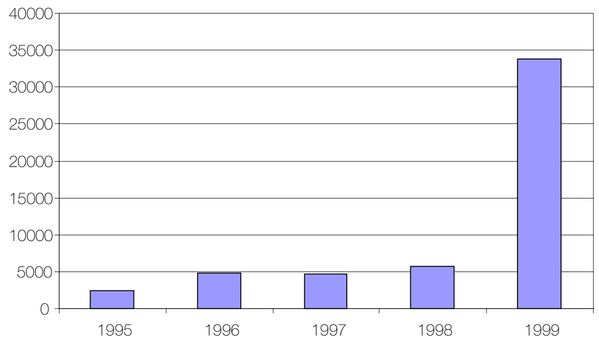


Figure 2: Number of virus incidents (1995–1999)

You would probably be amazed at the enormous virus turnaround in 1999. A closer look reveals that more than 8,000 of a total of 33,830 incidents reported were of ColdApe, which is actually a false representation.

However, even if we omit the ColdApe virus again we can see that 1999 was still the year when the highest number of separate virus incidents were reported. Who knows what will happen this year? Another VBS/LoveLetter epidemic will catapult 2000 into the lead. On the other hand, viruses are just like seasonal workers; it is very hard to predict when they will appear. In any single month the number of virus incidents could double or even triple because of a brand new virus.

VB Prevalence Table figures confirm that macro viruses are losing their dominant position to worm-style viruses. However, the macro viruses still prevail. Together with worms written in script languages, they will be the major threat to both individual and corporate users in the future. There is a reasonable explanation for this. Firstly, it is very easy to develop a macro or script virus. The only thing a virus writer needs to have is a basic knowledge of VBA or VBS programming languages. They are so simple that even a schoolboy could manage this in a couple of weeks. Secondly, these viruses are available as source code. This means other people can easily construct their own viruses by applying the slightest change to the original. Thirdly, these viruses are aimed at the most popular applications, which are used by millions of people worldwide.

Finally, these applications usually have poor protection with many security breaches discovered every month. We are very lucky that virus writers neither pay enough attention to security-related Internet conferences nor seem to have the money to subscribe to *Virus Bulletin*. Otherwise, they would issue a new virus exploiting security breaches each time they are discovered!

It is no secret that nowadays the most dangerous application a user can have is *MS Office* (43.3% of incidents in 2000 occurred on this platform). The problem is that *MS Office* usually runs on an operating system called *Windows*, which is not safe either (29.7% of incidents). By default, *Windows* has a *Scripting Host* installed, which has encountered 26.2% of incidents so far this year. The news would not be that bad if the vast majority of computer users stopped using any of the applications mentioned above. But the harsh truth is they do continue to use them.

I see two ways to alter this state of affairs. The first involves a general migration to alternative operating systems, office and email applications. Even so, it will not save us from viruses from here to eternity. As soon as, for example, *Linux* becomes as popular as *Windows*, the viruses will follow suit – probably even more dangerous than the ones we have now. The development of new technology as regards anti-virus protection will help. In addition to this, the now common practice of combining different anti-virus defence methods such as behaviour blocking (sandboxes) will make protection more effective. This is one of the industry's most promising technologies which allows virus detection not by searching for unique signatures but by blocking the virus's activity, which is limited by the application or operating system.

Conclusion

We can point out the breaches viruses can exploit, new applications they can move towards and even useful ways to protect against them. The only thing we are unable to do is to fix so-called 'mind breaches'. Users are responsible for protecting their computers. And nothing, not even the best AV software available today, could be more effective against viruses than the basic rules of 'computer hygiene'.

BOOK REVIEW

To Boldly Go?

Matt Ham

The Enterprise Anti-Virus Book
Robert Vibert
Segura Solutions Inc
US:$64.95/Canada:$59.95/UK:$69.95 (Can$ inc. p&p)

This book has a self-explanatory title and an author who, unusually if a recent *VB* Book Review is anything to go by, has recent relevant experience in the anti-virus field. It is definitely not, however, what might be expected from the title alone. The stated aims of the book are admittedly not particularly at odds with the title, these being to act as 'a practical tool …to use in conducting evaluation of anti-virus products', so why make such a sweeping statement?

This mighty enchiridion of the mystical arts is, not unusually, laid out in a series of chapters, each after an introduction consisting of a series of questions, followed by a summary of these questions in tabulated form. A glossary, bibliography, list of infectable objects and order form round off the work. From a purely aesthetic and style-based perspective the tome disappoints almost immediately, with ugly typeface and a peculiarly chaotic layout within some of the chapters. Information is in some cases culled from Web sites, with the point of insertion depending upon what appears to be an advanced form of feng shui rather than any publishing science. There are also esoteric references to order forms which do not exist, together with missing or scrambled text in numerous places. If there is not a great numerological explanation for this confusion then Mr Vibert would be advised to invest in a good proof reader.

Rather outside the tradition of AV-related documentation, the introductory chapter is prefaced with a series of commendations from some of the author's chums. More typically, there follows a disclaimer which, if taken literally, prohibits the book from being used in any way related to computers, warns that the advice it contains is '… not legal or professional …' and declares that no wombats were harmed in its production.

These and the previously mentioned arcana may soon be a thing of the past, however, as there is next mentioned a plan for the book to be updated approximately monthly – updates being available for a subscription fee ($15 for 1 year). This anti-virus company-style tactic has now been demonstrated and with new virus threats emerging at every turn, it is admirable that the author does not claim prescience. It is something of a relief in such a book to see plans for the future relying upon an honest 'anything can happen' outlook, rather than the wild tea-leaf reading applied as a certainty, which has characterised some attempted reviews of trends in viral matters.

After this cursory glance into the crystal ball, how does the book fare as a tool for the professional? The opinion engendered was that it asks many of the right questions, a few of the irrelevant questions and nothing but the questions. There are some didactic statements of more or less debatable truth, but by and large there is a paucity of answers which serves all the more to intensify the feeling that the book is some variety of ill-favoured seance.

Only at one point does Mr Vibert digress to a more opinion-based format, and this is to berate *Virus Bulletin* for its apparently overly theoretical viewpoint – the whole object being yet another voguish attempt to claim that *VB* is biased towards a certain anti-virus company.

Though some of the questions manage to plumb the depths of the ridiculous – would anyone judge an anti-virus product by asking the developer how many courses their programmers have been on? – there is a great deal to be said for many of the questions, inasmuch as they are not obviously of importance until further implications, mentioned in the text, are considered.

For the anti-virus tester, there is a wealth of material for instigating new and interesting (not to mention bizarre and outrageous) test regimes. For the implementor in an enterprise, however that may be defined, with neither the time nor the inclination to perform these tests, the utility of the book would seem to be less obvious.

Would I recommend the book? Certainly, but not primarily to the intended audience. Mainstream magazine reviewers of AV software should have this book, or a close equivalent, as required reading. An idea of the relevant issues for real users, combined with the depth of investigation required to make a realistic judgement on a virus solution, might well improve the generally ectoplasmatic nature of such reviews. It will certainly make this reviewer's task easier.

Multi-product resellers too could benefit from the questions within, as an aid to reviewing their clients' needs and producing detailed reasons as to why a particular product matches each customer's requirements. Reviewers and resellers share common traits, the need to know anti-virus software in detail and the degree of priority that enables this to be given time.

These very reasons why the book might be useful to the aforementioned types, would not, on the other hand, recommend it to all but the most keen and AV-dedicated corporate user. The vast majority of IT specialists within a company will need answers rather than questions, and lack the time to cast the auspices directly. Ironically, while creating a book which ostensibly lessens the need for third party sages, the author seems to have proved all the more a need for such dedicated information sources.

CONFERENCE REPORT

Tracking Bugs in Ontario

Nick FitzGerald
Computer Virus Consulting, New Zealand

Hosted by Russ Cooper, moderator of the mailing list, the 2nd Annual *NTBugtraq* Conference was held on the shores of tranquil Sturgeon Lake, one of the numerous Kawartha Lakes in Ontario, Canada. Perhaps unintentionally, the proliferation of winged, biting insects added more than a hint of irony to proceedings.

Aside from the surroundings, the conference attracted a diverse range of about forty systems administrators and consultants with more than a passing interest in security issues. I attended as the first of the three days squarely focused on virus and Trojan issues, and it seemed likely much of the third day would be slanted that way too.

Of Shoes and Ships and Sealing Wax

Bill Murray's session on the first morning focused on client technology as it relates to worms, viruses and Trojans. Issues surrounding the 'virusability' of computer systems and related risks were covered, with the old adage 'separate data and code' arising several times. Many examples illustrating how failure to enforce such separation had led to compromised integrity and/or security were presented.

An example Murray discussed that seemed particularly apposite was the .sy command in the message system of *IBM's* PROFS. This was the 'system' command which caused the message interpreter SCRIPT to run the program supplied in the .sy command's argument. Of course, this process was run with the full privileges of the user reading the message. Thus, a less privileged user could elevate their privileges (or run a command with higher privileges than they had) by sending a suitable email message to a more privileged user, such as a system administrator. Once the scope and seriousness of this (potential) problem was realized, *IBM* fixed it by adding a command line switch to SCRIPT. This switch had to be specified when invoking the interpreter to enable it to act upon .sy commands. Of course, this 'fix' broke any existing mechanisms that depended on the original behaviour, but it was decided this was the price that had to be paid.

Parallels with recent issues around embedded scripts within HTML files and macros *in* document files will be obvious. However, it is interesting to note that absolute methods for blocking such functionality are not as forthcoming from the latterday vendors. Of course, there have been other, very similar, cases. For example, the .pi, .pso, .sy, .opena and related macros in the Unix roff family opened the possibility of Trojan man pages through an identical flaw, and no doubt there are several others. When will we ever learn?

Another major thread to Murray's presentation considered how the malware scene may change if specialized, but fixed functionality, computing devices become common. We are already seeing a huge proliferation of handheld, immutable code computers (cell phones), and a future with many other such devices specialized to one function, which they will perform very well, does not seem unlikely to Murray.

In such environments, the core requirement of 'virusability' – the ability to change the code of the (potential) host environment – is missing, so viruses cannot exist. These devices' functions are limited and fixed near manufacture. In typical desktop computers, functionality is not fixed until run-time and may change before the next run. The *iPic* may be a precursor to that future – a Web server the size of a match head that can be made for less than a dollar. The content of an *iPic* is set at manufacture which may seem rather limiting, but at less than a dollar, replacement is not prohibitive should upgrading ever be necessary. Murray sees great hope for such early binding of functionality as a solution to increasing demands for integrity and security in consumer computing devices, rather than trusting yet more services to typical computers and the Internet.

The rest of the sessions covered more traditional issues of concern to the computer security camp, with the expected slant toward *NT* and *Windows 2000* OSes and applications. Given the ratio of systems administrators present, I was surprised by the acceptance among attendees of their users being able to significantly alter the code on corporate machines. Of particular value was *Microsoft's* participation in the conference, with Scott Culp (secure@microsoft.com) and Chris Walker (head of its internal tiger team) attending, openly asking and answering questions (quite often pointed ones) and generally engaging in proceedings.

On a Personal Note

It was a great pleasure to meet Bill Murray, and to converse with him, Bob Abbott and some other 'seasoned old-timers' (I hope they do not mind being characterized thus). A theme common to many of these private sessions was how so little seems to have changed in forty (and more) years of computing. Mistakes uncovered in the 1960s (and earlier) are still being repeated today. The languages and technologies involved have certainly changed, but similar classes of design and/or implementation error keep recurring. It is too simple to blame lack of interest in the collegial history of computer science solely for this, but I was left wondering whether lack of interest in the discipline's history might not somehow be overcome to help break the 'cycle of errors' that kept cropping up in these discussions. I doubt I'll attend another *NTBugtraq* conference unless there is a strong focus on anti-virus issues again, but I am pleased to have attended this one.

OPINION

Form, Concept and Other Pleasant Utilities

Randy Abrams
Microsoft Corporation, USA

At VB'99 in Vancouver, I made an appeal to the anti-virus community in my presentation 'Giving the EICAR Test File Some Teeth'. I asked the anti-virus industry to extend the *EICAR* test file. Specifically, a macro 'test virus' and a boot sector 'test virus' were requested. Subsequently, the audience, and I on the podium, were graced with the comments of several of the researchers who brought the *EICAR* test file into existence.

Some anti-virus researchers were less than keen on the idea of these test files for a variety of reasons. Most of the objections dealt with a macro-based test. The main objection or concern was the problem of macro mating. It was reasoned that the test macro might be hijacked and one of two calamitous situations might result.

In the best case scenario, the test macro might be caught and the virus missed. In the more severe case, the mating might create a new, and presumably undetected, macro virus. The possibility of an *EICAR*-style macro test was discussed as being something that might best be done independently – each company makes their own.

Little was said with respect to a boot sector 'test virus' other than the obvious fact that when you toy with a boot sector, particularly on a hard drive, you run the risk of complications that result in data loss.

Macrostein – the Zipper-head Macro Virus

We cannot just dismiss the problem of macro mating out of hand. Let us delve into the problem a bit. To begin with, macro mating is here. This will not be a new problem brought on by a test macro.

Unlike current macro mating projects, an industry-standard test macro means that all vendors have a consistent macro with which to study the effects of mating. If each company produces their own macro test, the complexity of the job will be increased. It must be noted that in order for the mating to occur, the file under test is almost certainly infected already. Not just infected, but probably infected with a virus that anti-virus software is not detecting!

We should also note that if the test were to fail, i.e. the users' anti-virus software is not functioning properly, the ability to test, assess, and correct the problem might lead to the discovery of infected documents and result in disinfection – a desirable result.

Don't Give Your Data the Boot

Admittedly, a boot sector test virus is a trickier problem. It is likely that the best we could hope for is a floppy disk boot sector test. The requirement for all of the data on a hard drive to be backed up before playing with the boot sector tends to make hard drive testing more complicated than floppy-based testing. A floppy disk test is quite possible, however.

Of course, we have the potential for compound infections but again, this requires an infected test disk that, presumably, the anti-virus software is already unable to cope with. We would also have the benefit of a known boot sector modification against which testing can be performed if a standard boot sector test is adopted.

Do You Really Need a Letter Opener for Email?

So, the obvious question 'Is there a need for these test files?' will be asked and must be answered 'Yes!'

I am certainly not going to get off that easily, so I shall presume that 'Why?' is the next question. First, the case for testing must be made. Once this is done, the case for safe test files follows naturally.

My experience with several virus scanners and millions of scanned files has taught me that anti-virus software, like any software, can experience problems. The problems range from bugs to user error, with bad, or less than intuitive, user-interfaces falling between the two classes of problems. I have yet to test or use a product that did not at one time have an issue serious enough to be able to cause the scanner to miss an infection.

Without giving too much of a future presentation away, I will share a couple of cases with you.

The context-menu scan (right-click and scan) for one product I have recently tested will not scan for boot sector viruses when the target is a floppy drive. The dialog box after the scan does indicate that no boot sectors were scanned, but it is definitely not a clear, intuitive, or expected message. The failure of the context scan to attempt to search for boot sector viruses when applied to the floppy drive is important to be aware of. Users need to know that this is the behaviour.

So surprising was this result that I went ahead and tested a few other scanners. None of the other scanners I tested exhibited this behaviour – at least, not in their current incarnations. Is there really anyone in the reading audience who questions the need to know that your scanner reacts this way? How do you determine this? I know what I did; I used a live boot sector virus.

Without naming names, I think most of us recall a rather embarrassing incident in which a product reviewed by *Virus Bulletin* would not detect boot sector viruses on floppy disks that had no files. Fortunately for most users, the product was only available for a short time. Unfortunately for me, I managed to download it during that short time. Fortunately for me, I had no blank floppy disks with boot sector viruses and the version of the product had other problems that prevented its deployment.

Does an IT professional need to know if their product detects boot sector viruses? I'd be surprised if *Virus Bulletin* receives a single letter from an IT professional who says no to this. While this case points out a limited circumstance (blank floppies) I have encountered much more global examples.

In one case, I personally witnessed a product that was failing to detect macro viruses inside documents with .HTM extensions. Currently, the use of a live virus is the only way I have to test for this scenario. When O97M/Tristate came out, I needed to know if my scanners would detect a macro virus in a *PowerPoint* presentation. I was able to embed an *EICAR* file and determine which scanners were detecting embedded objects, but this is not the same as testing for macros. How do users determine their scanner's ability with respect to new VBA implementations in *MS*, or other products that licence VBA? I know how I found out. I used a live virus.

I have several more examples of the need to test. Some of these cases are addressed by using the current *EICAR* test file. Some of the cases currently require a boot sector virus and some testing does require a macro virus. When I present these failure scenarios, I fully intend to share solutions. What solutions do I have to share for boot sector and macro testing? Currently I have no palatable solutions.

While I must advise testing, I am left with no choice other than to recommend the extremely careful use of live boot sector viruses and live macro viruses. This means users will have to collect samples by some means. It is likely that some users will find that, for boot sector testing, the Form virus is the easiest to find. Unfortunately, this is because it seems to spread so well. All of the tests that I am recommending to users should be able to be performed without the use of a live virus. The lack of a safe, non-viral test for boot sectors and macros makes this impossible.

An industry standard for each test type is preferable. When a typo is made in the *EICAR* test file, one can cross-check their work with a second scanner. Typically, if two scanners miss the *EICAR* test file it is due to user error and not a scanner problem.

Recently, a colleague was running a battery of tests and found that the scanner was not detecting the *EICAR* file in a ZIP archive. The ability to cross-check his work with a second scanner and prove he had the file right was valuable. When testing for boot sector detection, the ability to rule out hardware malfunction is quite handy. I have no doubt that a macro test file would also lend itself to typographical errors every now and then.

EICAR-amba!

Since October 1999, scripting viruses have come to the forefront. Since a Visual Basic or JavaScript file is a plain ASCII text file, detection of the *EICAR* test file in such files does not give meaningful information about whether or not a scanner can detect scripting viruses. To muddy the waters further, one scanner (at the time of writing) detects *EICAR* and detects scripting viruses, but does *not* detect *EICAR* in files with the extensions .VBS, .HTA, .HTM or .TXT. As scanners become more intelligent about what they look for and where they look for it, users will require a wider variety of effective test files or they will be forced to turn to live viruses for verification of scanner functionality.

Home users already have the need for a script test file. At least one product does not check for scripting viruses in its freeware version. With JS/Kak and company occupying the top two positions in the *VB* May 2000 Prevalence Table there can be no argument against the importance of knowing if your scanner can detect scripting viruses.

The Jury Finds the Defendant

So, what will the verdict of the anti-virus industry be? We can find any number of reasons why a test file might be a bad idea. Of course, we have no proof one way or the other, until we have such files available. As the professional anti-virus researchers ponder the implications of the proposed files, I am compelled to request that when you deliver your answer, you put it in the context of an answer to the following questions:

• Do the risks associated with macro mating, and/or any boot sector combos, outweigh the obvious benefits of being able to eliminate the need for live viruses in an entire class of tests that users have a need to perform?

• I am still writing my presentation. By November I will need to know what to tell the audience. Do I tell them that they will have to find some viral samples and learn to handle them carefully? Do I suggest specific samples? Or do I tell them that the anti-virus industry has provided them with the tools they require to ensure that each time they upgrade their product, or their environment changes in any of a number of ways, they can test their products safely without fear or risk of infection?

I enjoy the ability to do a significant amount of testing on a networked machine with the non-viral *EICAR* test file. The ability to extend the scope of these tests should be valuable to many other IT professionals as well. Perhaps in the coming issues of *Virus Bulletin* I will find the materials for the portion of my presentation that deals with ensuring your scanner can detect macro and boot sector viruses… a script test file would benefit consumers too!

A DAY IN THE LIFE

Manic Monday

Patrick Nolan
NAI, USA

Just as the song says, ‘Six o’clock already? I was just in the middle of a dream…’, I too have fallen victim to the serenity that we call sleep. Without it, we would fail to function on all eight cylinders. I have to remind myself of this once in a while because every so often I find myself staying up with the nocturnal creatures watching over Internet postings and sample submissions like a hungry cow watching grass grow, or something like that. That’s what I found myself doing yesterday, but today is a scheduled workday. I can’t call in sick, I haven’t done that since I was 19. Oh well, you only live once, unless you’re Shirley Maclaine. I’m not Shirley.

I get in the car and start off for my morning trek to work. I count the days until we move into the new house. It is three times closer to the office than where I live now, which is three miles away. Two stop lights and two stop signs – now that’s living. My objective is to live as close to the office as possible. There is some statistic about how most driving accidents happen within one mile from your home. Maybe if I live close enough to the office, I won’t have a driving accident. I haven’t had one yet, maybe this tactic is working. I quickly knock on my forehead in lieu of any other wood to knock on.

I get in to the office after passing my badge across the electronic door lock device. A few mornings in my haste, I have been without my badge. I contemplate getting it implanted in my hip. My area is fairly close to the main entrance. If it wasn’t for my gym membership, I don’t think I’d get any exercise.

I mark today’s date on my journal and reboot my machine. If I don’t reboot my system, strange things can happen. Maybe I’ll install *Windows 2000*, I’ve heard that is a fairly stable OS. I use *WinNT* on two systems but the one I use most often is *Win98*. This is the one that requires rebooting.

The *Win98* system has two LAN cards that run on different networks. I have two hardware profiles; one profile has LAN card 1 enabled, two disabled and the other profile is reverse. It’s always funny to me when I boot the system and I get the message ‘Windows cannot determine which profile…’. After logging in, I start up the email application. The best part of the email program is that you can set up rules. The bad part is that most of mine are client-based, meaning that they aren’t in use or functional until I start email. I don’t mind unless someone has been busy posting announcements to the list groups to which I subscribe. So what I usually end up doing is starting the email program and then take a break. After about five minutes the emails have been shuffled and sorted, and put into their neat folders by category. Today it seems that someone was busy over the weekend with postings.

Checking my Inbox, I find several requests for virus descriptions. I like writing them but I don’t get a chance to be as creative as I’d like. I am not allowed to use hyperlinks in the first 200 characters of the description.

Apparently, this messes up some automation script that IT uses for parsing details into usable HTML keywords, or something like that. I would like to add some flashy Java or maybe some background sound, or perhaps some blinking text or a marquee. Oh I know, not everyone uses *Internet Explorer*, not to mention that some of those styles can get pretty annoying. Some would find it annoying to have blinking text ‘Update your DAT files unless you want to become infected by this virus’.

Scanning through other email messages, it appears that someone wants clarification on how we handle the Kak worm. Do we clean up all Registry entries, what about the AUTOEXEC.BAT file? No, we don’t touch the Registry, no the user needs to delete the AUTOEXEC.BAT file and replace it with the AE.KAK file. I ask them if they read the on-line description. They replied with a question – ‘What description?’. I sigh.

Another person wants to know how to clean a password stealer. I am led to believe that most people consider everything a virus that requires repair and usability of the file that was detected. I explain to this person that they really don’t want to keep this Trojan and that they need to delete it from their computer.

My Inbox rule alerts me that I have a new item to look at. Checking briefly, it appears to be a new virus of the VBScript type. I’ll need to make replications and check to see if this is an announced virus or if I will need to assign a name for it. It doesn’t replicate and I don’t see a posting for anything like this. I decide to call it VBS/Moron.intd after the person that wrote it. Analysing the code, I look at why it fails. Yes, here we are, looks like the writer used some type of editor whereby they could use a ‘find/replace’ method because the variable is two words with a space separator instead of one word – hence the name Moron.

Here’s another email which someone is convinced is an automated process that emails to everyone in the address book. It’s a new kind of Trojan or mass-mailer inside a *PowerPoint* presentation according to the customer. Apparently this new virus is hitting them hard. My eyebrows rise in scepticism – I fire up the replicator system and image a drive for the necessary OS and applications and then copy the sample to this system. I start the presentation and find out that it isn’t a mass-mailer.

It isn't even a Trojan. It is exactly what it says it is, an off-colour parody of the game show program 'Jeopardy' in which players must reply in the form of a question to the answer presented. There isn't an embedded object anywhere in the slideshow. The structure analyser indicates that it is completely textual with a few inserted clip art images. I determine that users at the customer site were forwarding the slide show presentation manually and the system administrator picked up on the seemingly replicated email messages as a trigger. You can't be too careful, I suppose.

I settle down to write a few descriptions for my adoring fans. The coolest part about writing them is that you can be anonymous. Sometimes I add a little something but most of the time I just write them how I see them. That reminds me, there's something in my Inbox that needs some attention. I receive a sample of a new, cross-application email worm and it looks like it could be something. I run it through the test system and it does everything that it suggests it will do. This one uses social engineering and an exploit of the OS. I call this one IRC/Stages.worm and write a description for it.

Don't look now but it's lunchtime already. I'm not ready to break from my computer yet because I have two more Internet worms to test out. Success breeds success, or so they say. I'm determined to find out if these other two samples are as successful at breeding as the first one. The next one is kind of silly really. It's a prepender, which announces itself quite irreverently. This one I'll call IRC/Scrambler because of its nature. Quite an exotic name for such a simple worm though. It would be more interesting if it actually scrambled something. I like scrambled eggs. Oh, that's why we call it Scrambler, take a look at the MP3 file that it jumbled. That wasn't a nice thing to do, no sirree. Halfway through lunch, I get a page from someone in support wanting to know if I'd ever heard of a virus called 'Jeopardy'. I think I'll let this guy sweat for a few minutes while I enjoy this fresh slice of pizza.

I get back to the office and there is a request for a description called Cybernet. Apparently it is the next big craze. I wonder what the world is coming to. Looking through the code, I see tell-tale signs that this virus writer believes he has the perfect virus, one that will certainly avoid detection by AV software. There are a couple of demeaning comments in the code, one that catches my eye:

```
'anti-heuristic for stupid McAfee antivirus
scanner
```

I think to myself 'foiled again'.

After processing a few more customer issues and writing responses to them, I check out what's going on in the newsgroups. ACV has a life of its own it seems. It bobs and weaves through thick and thin, filled with lurkers, experts, novices, some with propensity to learn, others with propensity to be dense. I can't remember a day when there wasn't a controversial posting. Sometimes it wouldn't take more than a posting of a binary to start a thread war while other days the posting of a request for copies of the latest virus would get you into hot water. It's always good for entertainment of some sort. Today nothing catches my eye so I go to the cafeteria for another Pepsi.

I get back to my area and find a discussion already in progress. It appears that a client has an infestation of the FunLove virus. There are questions about how this could happen and how do we remove it. After a bit more digging, we find out that the server software was down-level and much of the client software engines were not upgraded either. I think back to the blinking text.

Then I wonder about virus-naming conventions. They sound soft sometimes, I mean think about it – FunLove, NewLove, LoveLetter, Pretty, FreeLinks, Happy99. Then I remember that there are some other ones that did get appropriate names – W97M/Backhand, W97M/Pathetic, W32/BadAss, and W97M/Jerk.

Soon we get to go play. In the cafeteria there is a 'foosball' table brought in by one of the developers. It gets heavy use, mostly by a handful of backline support persons and QA testers. I've also joined the banditos and try my game during our available time after 5pm. It's a really cool way to unwind. There is something refreshing about the sound of a plastic ball landing soundly in the defender's goal. I'm perfecting something I call the Z-shot. Basically you do it from the goalie position. There is a certain way you can smack the ball and it bounces off of the closest sidewall and off the opposite wall then into the goal. Maybe I'll trademark the shot. Maybe not. I like playing defence rather than offence, unless I'm defending against Jimmy Kuo. That's OK though, someone said what doesn't kill you will make you stronger. I'm not dead yet – I don't feel stronger yet.

A few more items to check out in my Inbox – someone believes they received a Trojan by reading email. It doesn't pan out so I dismiss it. Another user sends source code of a proposed virus written in Assembler. I thank them for their efforts anyway. I receive a few macro viruses that are already detected. Again I remember the blinking text.

My radio is tuned into a station that plays only music from the eighties. I seem to remember that era as my favourite. I know the words to almost every song that plays. I don't always listen to this station. Sometimes if I'm in a kick-butt mood, I'll pop in some Limp Bizkit and put on my headset as to not disturb my fellow virus hunters. After three hours of listening to this album repeat itself, I've had all the head bang I need to play basketball at the local gym after work.

PRODUCT REVIEW

VirusBuster for NT

A debut for *VirusBuster* in the Comparative Review last month might have led to slight inquisitiveness in our readership concerning what appeared a promising newcomer, if one with some ground to make up. This thirst for information can now be quenched.

Although a newcomer to *Virus Bulletin*, *VirusBuster* is certainly not new to the world of anti-virus, a fact attested to by its fair performance in that recent Comparative. Lending its name to its product, the Hungarian company *VirusBuster*, aided by a good user base in its homeland, is now attempting, as many locally successful developers have, to become a player upon the global stage.

The versions reviewed, therefore, were something of a mixed bag. The full, currently available boxed product was supplied, the applications on it being bilingual English/Hungarian. The manual and some help have not yet been fully translated, however, and thus a handy Hungarian was obtained to get a feel for the bulky printed documentation. As for what the conclusions were after the *Virus Bulletin* inquisition had finished, the examiner's report follows.

The Versions

The *VirusBuster* team had clearly read past reviews by this tester, as the box sent was pristine when unwrapped from its packing material. This was, admittedly, achieved by the box itself being a self-assembly affair. The 'contents' would otherwise have been a CD and manual, the latter in Hungarian. Platforms supported are DOS, *Windows 9x*, *NT* server and workstation, *MS Office*, and *NetWare*, with the addition of an *Exchange* server product.

The presence of said *Exchange* product is not so much of a surprise because this is *Antigen* by *Sybari Software*, rather than *VirusBuster's* own creation or, as yet, *Antigen* incorporating a *VirusBuster* engine. Accompanying documentation explains that *VirusBuster* has its own such application but has decided to become resellers of *Antigen* rather than market their own product in Hungary. This leaves the peculiar situation in which an anti-virus developer is currently reselling licences for other companies' software, namely the choice of *Network Associates Inc*, *Norman Data Defense* or *Sophos* components of *Antigen*.

For the purposes of detection and speed tests, the *English NT* version was used. Comparisons were also made with the recent *Windows 98* Comparative results, so as to obtain a feel for the extent of improvements. The *NetWare* version will be included in the Comparative next month, and readers are reminded that *Antigen for Exchange* was reviewed in the May 2000 issue. Excluding *Antigen* from the reckoning, the products all state quite modest system requirements – DOS at a *386* with 4 MB of RAM, rising to a maximum of 14 MB for the *NetWare* version. Hard drive capacity required is also on the smaller end of the scale, ranging from 2.5 to 5 MB.

Installation and Documentation

As has become the expected norm, the *NT* version is an InstallShield production and thus well supplied with those little bars which go up and down for no readily obvious reason. This gives three options for installation, the usual typical, minimum and custom being offered, though in an attempt to confuse the onlooker the minimum install is the default offering.

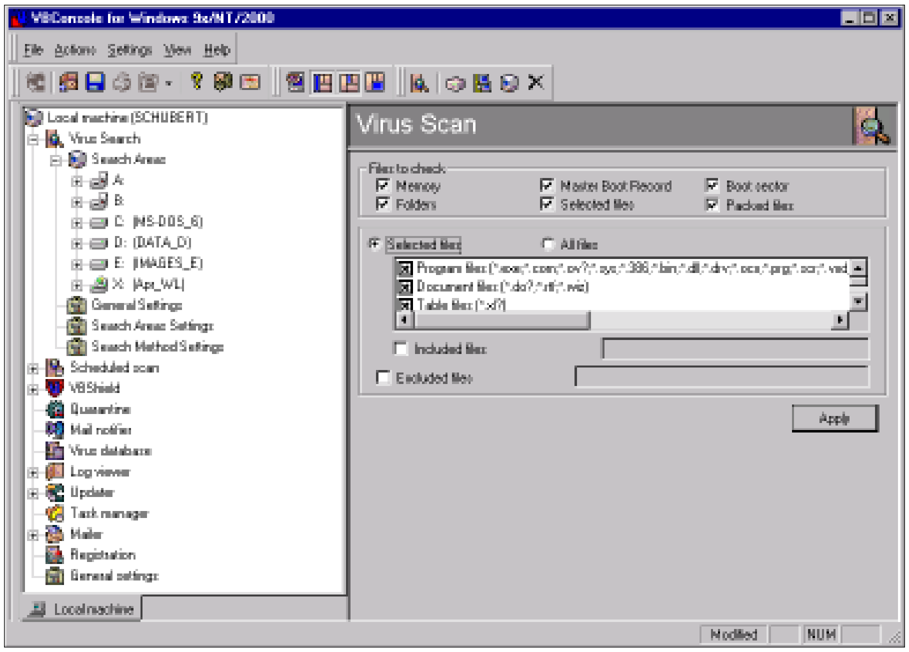
Since this was not a Comparative Review, the custom method was chosen, with categories being the on-demand *VirusBuster for NT*, the real-time VBShield and mystical 'Optional Components'. All were selected, mainly because the optional components were not described sufficiently to give any idea of what they might be – this information being hidden within a further submenu. After this confusing start, the actual installation process proved to be as simple as might be hoped – the only anomaly being a lack of the registration input screen common at this point. One observed difference between custom and other installations was that SMTP notification settings were also configurable during that type of installation.

For completion of the software's setup a reboot was required, after which the VBShield component declared itself unable to start due to a *Microsoft* shared DLL problem. When a newer version of the offending file was pilfered from another of *VB's* test machines all did, however, load as initially hoped. The Start bar icon for the VBShield acts as link to the configuration page for this on-access scanning component, with the main Console providing more universal configuration via the Start menu.

The Start menu is also blessed with various other related application links, though being labelled in Hungarian the uses of these, other than Uninstall, were not too clear. The console is also the site where the registration number is entered for the application.

Interface

VirusBuster is controlled almost entirely within the VBConsole application – those areas where it is not are limited to installation, which would be tricky indeed to perform from within the application to be installed. Access to program controls is through categories of activity, these being accessible through nearly all the possible methods which could have been included. Drop-down menus and a



Files to check
- [✓] Memory
- [✓] Folders
- [✓] Master Boot Record
- [✓] Selected files
- [✓] Boot sector
- [✓] Packed files

[✓ Selected files] All files

- [✓] Program files (*.exe;*.com;*.ov?;*.sys;*.386;*.bin;*.dll;*.drv;*.ocx;*.prg;*.scr;*.vxd
- [✓] Document files (*.do?;*.rtf;*.wiz)
- [✓] Table files (*.xl?)
- [ ] Included files
- [ ] Excluded files

horizontal toolbar are both supported, while those using laptops or with dimmer visual acuity will be pleased by the large alternate vertical icon bar. These all link to the central *Explorer*-style configuration tree. Each of these, or the tree, can be used to bring up a settings page for the appropriate area on the right of the screen, with a status and message log being placed below.

The machine-specific configuration tree is tabbed, though the extra *Management Server* is required for access to other machines. In the *NT* workstation version upon which tests were performed, however, this is the central point for configuration with only the one local machine tab.

Despite the presence of several other methods of reaching and editing this information, the tree seemed by and large the most convenient control point, though credit must be given for the supply of the other methods.

The layout as described so far is also merely the default installed configuration. It is possible to remove most of these components from the main view, or to rearrange them, though the implementation of a truly floating interface design is only partial. It is also possible to close some portions of the interface with no method visible for returning them, short of exiting and returning to the program. Another oddity is that the vertical icon bar is implemented as a loop, which at first caused some confusion before making it easier to use in the long run.

Configuration

After such a discussion of the interface, it is obvious that there are many configuration options and indeed many ways to access them. So without further ado (other than this introduction) these options will be revealed. As with general categories alone, numbers of controls do not disappoint – rather than the ‘list then reiterate’ method so common in school rooms, subheadings seem a more user-friendly system.

Virus Search

Not unexpected in an anti-virus program, there are a few nice surprises in this section. Options allow for the selection of search areas, though a little disappointingly these are not able to be saved as specific jobs within this part of the console. This must, paradoxically, be done with the scheduled scanning options – creating a job but not implementing it as part of a scheduled job. There is provision, however, to save overall configurations for later use, if specific combinations of areas are to be scanned on an irregular basis using specific scan settings.

Areas to be scanned are also cleared after each scan, which caused more than slight irritation in the testing procedures but would probably be less of a worry to the average user. Selection of areas is aided by context-specific additions to the toolbar, these allowing mass selection of all, local, or network drives.

Within the settings for scans, interactive communication (that is, the necessity to respond to virus alerts) may be deselected. This should, and did, inspire a hearty sigh of relief from any *Virus Bulletin* product tester. Somewhat more confusing was the option to remove the progress status bar from quick scans – since quick scans are mentioned at no other point.

Exclusions are supported, though without an option to browse for areas to be excluded which could prove frustrating. There is, on the other hand, a pleasing degree of control available over which infectable areas should be scanned, with not only, for example, memory, boot and packed files being selectable individually, but a categorization of file extensions within the ‘types to scan’ choices.

Unlike the usual incomprehensible list of extensions, there is an overall ‘all files/selected files’ option and within the selected files the extensions are categorized under document, program, script and the like. Within these categories each may be individually activated or deactivated for scanning, or this may be done at a group level. This not only gives a good idea of why each extension is there, but it also makes fine-tuning simpler for the operator who does not know his OCX from his ADP.

As far as control of scanning methods is concerned, there is also a good deal of fine-tuning possible. Scanning is broken down into Normal, Heuristic and Macro-Heuristic sections, each of which may be set at three different levels of sensitivity. Heuristics of both types may also be disabled.

The detection and scanning speed tests performed gave interesting data as to the differences between these settings, of which more later.

In the same area, action upon detection of a virus can be chosen with, again, the distinction being made between the three types of scanning. This differentiation is useful for those situations where positively identified viruses must be deleted, whereas heuristically detected potential viruses would be best dealt with on an individual basis.

The standard options of Keep, Quarantine, Rename and Delete are available for Normal detection, with Delete unavailable on Heuristic, and a purely macro deletion option for the Macro-Heuristics. The slightly misleading category Kill is used for disinfection.

Scheduled Scan and Task Manager

As previously mentioned, the former is not entirely an area where scheduled jobs are run and it would perhaps be better labelled as ‘user defined jobs’. Jobs can be saved for selected areas, and a browser is supplied to facilitate this, with a direct link available for the Task Manager. Tasks can be scheduled from the overly optimistic ‘once every 9999 years’ to the ultra-paranoid ‘once every minute’ with a good range of user definable and sensible periods between.

Days of the week may be specified and date or time ranges given during which the scans should be performed. As with the choices for scanning, the control, although looking at first glance quite standard, has added a nice feature to the standard set without sacrificing usability. Also as before, there is a slight niggle to pair with this, namely that the global settings for scans seem to be applied to all scans, thus disallowing the choice of a paranoid scan each week combined with a more relaxed regime for each other day.

VBShield

The on-access element of *VirusBuster* is very similar in its configuration method to that of the on-demand component described above. There are some extra commands available, though in at least one case these would not have seemed out of place if available in both areas.

The primary option in question is that of being able to set areas as excluded from scanning – although not strictly vital in the on-demand scans, this might be expected to ease matters sufficiently when performing complex scans, and it might be worthy of inclusion on-demand. The trend of novel features continues with the inclusion of a protection scheme for oppressed files. This allows, for example, the protection of all .DOC files from edits, renames or deletion, except those in certain directories – a useful feature despite its being more or less available in *NT’s* permissions system.

Quarantine

There are definitely no surprises here, with only the ability to set a quarantine area available.

Mail Notifier and Mailer

The mail alerter allows a wide choice of alert conditions under which mails will be triggered, lacking, alas, any ability to alter the contents of these mail alerts. Virus alerts may be selected for suspicious files or just definitely viral ones, with most other alerts concerned with errors within the application (this is the first such program – with a specific setting for Exceptions as distinct from Critical Errors – that this reviewer is aware of).

For the security-conscious user, alerts can be issued upon the alteration of program configuration, though it is to be hoped that all notified users share the same level of enthusiasm and responsibility, since there is no provision for individuals to receive independent levels of alert depending upon their preferences.

Virus Database

This part is apparently still in translation and thus remains a mystery for the time being.

Log

The Log turned out to be the area in which *VirusBuster* most easily frustrated, a problem which will probably apply only to those who seek to reproduce *Virus Bulletin’s* results rather than real-world users. The nub of the glitch is that the log is produced not in plain text but in a large conglomerate file with the traditional coding of format by slash-controlled characters. This may be exported to a text file but this did not appear to function with the vast *VB* logfiles. Enough of self-pity for the log is, in fact, quite useful from a purely visual point of view, though it is not a particularly spectacular specimen of the breed.

Updater

The Updater, as the name so cunningly suggests, allows for updates and is linked to the Task Manager described above. Obviously, therefore, updates can be scheduled at particular times, with the sources being selected from CD, local or network drives, or ftp server. The ftp server in question is preset to the *VirusBuster* ftp site but can be adjusted for

Scanning Tests	ItW Boot		ItW File		ItW Overall		Macro		Polymorphic		Standard	
	Missed	%	Missed	%	Missed	%	Missed	%	Missed	%	Missed	%
On-Demand Tests	0	100%	118	86.4%	118	87.3%	262	94.0%	2473	77.1%	34	93.2
On-Access Tests	N/A	N/A	118	86.4%	N/A	N/A	262	94.0%	2473	77.1%	34	93.2

those organizations in which centralization might be more important than immediacy.

General Settings

The resting place for a rag-tag assortment of choices, here the user and password for network connections may be selected, as well as the *VirusBuster* temporary directory plus a couple of more aesthetic choices.

Scanning

Finally we get to the meat of the matter, and the scanning tests themselves which used the same test-sets as the July Comparative, based on the April WildList. As mentioned before, the parsing of results files was a complicated matter indeed, and thus scanning tests were eventually performed by the expedient of deleting positive infections and quarantining the files logged as suspicious.

This revealed there to be differences between some of the scanning methods, whereas others which had sounded wildly different were, in effect, absolutely identical. Some of the weaknesses seen in the *Windows 98* Comparative were also apparent here.

The first stop was the on-demand boot sector tests, which were an area of slight improvement over the *Win 98* product; slight by dint of being a rise from 96.5% to 100%. The *VirusBuster* interface for this made scanning simple and pleasant enough. On-access boot sector testing was a little more difficult, with no detections and no mention of boot sectors in the 'areas to scan' configuration, it was clear in the end that boot sector detection on-access is not yet implemented. [VirusBuster *informs us that on-access boot scanning does work if file operations occur.* VB's *boot samples contain no files, thus no viruses were found. Ed.*]

Two not entirely pleasing facts came to light during this batch of testing. Firstly, the scanner did not offer to disinfect, but rather innocently suggested 'boot sector replacement' as a panacea when a virus was detected. The statement is all very well but is akin to suggesting 'viral code removal' as a disinfection method in a file virus.

More worrying, but less attributable to *VirusBuster's* activities, were a not inconsiderable number of blue-screen crashes when accessing floppies, despite the apparent lack of on-access checking of such media.

The on-demand tests against the *VB* set provided more in the way of curiosities than surprises, with on-access following much the same overall pattern. ItW misses still included JS/Unicle and W95/Babylonia, though admittedly these are still no real threat, and the more worrying selection of *Word 8* macro viruses. This latter was noted as a problem on the *Win 98* platform and remains so here.

Polymorphics were once again the other major area of weakness while overall detection was slightly down on the recent Comparative outing. This might, however, be

False positive Tests	Clean-Set		Macro-Set		Total-Set	
	Time (s)	False Positives	Time (s)	False Warnings	Time (s)	False Alerts
Standard Scan	370	0	55	0	425	0
Full Scan	810	3	55	0	865	3
Maximum Heuristics Scan	375	16	55	4	430	20

accounted for by differing default settings as much as engine differences. The default settings for *VirusBuster* were 'selected files' (rather than all), 'fast sensitivity' and 'normal' for both heuristics settings. Increasing the selection to 'all files' made no difference to results.

Of the detected files 148 samples of ACG, nine each of WM/Junkface and four of WM/Taguchi.F were detected by heuristics. Upping the paranoia levels to the maximum on both heuristics levels added detection for five more of the remaining ACG samples. Applying a 'full' rather than 'fast' scan added positive identification for the nine W95/Navrhar VxD samples, though the 'careful' scan seemed to add nothing that the 'fast' one had not detected.

The gains from the various heuristics and extra scanning methods are real therefore, but are they sufficiently small that time taken to scan and false positives gained are in balance? For a view of this, the *VB* Clean set was used, with the default, default with full scan, and full heuristics scan being chosen for comparison with the above data. Although a moot point, given the number of misses in the various ItW sets, there were no false positives. The timings as used were without the onscreen log which had a tendency to slow matters far more than any other feature of the scanning process. It would have been preferable to have had a simple 'update log only at end of scan' option. This is a problem in many products currently on the market.

The results were interesting, and are included as a small table on the facing page. From this it is clear why, despite the improvements in detection when using the more stringent scanning settings, there are good reasons for the standard settings. The full scan, as might be expected, results in a massive slow-down and the detection of W95/Navrhar is counterbalanced by the three false positives in the standard Clean set.

More surprising, however, is the full heuristic timing, which shows a negligible increase in scanning time despite suspecting both more viruses and more innocent files. This oddity might suggest that either some efficiency could be added to the lower heuristics setting, or that the higher heuristics levels are only looking in a slightly more paranoid fashion but that the normal threshold has been set with a great deal of expertise.

For those who feel a need to know such things, the percentages achieved for the standard on-access and on-demand scans over the *VB* WildList are also included in tabular form. As has been mentioned earlier, they are a slight improvement,as expected, upon the *Windows 98* tests, though the results between platforms are not indicative of overall improvements or slippage. For this information to be more valid, we must look to future Comparatives.

In Conclusion

Reviewing *VirusBuster* brought back memories, not at all from the point of view of its being antique but from the features of the product. In days of yore, more specifically in March 1998, this reviewer inspected *Grisoft's AVG*, now a respectable regular in the *VB* Comparative Reviews. The feeling that *VirusBuster* engenders is much the same as *AVG* did on that first outing.

There are a good number of novel features available in *VirusBuster*, some of which are potentially very useful, with a definitely above average interface. This is to be contrasted with a lack of those features which are more universally demanded, notably on-access boot scanning. This is of particular note on *NT* machines, where boot sector and register infections can be vastly more damaging and difficult to remove than on a *Windows 9x* machine.

There were slight stability problems too, mostly seen in the very same area of floppy accesses, though as noted before, only circumstantial evidence points to *VirusBuster*.

As for detection, there are certainly weaknesses in the product as it stands, most notably in the areas of the *Word 8* macro viruses and the Polymorphic test-set. This seems to be a product- rather than platform-dependent shortcoming (more data will be available in the next issue's *NetWare* Comparative Review) which may be curable on the part of the macro problem by means of a serious session of database-updating by the *VirusBuster* team.

The polymorphic detection problem too seems likely to be at least partially due to a lack of data rather than engine capability (the majority of misses came on the older polymorphics), so both these areas will be of particular interest in future tests. Overall, in time-honoured fashion, one to watch, with potential for improvement.

Technical Details

Product: *VirusBuster for NT*.

Developer: *VirusBuster Kft*, 1031 Budapest, Kalászi ut 11, Hungary; Tel/Fax +61 240 1546, email mail@vbuster.hu, WWW http://www.vbuster.hu/.

Price: For one user – £37 or US$57, for 10 users – £266 or US$404, for 100 users – £1253 or US$1903.

Availability: Immediately within Hungary, fully internationalized version expected from September 2000.

Test Environment: Workstations: Three 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disks, CD-ROM and 3.5-inch floppy, all running *Windows NT*. The workstations were rebuilt from image back-ups, and the test-sets were scanned on local hard drives.

Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/Win98/200007/test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

REVIEW NEWS

Of Macs and Men

There have long been plans afoot at *Virus Bulletin* to produce a Macintosh Comparative Review – or, in fact, a Mac review of any sort – but so far these have come to naught. To understand why no previous *VB* reviewer has strayed into this area there are matters both practical and historical to consider. Since these have a bearing upon the, as you may have guessed, forthcoming (December issue) Mac mini-Comparative, they are laid out here in all their glory before the test schedule itself.

Different …

The main stumbling block to testing Mac products is the distinct lack of a *VB* Mac test-set, and the knowledge of what to place in both the main set and the various subsidiary sets. The WildList has traditionally been a preserve of IBM-compatible PCs (referred to in the following simply as PCs) only, which served as another problem, while the nature of the Mac is in itself a cause of extra concern. The WildList problem can be safely ignored for the moment but the other cannot – not only are Macs different from PCs, they are in a further twist rather different from one another.

A 68000 Mac of days gone by can theoretically run the same apps as the fastest modern PowerPC Mac, be it iMac or G3. This may seem obvious to those weaned on PC technology but the reality is more complex. Although a PC can directly trace a path through its processors back to the primeval goo, the two processors in use in Macs are sufficiently different that the same code cannot run on both.

This is side-stepped by a fancy combination of emulation and the construction of Mac executables with both sets of code enclosed within. This does mean, however, that there is a definite difference between running a scanner on an ancient and a modern Mac and indeed on executables designed for an ancient or a modern Mac.

It also begs the question whether speed tests should be performed on each file type on each machine, plus extra tests for varying mixtures of files, or whether a more pragmatic approach can be taken. For the sake of sanity, the speed tests are planned along pragmatic lines, and where test validity is not severely impacted, this has been the general theme of the test selection.

For the purposes of the test schedules planned, therefore, the existence of older Macs as a platform will be, if not ignored, swept somewhat under the carpet. Time allowing, however, the existence of varying file types will not. Pragmatism also allows for PC files to be more or less ignored as far as executables are concerned. A Mac can 'see' PC format files on media, but for the purposes of *VB* testing, this ability will probably be ignored, unless of course there is a great outcry from readers that this is a vital and interesting test without which their lives would be incomplete.

… But the Same

So far all seems well – a small number of Mac viruses, a few of the same old tests for speed and overheads, nothing to worry about. Sadly the small matter of the macro now jumps into the ointment in the style of a particularly dung-covered fly.

In their push for market share or universal ease, depending upon your viewpoint, *Microsoft* have blessed Macintoshes with their very own version of *Office*. Despite being for a different platform, however, the whole point is that it can interpret files produced on a PC, including the macros within such files. This is not a problem for earlier versions of, for example, *Word for Macintosh*, which will blithely ignore such things, but unfortunately macro viruses are now perhaps more of a problem on the Mac than Mac-specific viruses ever were.

This means that macros must be included in the *VB* Mac test-sets, though only the relevant subset. The exact choice is fraught, since some recent macro viruses – a very good example being {W32, W97M}\Beast – use PC-specific methods of infection as well as the more common PC-specific payload or subsidiary infection methods.

I've Got a Little List

The issue of a Mac WildList was mentioned earlier with a blasé comment about this no longer being a problem. Such sentiments are possible due to the *WildList Organization* once more coming to the rescue.

Due to the *WLO's* willingness to gather information on malware threats to Macintosh computing, a Mac WildList will be available by the time that the tests are planned. Any questions concerning this, or indeed offers of aid in their ongoing quest for information, should be directed to info@wildlist.org. With the format of this list as yet not totally decided, the schedules here are subject to change as information is available.

Testing Times

So, the tentative schedule is as follows, to be altered and firmed up as December approaches. The tests will be based on a *VB* Mac test-set consisting of:

- *ItW*: based upon the Mac WildList. This can be expected to consist of a majority of macro viruses, with perhaps a sprinkling of mac-specific viruses. VB 100% awards will be given to those products which detect all of these viruses both on-access and on-demand, together with no false positives. These awards will be subject to the same conditions as the extant PC VB 100% awards as described in full at http://www.virusbtn.com/100/whatis.html.
- *Macro*: a selection of macro viruses using a macro language able to be recognised and executed on a Mac. This will be based on but not identical to the existing macro set.
- *Polymorphic*: a selection of polymorphic viruses from the macro test-set.
- *Mac-specific viruses*: a selection, as complete as can be managed, of Mac-specific viruses.
- *Others*: due to the nature of the autostart worm/virus, this may be tested, as the boot sector viruses have been traditionally tested, on a PC. Other similarly 'unique' or just plain odd Mac-specific viruses might require a separate test.

Other planned tests are:

- *False Positives*: based upon a mixed bag of clean Mac files and macro-containing files. This is likely to start relatively tiny due to the effort of proving conclusively the non-virality of files.
- *Overheads*: the method is not yet finalised, though it is likely to consist of a test similar to those currently employed for the PC tests, though these err a little too much on the side of pragmatism.
- *Archive Tests*: once more a mirror of the PC tests, we hope to test the ability of the Mac scanners to detect viruses within archives created by the most popular Mac archivers. BinHex, MacBinary and Stuffit are the most likely formats to fall within *VB's* area of interest.

Over to You

Though it may seem otherwise at times, *Virus Bulletin* is primarily designed as a source of useful information for our readers. There is no point, therefore, in producing test results which fail to pique the interest of at least one reader, even less so if the test could have been rendered interesting simply by the inclusion of yet another set of statistics.

For this reason, it is at this point that we are duty bound to request your feedback on the schedule as presented, not to mention suggestions for possible areas which have not been outlined at all. Developers are, of course, welcome to add their comments, though most especially if they do not have a Macintosh product and never intend to release one. The most weight will be given to the users, for whose benefit the tests are ultimately performed.

So, any comments which you may wish to make should be forwarded to me at matthew.ham@virusbtn.com. Such areas as extra tests, tests which are considered redundant or ill-judged or relevant real-world experiences will all be given close attention.

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:

VB, 50 Sth Audubon Road, Wakefield, MA 01880, USA

Tel (781) 2139066, Fax (781) 2139067

END NOTES AND NEWS

Registration bookings are now being taken for VB2000, *Virus Bulletin's* 10th international conference, which takes place on Thursday 28 and Friday 29 September 2000 at the Hyatt Regency Grand Cypress Hotel in Orlando, Florida. Discount prices are offered to current subscribers, bona fide charitable/educational organizations and for multiple delegate bookings. For your full colour conference brochure containing programme details of the line-up of technical and corporate sessions, evening social events, product exhibition and hotel accommodation information, contact Karen Richardson; Tel +44 1235 544141, email VB2000@virusbtn.com or download a PDF copy from http://www.virusbtn.com/.

Panda Software* has recently signed a distribution agreement with *GEM, the specialist software distributor and supplier to major UK chains such as *Dixons* and *PC World*. For more details, visit the Web site http://www.pandasoftware.com/.

The 17th world conference on Computer Security, Audit and Control focuses on all aspects of e-commerce. **CompSec 2000 takes place from 1–3 November 2000 at the Queen Elizabeth II Conference Centre in Westminster, London, UK.** For details, visit the Web site http://www.elsevier.nl/locate/compsec2000 or contact Gill Heaton; Tel +44 1865 373625.

Symantec* has released *Norton Personal Firewall 2000 to protect home users against hackers and privacy invasions. It offers out-of-the-box protection for home users on the Internet regardless of skill level or the type of Internet connection. *Firewall 2000* is available for £39.99 including VAT and can be ordered from the on-line service http://www.SymantecStore.com/.

There are currently opportunities for companies wishing to exhibit at the **Windows 2000 eNTerprise Exhibition and Conference** which take place in the Grand Hall at Olympia, in London's Earls Court from 21–23 November 2000. Contact Deborah Holland for more details; Tel +44 1256 384000.

The 16th Annual Computer Security Applications Conference (ACSAC) will take place from 7–11 December 2000 in New Orleans, Louisiana, USA. Email publicity_chair@acsac.org or visit the Web site http://www.acsac.org for more information.

***Yui Kee Computing Ltd* in Hong Kong has announced it is** to co-operate with the Hong Kong Productivity Council to provide data security training – starting in August 2000 with an anti-virus course aimed at systems administrators and other technical staff. For more details contact Allan Dyer; Tel +852 2870 8555 or visit the *Yui Kee* Web site http://www.yuikee.com.hk/computer/.

The UK Security Show 2001, incorporating The IT Security Showcase, is to take place at Wembley, London from 14–15 February 2001. For details about the programme and exhibition opportunities, visit the event's Web site http://www.securityshow.com/.

***Central Command Inc* (*CCI*) has launched PerfectSupport**, an anti-virus support service available by subscription only. Available 24 hours a day for 365 days of the year, with guaranteed response times and dedicated Web reporting, it provides unlimited toll-free telephone advice from a personalised support engineer. For more details contact *CCI* in the US; Tel +1 877 994 8287 or see http://www.avp.com/.

The organisers of **iSEC Asia 2001, to be held at the Singapore International Convention and Exhibition Centre from 25–27 April 2001**, are looking for exhibitors for the event. The conference and exhibition covers IT security topics from anti-virus through encryption to biometrics and digital signatures. An early bird discount incentive runs until the end of July. For more information and a booking form contact Stella Tan; Tel +65 322 2756 or email stella@aic-asia.com.

myCIO.com*, a *Network Associates Inc* business, has launched *VirusScreen ASaP – a hosted, centrally managed, server-based solution which scans all email before it enters a customer's network. It includes *McAfee* AV technology, is monitored 24 hours a day, 7 days a week and has a Web-based reporting system for users to check in event of a virus outbreak. For details, contact *myCIO.com* in the UK; Tel +44 1753 217500 or see the Web site http://mycio.com/.

***Sophos* is to host a two-day Anti-Virus Workshop on 19 and 20 September 2000** at the organization's training suite in Abingdon, Oxfordshire, UK. On 21 September, a one-day course entitled 'Best Practice for Anti-virus' will take place at the same location. Contact Daniel Trotman for details of how to register; Tel +44 1235 559933, or email courses@sophos.com.