ISSN 0956-9979 *OCTOBER 1999*

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Anyone got a tin opener?** Our Letters page is a real can of worms this month. Readers' opinions open a packed issue, starting on p.4.

• **Business as usual:** This issue has plenty for the corporate professional with a feature on the hidden costs of virus management and a follow-up tutorial on *Office 2000* macro security. Corporate news starts on p.12.

• **Happy New Year?** Breaking with tradition, *VB* tackles the virus implications of the year 2000 date-change. The explosion of Y2K hype features on the News page and continues on p.9.

• **Design fault or glitch?** *GeCAD's* Costin Raiu discovered a few surprises recently during tests on *Windows 2000*. He shares his findings on p.8.

CONTENTS

COMMENT
The Changing Face of Security Software 2

VIRUS PREVALENCE TABLE 3

NEWS
1. Sad Spawn 3
2. 2000 Reasons to Panic 3

LETTERS 4

VIRUS ANALYSIS
AntiSocialite 6

OPINION
1. The Little Fixed Variable Constant 8
2. Minding the Millennium 9

INSIGHT
Me and Microsoft 10

CORPORATE FEATURE
Nine Tenths of the Iceberg 12

CORPORATE TUTORIAL
Office 2000 and Macro Security 14

FEATURE SERIES
Macro Viruses – Part 2 17

PRODUCT REVIEW
RAV v7.0 for Windows 98 20

END NOTES AND NEWS 24

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

COMMENT

The Changing Face of Security Software

What do the users of security software really want? They probably want to get rid of it – they would love to live in a world where security software is not needed at all. Since it *is* needed, it should be as invisible as possible. What users really want is to get rid of security problems. The less they have to worry about security, the more time they can spend doing their real work. Very few people get paid for looking after security holes.

Viruses used to be the only every-day security problem for companies and organizations. Then came the big Internet rush of the late 1990s and everything changed. Viruses are still the biggest problem, and can now go truly global in hours. However, almost every company now has to worry about network-related security problems; hack attempts to their public servers, Web page defacing, sniffing email, remote installation of BackDoor Trojans…

“… the first steps towards integrating security solutions have been taken.”

Almost all of today’s real-world security problems can be stopped with common sense and basic security software; real-time network encryption, real-time hard drive encryption and real-time anti-virus solutions. So, users should be buying and using these tools. But should they be buying them separately, from different vendors, or from a single vendor as a suite?

Analysts and industry experts disagree on the benefits of integrated security solutions. Some insist that the best security solutions are built by independent companies, each experts in their own area, be it building anti-virus protection or developing encryption solutions. Others point out that by using an integrated suite from a single vendor you gain better compatibility, manageability and performance than by mixing products made by several vendors.

All agree that a vendor cannot build a solution just by merging specialist vendors under one name and spurting out a ‘best of’ collection of their individual products.

It would be easy to think that the *Office* phenomenon could be repeated within the software security industry. When *Microsoft* released the first version of *MS Office* in 1993, the idea was revolutionary – it was taking *Word*, *Excel* and *PowerPoint* and putting them in a single box! However, this idea changed the industry overnight. It is very rare nowadays to see someone buying, say, *Microsoft Excel* – people buy *Office* instead. The competition that could not react immediately to *Microsoft’s* move just disappeared, and now *Microsoft* rules this market.

There are very few companies in the world that have world-class expertise in both anti-virus and cryptography areas. This is not surprising, as these are complex and fast-changing areas of research. More surprisingly, when looking at the global players in the anti-virus industry, very few of them are actively promoting an integrated solution, offering protection for all basic security problems within one product.

Why has the *Office* phenomenon not happened in security software? It probably just has not happened yet. Real integrated suites have not been widely available. Export restrictions limit the trade of encryption software. Managing distributed security solutions has been expensive.

However, the first steps towards integrating security solutions have been taken. In fact, this has been one of the key reasons why several well-known anti-virus companies have disappeared during the last two years: these companies were experts in anti-virus but anti-virus only; they were incapable of producing an integrated offering and had no clear growth path once standalone anti-virus products had become a commodity.

I believe that the future of security software is in centrally-managed, policy-based suites. I also believe that suite products will take over the traditional single-tasked anti-virus products. The question really is: when?

Mikko Hyppönen, Data Fellows Corp

NEWS

Sad Spawn

In August, the virus writer known variously as Dustin Cook, Raid or Casio released HLL Toadie.7800. This high level language (Asic) virus is also a direct action prepender.

Toadie is a DOS-based virus that was designed to infect both DOS and *Windows* executables (the only problem with it is that a DOS box will temporarily appear on execution of an infected *Windows* executable). The virus will not infect between 3pm and 5pm and infected files will not run between 9pm and 12pm. When an infected file is run at 17 minutes past the hour, the virus will display the message:

```
TOADiE v1.2 - Raid [SLAM] <It’s time for a
reinstall... HeHeHe>
```

As well as spreading in the ‘normal’ manner, the author tried to make the virus spread via USENET groups (in the form of a cell phone cloner and a generator for adult Web site passwords). Similar distribution of a porn list via USENET may soon get the Melissa author into trouble.

If the directory C:\MIRC exists then the virus will place a copy of itself in it, and drop a SCRIPT.INI file. If *Pegasus Mail* is installed, it will attempt to add a copy of itself as an attachment. This procedure is very temperamental and does not seem to work in the majority of cases. If the virus is spread by one of these methods it will display one of five possible messages at the DOS prompt. The messages are puerile attempts at humorous poetry.

Unfortunately, this virus is in the wild, having infected the Austrian office of a big multinational on its release ■

2000 Reasons to Panic

No pun intended but the Y2K thing beginning to bug us. In this issue (p.9) we feature a suitably short piece on how the hype has overtaken the real date-change problem.

We thought the summer silly season had passed – no such luck. *Virus Bulletin*, once again, feels compelled to name and shame as the big anti-virus players start jumping on the Y2K bandwagon.

These AV vendors are definitely on the ball. *Trend Micro* foresees with confidence the ‘annual event’ of Christmas coming around again this year. Thank goodness for *NAI’s Y2K ViruLogic*. Hopefully, the aptly named panacea will detect every single one of the 200,000 Y2K viruses that *Symantec* has spookily predicted.

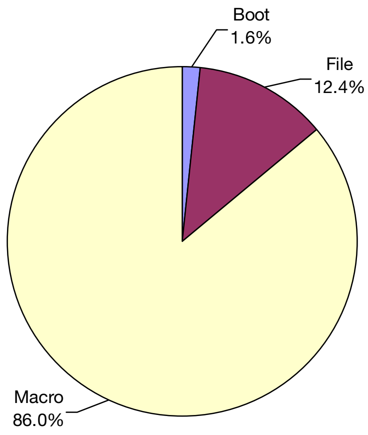
These predictions are nothing short of incitement – for virus writers to get on with producing new viruses and for customers to panic unnecessarily. The first day of January 2000 will be business as usual for viruses ■

Prevalence Table – August 1999

Virus	Type	Incidents	Reports
ColdApe	Macro	897	47.4%
Win32/Ska	File	169	8.9%
Ethan	Macro	142	7.5%
Marker	Macro	121	6.4%
Class	Macro	85	4.5%
Laroux	Macro	81	4.3%
Tristate	Macro	55	2.9%
Melissa	Macro	53	2.8%
Win95/CIH	File	35	1.9%
Story	Macro	30	1.6%
Cap	Macro	27	1.4%
Footer	Macro	26	1.4%
Temple	Macro	23	1.2%
Toadie	File	12	0.6%
Ded	Macro	10	0.5%
Parity_Boot	Boot	10	0.5%
Form	Boot	7	0.4%
Locale	Macro	7	0.4%
Win32/Kriz	File	7	0.4%
Cont	Macro	6	0.3%
Thus	Macro	6	0.3%
Groov	Macro	5	0.3%
Pri	Macro	5	0.3%
Others [1]		72	2.4%
Total		**1891**	**100%**

[1] The Prevalence Table includes a total of 72 reports across 35 other viruses. A complete summary can be found at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

The AV Road Not Travelled

What we all need is a non profit-making international association that will field virus complaints with the *specific purpose* of:

(1) tracing viruses to their originators with the help of software and Internet experts;

(2) taking strong legal action against the scum causing so much havoc internationally on the Internet; and

(3) pushing all levels of government to strengthen laws against virus originators.

I do not see this as an encouragement of government's invasion of our privacy, by the way! Virus originators serve absolutely no beneficial purpose – they simply cause severe disruptions to a very important new communications medium! Censorship is not the issue here – only the effective elimination of those damaging our vital new medium which we increasingly rely upon to carry out both business and personal matters.

As an avid email user for both work and pleasure, I receive an average of thirty emails per day, many with important file attachments. I feel that with the advent of e-commerce, this medium of communication is becoming a vital element of human interaction as we move into the 21st Century.

However, I feel strongly that this (Internet) medium is being severely degraded by the continual barrage of new viruses appearing daily on the Internet! I do not go along with the 'so be it' attitude where one simply accepts viruses as a necessary evil or inconvenience. It appears from my correspondence that most Internet users are more concerned with updating their anti-virus software than attacking the problem at its origin, i.e. eliminating the virus originators!

Would a community simply use another post office if one were threatened by bombings disrupting the mail delivery to their homes and businesses? Or would the community push for criminal prosecution of the perpetrators while increasing security around their mail facilities? How about the destruction of phone lines? Should people simply attempt to make the circuits more secure without investigating and taking action toward those found to be responsible for the malicious communication disruption?

I believe there is a threat to our safety and economic well-being when viruses are let loose on the 'Net and many hours are lost trying to constantly upgrade virus-protection programs to counter the new ones. I'm still trying to recover from an incident which occurred on 24 August.

Virus originators (I call them virators) are worse than terrorists! Terrorists strike and do damage in one specific geographical area at a time. Admittedly, bodily injury and death often accompany the disruption of people's daily interaction and the economic loss in the locale of the terrorist incident. However, virators not only cause severe disruption to communications which could involve life-threatening situations in addition to gigantic financial losses to corporations due to lost time, but the very nature of viruses make it a global incident in a very short time-frame! The mental anguish, heartache, loss of revenue, and possible health and safety implications are magnified world wide each and every time a virus is released!

Enough is enough with these malicious virators! I'm starting an action group in Houston, Texas, to solicit support from corporations and individuals in establishing an organization to carry out what needs to be done!

Glenn W Rossi
International Consulting Agent
Greater Houston, USA

Playing Safe

In last month's issue, Dr Muttik, in his informative piece on macro viruses, stated that the easiest way to clear the Template Bit was to go through the procedure that he outlined. I would disagree with the procedure on the grounds that it is necessary to keep the suspicious file open and Shortcut keys could be subverted.

My preferred method of clearing the Template Bit or Macro Warning Box is to open up a New Document and InsertFile Oldfilename and then FileSaveAs Newfilename. This procedure is equivalent to the one Dr Muttik outlined but with fewer potential hazards.

Paul Baccas
Technical Support
Sophos Plc, UK

Taking Issue^2

In my August 1999 letter I detailed the risk to corporations from computer viruses in files with non-standard extensions (See *VB*, August 1999, p.4). In last month's issue, Nick FitzGerald of *Computer Virus Consulting Ltd* responded to my letter, but sadly I feel that he missed the main point.

First, the primary purpose of my original letter was to discuss not anti-virus technology but corporate policy. The letter was meant to warn corporate customers that their existing scanning policies will probably not be effective against future viruses and Worms. As Mr FitzGerald correctly points out, some (most?) anti-virus products have the capability to scan files selectively, based on their content. However, I would argue that the vast majority of corporations still rely upon extension lists to improve scanning performance – especially at the gateway and on servers which cannot afford to suffer from performance degradation by anti-virus software. By scanning only those files with common extensions (e.g. .XLS, .DOC, .EXE, .COM), these corporations put themselves at risk.

Second, as stated in my initial letter, *SARC* is working to optimize performance of the *Norton AntiVirus* range to scan all files more efficiently. As Mr FitzGerald points out, some anti-virus products, including ours, already perform 'intelligent typing' to allow the engine to determine which files to scan for which viruses. We are seeking to improve *NAV's* performance further, and it's good to hear that other anti-virus products are doing (or have done) the same thing.

It's important to note that in the last few *Virus Bulletin* product reviews *Norton AntiVirus* was configured to scan all files; yet it incurred overhead comparable to most of the other reviewed products. Given that *Symantec* customers have asked for even more efficient scanning of all files, it does not strike me as outrageous that users of other products have similar concerns – whether these products have implemented 'intelligent typing' or not.

In summary, corporate customers need to give serious consideration to new policies which scan all files – with the highest priority at the WWW gateway, the email gateway, and the email server. Anti-virus product developers should continue to improve their scanning performance to limit the overhead impact of these products and make such policies feasible. Thanks for this forum; it is great to see that the *Virus Bulletin* letters section has started to generate some passion and will hopefully have a positive effect on both users and vendors alike.

Carey Nachenberg
Chief Researcher, Symantec AntiVirus Research Center
USA

Taking Issue[3]

The point of Nachenberg and FitzGerald's exchange is that AV products for *Microsoft* operating systems no longer have the luxury of relying on the file extension. This is true, but they both neglect to mention that AV scanners never did have this luxury. It's always been possible to use *Word* to open files without a .DOC extension. It's always been possible to use *Notepad* to open files without a .TXT extension. Scanners merely exploited the fact that it was conventional to use the 'correct' extension.

It is also worth pointing out that, if one ignores the *Internet Explorer*-related problems for a moment, the filename extension is becoming more and more embedded into *Microsoft's* OSes. A default installation of any *Windows* variant, for example, will result in an *Explorer* that hides the extension from you. You can reveal the extensions by tweaking *Explorer's* options, but most people never do that. Assuming that you never use a command prompt (most people don't), and you don't change this *Explorer* option (most people don't), you'll never see an extension. It could be argued, from this point of view, that reliance on extensions is, in fact, safer than it has ever been.

As both point out, this is undesirable, due to the behaviour of certain Internet browsers and mail tools. FitzGerald suggests adding the capability to update a scanner's extension list automatically (the list of file extensions which will cause it to check the file) as new signatures are downloaded and installed. At least one product on the market can already do this. However, it doesn't help. In Nachenberg's example, '.I' would have to be added as an extension to check. It's not a design solution, it's a kludge. The next virus will simply not use an extension, or use random extensions, and then FitzGerald is stuck again.

The only design solution is to not rely on the filename as any indication of what is inside – i.e. perform type identification on all files. The downside is that it really doesn't matter how computationally cheap the type identification process is. A huge overhead will be introduced merely by opening every file on the disk.

Users can try this for themselves. Run your scanner across your hard disk in default mode five times, and time it (don't run anything else at the same time!). Discard the time for the first run (this is a primitive attempt to eliminate cache effects), and take an average of the other four. Now, change your scanner configuration so that it scans all files. Don't change anything else on your scanner or your system. Now, do the five scans again, and again, take the average of the last four. I guarantee that the second average is much higher than the first.

This is a valid comparison because, internally, scanners always do type identification – all scanners I am familiar with. If it opens a file called 'RANDOM.DOC', it won't simply assume it's a *Word* document, it will perform type identification. Similarly, if it is told to open a file called 'RANDOM.TXT', it will perform type identification.

The second average is so much higher than the first average that it's not going to be possible to make up that speed by optimizations elsewhere in the product (in fact, in some cases, writing a program that opens every file on the disk, reads one byte, and then closes the file, will be slower than your scanner when it uses its default extension list). Users will have to live with a much slower product. And it won't be a gradual decrease in speed – there will be a point when the manufacturer decides to scan all files, and at that point, the scanner will become very much slower.

Security versus usability. No surprises there! AV people are pushing the envelope in terms of losing security in order to keep up the usability of their product – in a competitive marketplace that cares mainly about usability, the first company to scan everything will lose.

Anon
USA

VIRUS ANALYSIS

AntiSocialite

Nick FitzGerald
Computer Virus Consulting Ltd

Virus writers are usually on the lookout for new ‘tricks’ to make their viruses more difficult to detect. As has been mentioned in these pages before, they often add features to viruses on a new platform in much the same order as they appeared on ‘already conquered’ platforms.

Early DOS viruses were simple self-replicating programs. As virus scanners came to detect them readily, encryption, and later polymorphism, was introduced. Both aimed to make the scanner writer’s work more difficult. After several years of simple macro virus development, it should not be surprising that encrypted ones are now appearing. The members of the W97M/AntiSocial family are amongst the first true encrypted VBA macro viruses. The .E variant has recently enjoyed some media coverage, although most of that has been under the alias the virus’ writer wished it to go by – Sixtieth Skeptic.

Why Encryption?

Encryption alone does not add much height to the hurdles of virus detection. A fundamental problem for encrypted programs is that, for the encrypted code to be executed, it has first to be decrypted. Thus, the decryption code has to be part of the virus, and as a piece of constant code, it can be detected using straightforward scanning techniques.

Oligomorphism is the next step up from simple encryption. Here, one of ‘several’ pre-programmed decryptors is chosen during infection, and dropped into the decryption ‘slot’ in the virus. This also makes a virus no more difficult for a scanner to detect, so long as the scanner provides for multiple definitions of the same virus. It increases the virus’ complexity, and can substantially increase its size.

Polymorphic encrypted viruses take the process to its logical conclusion, producing polymorphic decryption code. This ensures many decryptors are possible, and requires more advanced detection techniques, such as emulation and tracing.

Encrypted Macro Viruses

Probably the first polymorphic encrypted macro virus was WM/Dakota.A. It is fiendishly slow and cumbersome, as WordBasic’s string manipulations are not implemented as efficiently as those in VBA. Encryption is thus more likely to be viable in VBA. The first encrypted VBA virus was W97M/Walker.D, discovered early in December 1998. It was closely followed by both .E and .F variants and {W97M,X97M}/HalfCross.A.

These VBA viruses implement a trivial form of encryption. Logically XORing a value (the ‘plaintext’) with another (the ‘key’) is simply reversed by XORing the result (the ‘cyphertext’) with the key. These viruses encrypt their replication and, where applicable, payload routines. Those routines are decrypted before they run, then re-encrypted. There are other tricks to these viruses, but if their writers thought their use of encryption would make things difficult for anti-virus developers, they were sorely wrong.

Since these viruses use a constant key or a pattern of keys always starting at the same value, their replicants look the same, generation after generation. Their writers missed the point of encrypted viruses. Scanners that detect based on the source code or compiled (p-code) of the macros, were easily updated to detect these viruses.

Encrypting their ‘bad’ code may have provided them with one small benefit. With their ‘virus like’ code encrypted, they would have avoided most heuristic analysers because, as yet, few shipping products have VBA emulators. Walker.E and .F have reached the top half of the WildList. It is difficult to say how much that is due to their encryption. There are many more VBA viruses on the list that are not encrypted. ‘Catching a lucky break’ plays a major role in becoming sufficiently well distributed to make the list.

AntiSocial Developments

W97M/AntiSocial is a recent family of class infectors, comprising six variants. Family members follow the same basic design. Each consists of a Document_Open event handler in the ThisDocument object of its host, followed by either a series of comment lines or a second sub-routine that is a series of comment lines – the crypted virus code.

The readme file released with AntiSocial.A suggests the virus’ writer believes the techniques used result in polymorphic encryption. In part, the claim reads:

```
* Antisocial Encrypts Each   *
* Line Of Code With A          *
* Different String Each Time   *
* The Virus Is Executed. […]
```

So, was AntiSocial.A the first polymorphic encrypting VBA virus?

Running an Infected Document

Allowing macros to run when opening an infected file causes the Document_Open handler to run. Its code immediately disables Ctrl-Break checking, preventing the user from aborting the macro. Although they decrypt fairly quickly, there were slight but noticeable delays while the virus ran under *Word 97* on a 400 MHz *Celeron* test machine. Wary users may try breaking the virus’ run.

Next is the decryption loop. In the .A through .E variants, a constant key is added to the value of each character in the plaintext. To decrypt, the key is retrieved from the second character position on each cyphertext line, then subtracted from each subsequent character on the line. The leading single quote and the key are dropped from the cyphertext lines when decrypting. AntiSocial.F has almost identical code, but uses the XOR function, rather than subtraction.

More substantial differences between the variants appear once their code is fully decrypted. The .A through .D variants ‘wait’ until the now-decrypted Document_Close macro runs. However, the .E and .F variants force the issue, calling the newly-decrypted macros – Sixtieth_Skeptic and ViewVBCode respectively. Apart from the encryption step, these routines implement fairly standard class infectors.

First, they perform the usual configuration changes seen in Word viruses, disabling ‘prompt to save normal template’, ‘confirm conversion at open’ and ‘macro virus protection’ options. Next, they encrypt their own code, using the same approach as in the decryptor but using the complementary logical operator to that in the decryptor. The important difference between AntiSocial and the earlier ‘encrypted’ VBA viruses, is that this encryptor generates a random number at the beginning of each line. This has a value from one to eight and is used as the key for that line. The cyphertext version of each line is built up in a string variable, prepending a single quote character and the key for that line. Finally, the VBA ReplaceLine method is used to substitute the cyphertext line for the plaintext one.

Once it has re-encrypted its code, it uses the DeleteLines method to remove all macro code from the ThisDocument object of the active document. It then replicates by copying its newly encrypted source code to that location, and repeats this process for the normal template.

The Obligatory Bugs

The encryption described above is flawed. More accurately, its implementation is flawed. The encryptor does not seed the random number generator, so its calls to the VBA Rnd function produce a predictable key sequence. The writer of this virus must be quite unobservant, or unable to test the code. Any modest testing should have quickly established that all replicants within a generation always produced the same ‘random’ key sequence.

Investigating this, it became clear that the ‘problem’ has two causes. First, as noted above, the virus writer does not call the VBA Randomize function anywhere inside the encryption algorithm, but that alone should not account for every replicant within a generation having the same ‘random’ key sequence. The second cause appears to be a feature of the Visual Basic environment. When routines within a running module (perhaps even anywhere within the whole Visual Basic environment) are modified, much of the environment is reset to its startup state, including the initial, fixed seed value for the Rnd function.

As the virus rewrites parts of its code while encrypting and decrypting, Visual Basic recompiles the modules and resets its internal variables. Because the encryption code is the only code calling the Rnd function, and that occurs after the environment reset, the seed for Rnd is always reset just before it is called a fixed number of times. The result is the same key sequence is generated time and time again.

Payloads

All these AntiSocial variants remove existing code from the ThisDocument object of their hosts, so may delete user macros. Variants .C, .D and .F have no further payload.

If infecting in the 59th minute of any hour, the .A variant replaces all text in the host with ‘Antisocial…’ [sic] then saves the document. On the first day of any month, the .B variant attempts to delete the contents of drive C: – the method used will fail. Melissa-like mailing of an infected file to 60 addresses from the victim’s address books is attempted by the .E variant . This variant also creates a source listing in C:\SS.BAS and sets C:\SS.VBS to inject that into the normal template at each system restart/login.

Closing Comments

Truly polymorphic encrypted VBA viruses are probably not very far away now. Perhaps the anomaly discussed above will be noticed and fixed before this analysis gets to print? I hope not, but if not that virus writer, it seems likely that someone will, and soon.

Interestingly, some forms of polymorphism appeared in macro viruses before most of the encryption efforts. The nature of WordBasic and VBA accounts for this, providing rich text-processing environments. Macros can obtain and modify their source and are guaranteed access to the development tools necessary to recompile themselves. Early macro polymorphism involved such things as random identifier replacement. Coding that is simpler than the most trivial encrypted virus. So, although the general trend from simpler to larger and more complex is being followed with macro viruses, the order of the developments differ.

W97M/AntiSocial

Aliases: There are many aliases for members of this family. The .E variant recently received coverage as Sixtieth Skeptic, W97M/Skeptik.A and W97M/Sskeptic.

Self-recognition:
None – it deletes existing code in the ThisDocument object of its hosts.

Payload: See text.

Removal instructions:
It is best to use current anti-virus software, but the steps described in *VB*, April 1999, pp.17–18 can be followed.

OPINION 1

The Little Fixed Variable Constant

Costin Raiu
GeCAD srl

With the official release date of *Windows 2000* approaching practically at warp speed, many developers are digging into *Microsoft's* upcoming flagship operating system to see how its features can impact with the current AV technology. A few weeks ago, after installing the newly DVD-shaped MSDN library on my work computer, I decided to take a quick glance at the new features in the *Win2K* API.

I started with the functions used by *Office* applications to work with documents. While taking a look at the old StgCreateDocfile function, I noticed a small note in the help page which read: 'The StgCreateDocfile is obsolete for Microsoft(r) Windows(r) 2000 systems. The function still exists for compiling pre-Windows 2000 systems. New applications should use the StgCreateStorageEx function.'

Aha! So *Microsoft* implemented a new function to work with so called 'OLE2' files, namely 'StgCreateStorageEx'. Until now, we had to call StgOpenStorage to open a storage, but we had to use StgCreateDocfile to create one.

Since '*Ex' functions usually have extra parameters, I decided to check the API definition from MSDN:

```
WINOLEAPI StgCreateStorageEx(
const WCHAR * pwcsName,
DWORD grfMode,
STGFMT stgfmt,
DWORD grfAttrs,
STGOPTIONS * ppStgOptions,
void * reserved2,
REFIID riid,
void ** ppObjectOpen,
);
```

The trained eye will spot the 'stgfmt', 'ppStgOptions' and 'riid' parameters immediately – these were not present in the old version of this call. Checking the parameters further, I went to 'ppStgOptions' for help. A quick note in the MSDN help says: 'This parameter may be NULL which creates a storage object with a default sector size of 512 bytes. If non-NULL, the ulSectorSize field must be set to either 512 or 4096.'

Uh, oh – I read this twice to be sure I got it right. Indeed, it seems that *Win2K's* OLE32.DLL now allows the creation of document files with sector size larger than 512. If you are familiar with the native OLE2 file format, you should know that until now, only 512-byte sectors were created for any OLE2 compound file. However, in *Win2K*, 4096 is also a valid sector size.

OLE2 files have the sector size specified in the header, at offset 0x1e, as a two-byte WORD. This WORD is 09 00 (low endian) for 512-byte documents (2^9=512), but for 4 KB files should be 0C 00. (2^12=4096). I was very curious to verify this claim. I installed *Win2K* on a test machine, as well as *VC++ 6.0* and *Office 2000*, in order to be able to compile test programs and check if *Office 2000* would be able to create such 4 KB sectored files itself.

A short test program proved the first hypothesis to be true. The *Win2K* API allows the creation of such files, and I even created two test documents with 4 KB clusters. [*The respective documents are available for download from the VB Web site, http://www.virusbtn.com/ole4k/. Ed.*]

The second step was to check if *Office 2000* is able to create such files by itself. Thus far, I could not convince it to create 4 KB OLE2 files, but on the other hand, *Office 2000* had no problem loading a 4 KB sectored file. Furthermore, it had no problem loading a macro virus stored in a 4 KB file, which infected the system instantly. However, what I found out was that after saving the file back to disk using either FileSave or FileSaveAs, the 4 KB sector file turned into a standard 512-byte sector file. This is good news.

The bad news is that if you have a virus in a 4 KB sector file, most anti-virus programs will be unable to detect it. That is because back when macro antivirus engines were designed, 512-byte sector files were the only possible case, so many of the engine designers had their parsing routines written and tested only on 512-byte sectors. A quick test against a virus stored in a 4 KB cluster with some of the most common anti-virus products went very badly. Out of 15 products tested, two crashed on such files, 12 failed to detect anything, and only one (!) product managed to parse the 4 KB file correctly and detect the virus.

Suggestions

The AV community must not overlook this issue. Most of the large AV vendors were already informed about this problem, and I have information that at least some of them will fix their engines soon. The fact that such 4 KB sector files cannot yet be created by standard *Office* applications means we are not yet directly at risk – on the other hand, a Worm stored in a document with 4 KB clusters might be able to infect computers around the world in a matter of days (the Melissa incident comes to mind) and stopping the infection will require updates to our engines. If we fix this issue in time, such problems will hopefully never become reality. How many other similar problems still lie buried deep in *Windows 2000*? I think this would be the right time to find out, or those paying the price for our lack of care will be our users.

OPINION 2

Minding the Millennium

Graham Cluley
Sophos Plc

[*Traditionally, VB has steered clear of Y2K issues. These answers to frequently asked questions regarding the effect of the Year 2000 on viruses may help to explain why. Ed.*]

Only a few months to go now. Dark clouds are brewing, Armageddon is approaching, and the four horsemen of the apocalypse are stocking up on carrots for Dobbin. You must have noticed. The warnings on television, in the newspapers, on the radio. The doom-mongers predicting riots in the streets. Y2K is almost here!

Is Y2K an anti-virus issue?

Well, yes and no. The Y2K problem is not a virus, it is a bug. Any system that contains date-related functions may be susceptible to problems as we enter the year 2000. The problem is caused if software stores the date as two digits (‘00’) rather than four (‘2000’).

One of the problems is that computers are no longer solely used by nerdy academics wearing sandals – it is possible to use computers with very little technical knowledge today. Trying to explain the Y2K problem to your Auntie Ethel on *AOL* can be quite a challenge.

So what does Y2K mean as far as viruses and anti-virus software are concerned? Firstly, Y2K is a great opportunity for corporate organizations to check how well they have rolled out their anti-virus software. Many companies have been visiting the desktops in their corporation checking for Millennium compliance. This opportunity can also be used to see whether the computers are running any anti-virus software and, of course, whether it is properly installed and, more importantly, up to date.

You should also determine whether your anti-virus software is Y2K-compliant. Your anti-virus vendor may not have placed information on their Web site, or you may need to ask them for a written statement. Remember that it may not just be the main scanner which has Y2K-related issues, but also the administration and scheduling tools.

Will viruses trigger on 1/1/2000?

Of course, there are viruses whose payloads trigger every day of the year. The first day of January is no different. There will almost certainly be viruses written to deliberately trigger on New Year’s Day, but their threat is no greater than any of the other 45,000 viruses in existence. In fact, perhaps it is lower because of the small number of people who will be at work on that day.

It is unlikely that there will be a flood of brand new viruses on 1 January 2000. Remember that viruses typically take some time to spread. Even the fastest spreading viruses like Melissa require a human element to help them on their way (users opening email, double-clicking on the attached document). Since most users will not be at work on that day, even an email-aware virus is unlikely to spread far.

How can viruses exploit Y2K?

Viruses may try to exploit the turn of the century as a means of spreading themselves. For example, remember there were viruses which joined in the 1999 New Year celebrations (Win32/Ska, also known as Happy99)? It is inevitable that some viruses will attempt to disguise themselves in programs, presenting themselves as New Year 2000 celebrations (in the form of screensavers, electronic greetings cards, etc).

It is all too easy to imagine. Your users receive an email telling them they have a chance to win a holiday in New Zealand to see in the next Millennium – just double-click on the attached document…

Viruses may also try to exploit the confusion surrounding the whole issue of Y2K. You can be certain that come the new year all computer problems will be blamed on the year 2000 bug – even if they have no connection with it at all. So, a virus might create confusion by displaying a Y2K-orientated ‘error message’. For instance, a virus may display a dialog box saying ‘Program found not to be Y2K compliant. Process halted.’. This has the potential to create a considerable amount of confusion (especially in the more paranoid organizations).

Will AV companies protect you over the Y2K period?

Certainly, the company I work for is planning to have a support team available as usual, 24 hours a day, with the ability to add protection against new viruses if the need should arise. I imagine other anti-virus companies are taking similar steps to reassure their customer base.

It is important to remember that the Y2K issue is just a bug present in some software systems. Unfortunately, the difficulty in determining which systems the bug may be present in has made the problem a considerable one for industry to handle effectively.

Ironically, some viruses may themselves be affected by Y2K problems. As we know, many virus authors are less than concerned with code quality, and there are still many viruses which remain in the wild from the early 1990s (before Y2K became a pressing issue). It seems inevitable that some viruses will stop working as originally planned come the next Millennium.

INSIGHT

Me and Microsoft

[*Randy Abrams started working for* Microsoft *in 1993 and brought his passion for anti-virus with him. For the past two years he has been trying to persuade his bosses to let him devote himself to* Microsoft's *AV needs full-time. He shares how he started in Electronics and how, having sworn never to work with computers, he is now addicted! Ed.*]

I was born in 1960 in Poway, California, a suburb of San Diego. I grew up there and in San Bernardino – home to *McDonald's*, the Hell's Angels, and the left-hand turn lane. In 1978 I went to *Antioch University* in Yellow Springs, Ohio but, distracted by the social science of partying, I did not finish my Bachelors degree; several years later I completed an Associates degree in electronics.

After nine months of dating, six years of a long distance friendship, a reunion in Seattle which resulted in another nine years of dating, my wife, Carol and I were married in 1995. We keep busy trying to do a good job of raising our two cats – Ghost and Mrs Mewer. I play the piano and occasionally write music. In fact, I wrote a song for Carol that the band at our wedding played. We share a mutual enjoyment of travel and dining out.

Bitten by the Bug

While studying electronics I proclaimed that I would not work on computers because they were digital and analog was where the challenge was. After my first and only quarter of assembly language programming I added that computers might side-track me from electronics.

Around 1990 I bought my first computer. I was employed by a small electronics company where I had an arrangement with the management whereby I left my computer at work and they were free to use it. I spent a fair amount of time downloading freeware and shareware programs and trying to figure out how they worked.

One day, a programmer told me I should get rid of the Vshield TSR as it only hogged memory and I did not need it. Later, he asked if he could use my program to compile some software because his computer was acting 'strange'. To cut a long story short, he introduced my PC to the Form virus. I never again listened to a programmer who advised that I should decrease my security!

A short time later I obtained a sample of the Sunday virus from a neighbour who had an infected game. I would physically disconnect my hard drive from my PC and then boot into a configuration with a 3 MB RAM disk and test to see if the virus would replicate, and to see if my anti-virus software could detect the virus in compressed files.

All Change

I left my first electronics job at a company called *BASCO* for *Microsoft* for two reasons. First, I needed some dental benefits and *BASCO* was too small to afford it. Secondly, I wanted to spend more time on computers. In 1993, I joined *Microsoft* as a technician working on floppy disk duplication equipment. It was at this time that *Microsoft* opted not to renew their *McAfee* licence but to use *MSAV*. I notified my managers that while *MSAV* was adequate for home use, it certainly was not for industrial use. I was given the additional responsibility of choosing an anti-virus product and deploying it (and maintaining it) across the 30 or so production PCs on the duplication floor.

A year after I was hired I transferred to the Redmond Software Release Lab. *Microsoft's* software release labs are conduits between the product groups and manufacturing. In addition to copying masters for manufacturing, the release labs would perform a simple scan of the media. When I joined I noticed that the lab was using the same software that the rest of *Microsoft* was using (*F-PROT* at that time). I recommended that we get a different package to maximize our detection odds. It was to be my decision.

All was calm on the Redmond front until 1997. At that point my manager came to me and asked how we could make sure that no virus ever got out in a product that was released from our labs. I actually argued against us doing anything. Content has always been the domain of the product groups and I felt that they should take on this responsibility. My manager nodded and repeated the question. Realizing I wasn't going to 'win' this one with logic I appealed to fiscal responsibility.

I told him we could not be 100% sure, but if we wanted to attempt to, it was going to add at least four hours to our release process (an ugly proposition) and it was going to be expensive. We would need several multi-user licences for anti-virus software, several new PCs, and some development time to automate decompression. He bought it all and I was given the additional responsibility of making this all work, and the success or failure is tightly linked to me.

VB'97 was a turning point for me. I learned members of the anti-virus industry felt that they could do a better job if we at *Microsoft* would provide more information about our products for them. As I depend upon the AV industry to prevent infected code from leaving our lab, I recommended to my management that *Microsoft* comply. I have been given too much credit for our liaison with the AV industry. When I was given the support to go out and try to make it happen I was delighted to find that Daryl Pecelj, our corporate AV program manager had already created processes, in conjunction with the *International Computer Security Association* (*ICSA*) to facilitate this.

The product groups at *Microsoft* are also very helpful as they rarely submit an infected file for us. I look at the labs as kind of a hockey, or soccer game. The product groups are our defensive players. It is their job to make sure that there are no shots-on-goal. The anti-virus software is our goalie. No matter how good your goalie is, given enough shots-on-goal, one will get by.

My current job title is 'Release Technical Specialist'. I am responsible for the anti-virus product selection and procedures used in the labs where most of *Microsoft's* software is released. All retail and OEM products go through labs I set procedures for. Most of the programs, such as MSDN, Technet, and Select come through my processes. The code on Microsoft.com that has *Microsoft's* digital signature on it comes through one of my labs as well.

I also train release employees and have performed anti-virus presentations for internal training, a *Windows 98* users' group and some major *Microsoft* customers at Redmond. I fill in as a backup for fielding reports of viruses from users. My management agrees that it is very important to maintain a good relationship with the anti-virus industry. Part of my job is to maintain and enhance our relationships within the anti-virus industry.

My second main responsibility is to assess the impact of new technologies on the release labs. Releasing software was my primary responsibility for several years. I have more experience in the Redmond Release Lab than anyone else at *Microsoft*. This gives me a comprehensive understanding of how technologies such as CD-ROM copy-protection schemes are going to impact the labs.

For two years I have been trying to develop my position into full-time anti-virus. I have been getting closer to that, but I think my release experience will continue to require me to spend time assisting on that front as well.

Onwards and Upwards?

I would caution readers to not place too much confidence on my industry prognostications. I have often said that I believe it takes a great deal of knowledge to be an anti-virus expert, but only a little to be perceived as one. I think we will continue to see more viruses and Trojans aimed at fast spreading and destruction. This will probably lead to a home market for an immune system, but I think corporations will probably be a bit slower to adopt such a solution.

I distinguish between the people who write viruses that never leave their PCs, those who write viruses that are only sent to anti-virus companies, and those who write viruses and either carelessly or deliberately allow them to spread. I do not believe the first two classes are going to see a different future from the rest of us. It is the third that scares me. It is not fear of them stealing or destroying my data, it is the social consequence of their actions. As more people get frustrated at these writers the public mood will sway in the direction of allowing our government to restrict our freedoms and become significantly more intrusive.

We will see stronger legal penalties for virus writers, but I fear we will all pay the price if government intrusion and restrictions on our freedom increase. These rebel wannabes are among the most significant proponents of any government/intelligence organization seeking more legal access to our personal information. Those who write viruses that do not deliberately destroy data are not going to be viewed any differently by people calling for government action than those who write destructive viruses.

I am one of the few users who actually like false positives – not signature-based false positives, but heuristic false ones. In a recent training presentation I distinguished between 'legitimate' false positives and 'just plain mistake' false positives. The former occurs when there is residue from disinfection, or a program does enough suspicious things to justify heuristics getting alarmed. The latter usually occurs when a scanner just plain uses a bad search string. A DLL in *Word 97* contains the string 'Copyright Bandung Indonesia. One scanner indicates the file is infected with the *Word* Macro virus Bandung. This is an example of a 'just plain mistake' false positive. The nature of the release labs is such that I must rely upon detection. If I were installing products I would probably use some generic methods too.

I look forward to continuing to work with the anti-virus industry. It has been very pleasing to me to see some positive changes at *Microsoft*, and a growing interest in AV. This has been visible in terms of people requesting information and assistance in improving their processes, as well as in product groups looking to work more closely with anti-virus companies. I will continue to make myself available to the anti-virus industry, and I am sure my friends in the anti-virus industry know that I will continue to report the strange occurrences that arise from time to time when you install several anti-virus products on a single PC and then scan 100 million files!

CORPORATE FEATURE

Nine Tenths of the Iceberg

David Harley
Imperial Cancer Research Fund

How effectively does anti-virus software limit virus damage? Damage from virus and Trojan payloads is frequently discussed (if not always well understood), but many ‘successful’ viruses do not have a payload. What about the damage caused simply by the infective process? What about secondary (especially psychological/social) damage and anti-malware cost-of-ownership issues? This analysis presents a formal damage model and notes some of the hidden costs of proactive/reactive measures.

Most primary damage from malicious software occurs before and after installation (pre-infective or post-infective, in the case of viral malware). Damage during the actual installation may seem likelier in the case of malicious software than it is in the case of legitimate software (quality assurance is not, in general, what malware authors do best).

In fact, such damage consists mostly of the transient freezes most of us are resigned to accepting as characteristic of modern applications and operating systems. There are notable exceptions, such as the trashing of legitimate macros by WM/CAP as part of the infection process, or unintended damage due to factors unanticipated by the author. None-the-less, ‘successful’ malware is generally associated with damage caused by its presence once installed, rather than with the installation process. Indeed, non-viral installation often consists simply of file copying. (I have preferred to talk about installation rather than infection here, so as not to exclude non-viral malware.)

Pre-Installation Damage

Detection at the point of entry, prior to the execution of malicious code, is the best case in terms of damage prevention. However, it is not cost-free. From the consumer point of view, the ultimate target is malware detected and removed with complete transparency. Increasingly, vendors are offering us solutions modelled on quasi-biological immune mechanisms that are claimed to optimize automated response and thus increase transparency. Even if these solutions could be assumed to live up to their promise, it would be wise to assume significant implementation and maintenance overheads.

Consider, though, the uncertainties of transparent detection at point of entry in real life. Memory-resident known-virus scanners do a reasonable job of detecting and in some cases automatically disinfecting known viruses. However, recent ’Net-borne threats have demonstrated that malware may spread far and wide and cause significant damage within hours of being introduced into the wild.

Of course, a vendor may have a fix available almost as quickly, though there is many a slip between AV researcher’s lab and customer’s desktop. The malware’s impact on the network may prevent the malware manager’s pulling the fix from the vendor’s site, or the vendor from distributing it using push technology. Scanners that use advanced heuristic analysis do better at detecting unknown viruses or variants, but increase the risk of false positives. It only takes one potential false positive to destroy the illusion of transparency, except possibly in the most draconian of environments, where the scanner is assumed to be infallible and all suspicious files are discarded.

Sadly, this is an unsafe assumption. Not only do scanners miss real viruses and identify viruses which are not there, they detect a number of other objects (intended viruses, Worms, Trojan Horses, remote access tools, even jokes). Dealing with such a range of ‘attacks’ is difficult to automate fully, even with informed preparation and sensible policies, and such measures do not come out of the box with the installation CD. Should I disinfect or discard an infected file? Discarding a Trojan file is a no-brainer, but what about a fluffy joke screensaver passed on by the marketing manager?

Automation is not the straightforward issue we are led to believe. If we trust a scanner to disinfect a virus automatically at the point of entry, we may save a technician’s time and the recipient from panicking, but may also miss blocking a potential loophole.

If an organization sends me several Ethan-infected documents, that tells me something significant about their anti-virus arrangements. If the problem persists after I have advised them that they have a problem, it tells me something more about their level of security awareness, but also about the nature of our business relationship. If they react ‘appropriately’, on the other hand, our business relationship may improve, and the raising of their awareness contributes to a general improvement in the universal virus problem. Of course, auto-disinfection does not stop me tracking logs and taking appropriate action, but that is not quite automation as presented by the auto-immunity faction.

The cost of a potential attack forestalled at the point of entry is not usually taken into account when an organization tries to balance cost of implementation against perceived risk. In fact, researchers often assume that such an attack entails no cost, but the cost of implementing defences against such attacks is readily quantifiable in terms of software unit and update costs, human resources etc. It also attracts an incident management cost: tracing an incoming threat to its source, advising that source, incident logging, reassuring and advising the owner of the system on which the incident occurred.

Both real and imagined viruses (the latter including those described in hoax alerts) can also have psychological/social consequences. Determining the potential impact of a perceived threat can be a serious drain, not only on the security manager, but on first and second-line support staff, management, and users/clients.

Post-Installation Damage

This falls into two main classes: firstly, impact of installation/infection on the computing environment (changes introduced by the mere presence of the malicious software); secondly, damage caused by the delivery of the payload.

Modification of the environment: All viruses cause minor damage in some sense, in that they modify the environment so as to install their own replicative code and conceal their presence (stealthing). They may corrupt, modify or displace system files and system areas such as the DOS and MBR, or the File Allocation Table.

Macro viruses routinely modify the functionality of *Word* so that menu options such as Tools/Macro are no longer available and *Word's* own macro detection is disabled. Some boot sector viruses modify CMOS settings so as to compromise the system's ability to clean boot. In many cases, the effects of these changes are not perceptible in the absence of anti-virus software.

Types of modification and possible consequences are wide-ranging in their scope and impact. They include the theft of main memory, impacting on functionality/performance so that some code may no longer run and the theft of disk space (reduced functionality/performance: some code no longer runs; data, application files or system areas partly or totally over-written; infected files no longer function properly). Theft of clock cycles means slower processes; time-critical processes are unpredictable; resource-intensive software loses functionality/performance.

General incompatibility/de-stabilisation issues can manifest themselves in several ways. System software/applications/utilities display unpredictable behaviour due to conflicts with unauthorised memory-resident software. Symptoms include protection errors, parity errors, performance degradation, loss of access to volumes normally mounted and unavailability of data or applications.

Direct damage from virus and malware payloads ranges across all three of the security areas associated with the classic tripod security model.

Attacks on availability:

Renaming, deletion and/or overwriting of files and sub-directories

Encryption of files, disks or system areas

Unauthorised calls to potentially destructive system software and other forms of disk trashing

Attacks on integrity:

Corruption and displacement of system files and system areas

Data diddling – intentional modification of targeted data files

Corruption of application files and data files by unauthorised file writes

Attacks on confidentiality:

Capturing and forwarding passwords, PGP key rings etc

Forwarding personal/confidential files to newsgroups, email addresses abstracted from on-disk address books, ftp sites etc.

Secondary Damage

This includes primary damage, but cascaded to other systems by secondary infection. However, it can also include damage caused by inappropriate response to a perceived threat: unnecessary scrapping of media, systems or system components, unnecessary re-formatting, and inappropriate use of disk recovery utilities.

Other forms of indirect damage include psychological/social factors such as damage to morale/self-confidence; scapegoating; loss of business confidence and consequent loss of competitive edge; and, of course, bad publicity.

Any of these forms of damage may entail financial damage such as litigation costs, punitive response to non-compliance with policies, standards, or contractual agreements, cost of deployment of software and personnel for incident management, cost of data recovery, data replacement/re-keying, or discarding of damaged data. In addition, costs may run to systems' downtime/inactive personnel, post-traumatic reconfiguration and the cost of postural reassessment and finally anti-virus deployment costs.

Perhaps the greatest cause for resenting the encroachment of malicious software onto the desktop is that the cost of (functionally) effective malware management sometimes seems disproportionate to the perceived benefits, especially when considered in terms of procurement costs (risk v. cost analysis, product evaluation and licence negotiation and procurement, measured in terms either of unavailability of staff for other in-house tasks, or in terms of the costs associated with consultancy, outsourcing, management decision-making processes etc).

This also applies to the initial implementation (planning, configuration testing, compatibility testing, initial rollout), and maintenance costs (distribution of updates, incident management, user education, IT staff training, implementation of policies and standards, dealing with unacceptable hardware/software conflicts on non-standard systems). Also to be considered is the negative impact of defensive measures on system performance and on employee morale and performance.

To this, we need to add the cost of vendor-independent malware management such as hoax management, formulation and implementation of policies and standards, information gathering, and the costs of the ongoing postural reassessment cycle. The reassessment process must take account of changing corporate vulnerabilities, reflecting changes in malware and anti-malware technology.

CORPORATE TUTORIAL

Office 2000 and Macro Security

Darren Chi & Raul Elnitiarta
Symantec

Microsoft Office 2000 introduces a number of features that aid macro security. These features will help limit the spread of macro viruses and the potential damage that can be done by them and other malicious macros. However, *Office 2000* does not mean the end to all macro security worries. This article describes the new features, provides suggestions on using them, and talks about the issues and concerns.

Digital Signatures

Word, *Excel*, and *PowerPoint* in *Office 2000* support digitally signed VBA macros in documents. Thus, when opening a document containing signed VBA macros, *Office 2000* can verify the author of the macros in the document and that they have not been modified, for example by a virus, since being signed. *Access 2000* does not support digital signatures on VBA macros.

Several details are worth mentioning about how a digital signature affects a document. The signature is applied to the entire VBA project of the document. Thus, two different VBA macros or modules in the same VBA project cannot each have its own digital signature. Modifying the text of the document does not have any affect on the signature on the VBA project because the signature applies only to the content of the VBA project. Thus, the document text can be modified freely without invalidating the digital signature on the VBA project of the document.

Furthermore, modifying any part of the VBA project, such as adding a new macro, invalidates the digital signature. Finally, the digital signature information has no meaning to *Office 97*, so in order to prevent *Office 97* from overwriting the digital signature information on the VBA project, the VBA project is protected from modification.

Security Levels

Office 97 provided the Macro Virus Protection feature. When opening a document containing macros, this feature presented a dialog signalling the presence of macros in the document and allowed the disabling of the macros prior to opening it. Disabling the macros would thus disable any potentially malicious macros in the document.

Office 2000 replaces this feature with the ability to choose one of three security levels that work in conjunction with a list of trusted digital certificates, known as the Trusted Sources list, to provide a customizable level of protection.

Low Security effectively means no security. When opening a document with macros, *Office* will not present any warnings and the macros are fully enabled.

Medium Security means that when opening a document with macros, *Office* allows the user to decide whether or not to enable the macros. If the macros have been signed, *Office* also allows the user to add the digital certificate identifying the author to the Trusted Sources list. If the certificate is added to this list, the next time the user opens a document with macros signed using the same digital certificate, *Office* automatically opens the document with the macros enabled without any warnings.

High Security means that *Office* automatically disables unsigned macros without presenting any warnings first. When opening a document with macros signed using a digital certificate that is not in the Trusted Sources list, *Office* allows the user either to disable the macros or to add the certificate to the Trusted Sources list and enable the macros. Oddly enough in this case, *Office* does not provide the option simply to enable the macros without adding the certificate to the Trusted Sources list. If a document with signed macros uses a digital certificate in the Trusted Sources list, *Office* automatically opens the document and enables the macros without any warnings.

Word 2000 installs by default at the High Security level. *Excel 2000* and *PowerPoint 2000* install set to the Medium Security level by default. *Access 2000* does not support security on VBA macros.

Trusted Sources

The only way to enable macros under High Security is to add the digital certificate of the publisher to the Trusted Sources list. This makes it difficult to enable and disable macros in selected documents containing macros by that publisher. Remember, adding a publisher to the list causes all subsequently opened documents containing macros signed by that publisher to become silently enabled.

Another important point worth mentioning about the Trusted Sources list is that it is shared among all *Office* applications. This may be disadvantageous if you decide you trust *Word* macros but not *Excel* macros from a particular publisher. To clarify this point further, adding the publisher to the Trusted Sources list in *Word* automatically makes that publisher a trusted source to both *Excel* and *PowerPoint* too, and vice versa.

Add-Ins and Templates

In *Office 97*, there is no way *Office* can warn of the presence of macros in installed add-ins and templates that load automatically, examples of which are templates in the *Word*

Templates directory or files in the *Excel* XLSTART directory. The lack of such a feature allows an external program to drop an add-in or template containing malicious macros into the user's environment.

Office 2000 addresses this issue by giving the option to treat installed add-ins and templates the same way as documents opened normally. Enabling this option tells *Office* to apply the security level setting to installed add-ins and templates when they are loaded. Note that a default installation of *Office 2000* has this option disabled and so will automatically trust installed add-ins and templates.

Excel 4.0-style Macros

Excel 2000 still supports *Excel 4.0*-style macros but they cannot be signed with a digital signature. This weakness is compounded by *Excel's* inability to disable these types of macros when opening a document containing them.

When opening a document containing *Excel 4.0*-style macros under the High Security level, *Excel* presents a dialog stating that these types of macros can neither be disabled nor signed and thus the document cannot be opened. If you want to operate under the High Security setting but still open documents containing *Excel 4.0*-style macros, you will have to create the following registry key … Microsoft\Office\9.0\Excel\Security\XLM=1.

If you use this setting and High Security, *Excel* will warn of the presence of *Excel 4.0*-style macros in a document and allow the choice of whether or not to continue to open the document. VBA macros are still subject to the normal High Security level protection. Under the Medium Security level, *Excel* will warn of the presence of *Excel 4.0*-style macros in a document and offer the choice of whether or not to continue to open the document. Under the Low Security level, *Excel* automatically allows documents containing *Excel 4.0* macros to open without warning.

Anti-virus API

Office 2000 makes a rudimentary attempt to provide better support for third party anti-virus scanners through a new API. The API does nothing more than call upon registered anti-virus scanners to scan a document when the document is opened regardless of the security level setting. This includes the cases where *Office* automatically disables macros (i.e. unsigned macros under High Security) or automatically enables them (i.e. macros signed with a digital certificate in the Trusted Sources list).

There are some important points to know about the API. It only calls upon the registered anti-virus scanners when a document is opened. It has no facilities to tell the anti-virus scanner to scan all the documents stored on a hard drive. The registered anti-virus scanner is responsible for all aspects of scanning the opened document. The API provides no other support other than notifying the anti-virus scanner of the open event.

When the registered anti-virus scanner is called upon to scan a document, it may or may not scan embedded documents within the given document, depending on its capabilities. The API itself will only call upon the anti-virus scanner to scan an embedded document when the embedded document is opened.

Support for the API is not a substitute for on-access scanning because the API supports scanning of documents only when they are opened from within *Word*, *Excel*, or *PowerPoint*. The API has no effect when you access documents in other ways, such as when you send a document from your hard drive as an email attachment.

Web Page Documents

Office 2000 introduces a new file format called the Web Page document – literally a document designed for viewing over the Web. The difference between a Web Page document and a plain HTML document is that the former retains the same information as a native document, including macros, whereas a plain HTML document does not. Native format is the default format in which documents are saved. This means that *Word*, *Excel*, and *PowerPoint* Web Page documents can harbour macro viruses just as well as can those same types of documents in native format.

Fortunately, VBA macros in a Web Page document can be signed and are subject to the same security level protection as those in native documents. Furthermore, the new anti-virus API also calls upon registered anti-virus scanners to scan Web Page documents. However, Web Page documents are stored in an entirely different format from native documents and so anti-virus scanners must be specifically enhanced to understand this new format in order to be able to scan such documents properly. This requirement applies to both anti-virus scanner components registered with the new anti-virus API and to existing on-demand and on-access anti-virus scanners.

Vulnerabilities in Office 97

Naturally, *Microsoft* carries over the fixes to vulnerabilities present in *Office 97* into *Office 2000*. More details can be found at http://www.microsoft.com/security/.

Word 97 Template Security Patch

With the Macro Virus Protection feature enabled, *Word 97* warns of the presence of macros in a document when it is opened and allows the user to disable the macros. However, *Word 97* does not warn of the presence of macros when opening a document with an attached template that contains macros. This vulnerability allows for an attack such that malicious macros in an attached template can execute without the user's knowledge.

After the discovery of this vulnerability, *Microsoft* released the *Word 97* Template Security Patch. Applying the patch to *Word 97* changes its behaviour so that attached templates are also subject to the Macro Virus Protection feature. *Word 2000* applies the security level protection as well as registered anti-virus API scanners to attached templates. Thus, the vulnerability does not exist in *Word 2000*.

Word 97 and Right-Click Printing

When you right-click on a *Word* document from either the desktop or *Explorer* and select Print from the context menu, *Windows* starts *Word*, which in turn opens up the document and prints it. Curiously, *Word 97's* Macro Virus Protection feature does not activate when the document is opened in this way. This means that macros in the document can execute automatically without the user's knowledge.

Fortunately, *Word 2000* addresses this issue and applies security level protection as well as registered anti-virus API scanners to documents opened through right-click printing.

Excel 97 CALL Function Patch

In *Excel 97*, it is possible to call a function in a DLL from within the cell of a worksheet. When opening a document with such a call, even with the Macro Virus Protection feature turned on, *Excel* does not give a warning of its presence. Consequently, it is possible that a worksheet could have a malicious call to a DLL that executes without the user's knowledge.

Microsoft released a patch that addresses this vulnerability called the *Excel 97* CALL Function Patch. *Excel 2000* ships with the worksheet CALL functionality disabled and so there is no need to install a patch.

Registry Security Settings

Office 2000 stores its security settings in the registry. This means that it is possible for an external program to change the security level from High to Low simply by modifying entries in the registry. In fact, current macro viruses are doing so already. It is not possible to protect the registry from such attacks under *Windows 95/98*. However, it is possible to do so with *Windows NT 4.0* with SP3 or newer.

Office stores its settings in the HKEY_CURRENT_USER section. However, when starting up, *Office* first checks the HKEY_LOCAL_MACHINE section for settings. Any settings that are found there override those located in the HKEY_CURRENT_USER section. On *Windows NT*, a systems administrator can prevent user write access to the HKEY_LOCAL_MACHINE section of the registry and thus lock down and protect the security settings from attempted changes by both users and malicious macros. These security settings include the Trusted Sources list.

Microsoft provides a document entitled 'Microsoft Office 2000 Macro Security' that lists the registry locations where *Office 2000* stores its security settings. This document can be obtained from http://officeupdate.microsoft.com/2000/downloaddetails/o2ksec.htm.

Caveats When Signing Macros

If a VBA project is signed and the certificate that was used to sign it is installed on the system, then VBA automatically resigns the project with that certificate when the project is modified and resaved. This refers to the certificate used to sign the VBA project as opposed to that stored in the Trusted Sources list to identify a publisher.

Since the project is automatically resigned, VBA authors need to be especially careful that the system on which they are authoring VBA macros is virus free so that they do not sign a macro virus unintentionally into their VBA project.

Anti-virus Scanner Updates

Anti-virus scanners need to be updated specifically to handle *Office 2000* documents. Although *Office 2000* documents are compatible with *Office 97*, *Microsoft* has made changes to the internal storage format of VBA macros. Anti-virus scanners that have not been specifically updated to handle the changes are likely to be ineffective when it comes to the reliable detection of viruses in documents saved in *Office 2000*. This especially pertains to Web Page documents since they are in a new format.

The following recommendations for using *Office 2000* will help to eliminate the threat of malicious macros:

- Set all *Office 2000* applications to the High Security level setting.
- Disable the 'Trust all installed add-ins and templates' setting.
- Convert *Excel 4.0*-style macros to VBA macros.
- Sign all macros, even those in add-ins and templates.
- Insist that all who transfer documents with VBA macros to you verify that their documents are free of malicious macros and that all macros are signed.
- If you are an administrator of a *Windows NT* system, in the HKEY_LOCAL_MACHINE section of the registry, set and lock the macro security settings so that users and malicious macros cannot modify them. When logging in as administrator, be extra careful to ensure the macro security settings do not get modified accidentally by you or by malicious macros.
- Use a proven anti-virus program to scan incoming documents, even ones from a 'trusted source'.

Conclusion

Microsoft Office 2000 certainly does take a significant step forward in the macro security field. The wise and cautious user will find it prudent to take advantage of the new measures, such as setting the security level to High to guard against the potential harm that malicious macros can do. However, do be aware that anti-virus software is still essential in order to alert you to the actual presence of malicious macros in documents.

FEATURE SERIES

Macro Viruses – Part 2

Dr Igor Muttik
AVERT Labs, UK

Most of the old macro viruses work under and were written for the *WinWord 6.0* macro language which is known as WordBasic. The commonness of the *Word 6.0* environment at the time enabled *WinWord* macro viruses to become widespread. WordBasic is based on the good old BASIC programming language but has many (hundreds) of extensions (for example to deal with documents: edit, replace string, obtain the name of current document, open new window, move cursor, etc.).

WinWord 7.0 was included in *Office 95* and it also used WordBasic. The appearance of macro viruses forced *Microsoft* to release a version with some sort of protection. *WinWord 7.0a* was the first *MS* application to have the built-in anti-virus warning mechanism which is now present in almost all *Office* applications.

VBA3, VBA5, VBA6 and Excel Formulae

Excel 5.0 had a macro language called VBA3 (Visual Basic for Applications v.3). This was used as a prototype for VBA5 – the macro language for *Office 97* applications. VBA6 is the one used in *Office 2000*. For the list of new features and differences see *VB*, August 1999, p.13.

Old *Excel 4.0* had formula macros. They were kept for compatibility in all later *Excel* versions. There are also field viruses (e.g.XF/Paix) that use this macro language. Formula macros are written in an *Excel*-specific language that has nothing in common with either WordBasic or any VBA flavour and they live in the *Excel* spreadsheet's cells.

Visual Basic for Applications

In January 1997 *Microsoft* unveiled *Office 97* – it was a complete rewrite, no longer using WordBasic. In *Office 97* all applications use the same macro language – VBA5 (Visual Basic for Applications). *WinWord 8.0* (*WinWord* in *Office 97*) has an ability to convert (recompile) old macros into this new language. Many viruses can be recompiled this way resulting in completely different viruses (some of them non-viable as the convertor success rate was estimated by *Microsoft* at about 90%).

Further, *Microsoft* put some sort of detection of the most common viruses into the convertor to prevent their recompilation (so that, for example very common viruses like WM/Concept.A, WM/Wazzu.A and WM/Npad.A are not converted). Unfortunately, these precautions were not made in *Office 97* beta releases and several viruses were upconverted to the new format by the beta software.

Another feature of *Word 97* is that it produces a warning (like *WinWord 7.0a*) if somebody is trying to load a document containing macros. It displays a dialog box saying 'The document you are opening contains macros or customizations. Some macros may contain viruses that could harm your computer.' and presents three options:

1) Disable Macros (default)
2) Enable Macros
3) Do Not Open

This warning, however, can be turned off so that it will never appear again.

VBA5 is a far more complex language than WordBasic (in fact, it even includes WordBasic as a subset of its commands) and its data is stored in a file in a much more complex way. Macros written in VBA5/6 are represented in OLE2 files by two different entities – there is a compiled macro body and also compressed macro text (both are usually present in OLE2 files with macros). When macro text gets modified the macro body is recompiled from it.

Usually, both instances of a macro contain the same information (in simple terms, one is used by the VB editor, another by the VB interpreter). However, in the case of corruption this may be not true – for example, even if macro text is missing the compiled body could still be executed. Different scanners may choose to detect macro viruses in either form (compiled body or compressed text). This explains why different scanners may produce incoherent results on some corrupted samples (having, say, one form missing or damaged).

Under *Office 97* all major applications use the same macro language. That means cross-application viruses are possible (see *VB*, October 1998, p.9). What is more, files with PPT extensions (*PowerPoint 97*) can now have macros (which previous incarnations of *PowerPoint* did not have at all). Naturally, the first *PowerPoint* viruses appeared after *PowerPoint 97* had been released.

Upconverting and Downconverting

Excel 97 has an ability to save spreadsheets in old *Excel 5.0* format (i.e. in VBA3). So viruses in VBA5 format can be 'downconverted' back to VBA3 format. It is even possible to have both VBA3 and VBA5 incarnations of macros in a single spreadsheet file, recognizable by both old and new *Excels*. Downconverted viruses can be upconverted again, resulting in exactly the same virus body. However, it is known that the virus' formatting (for example, spaces, tabs and empty lines) does change. This is why, in the proper identification of *Excel* viruses, these variable parts should be ignored.

Following the release of *Office 2000*, macros written for *Office 97* (VBA5) could be upconverted to *Office 2000* (VBA6) and downconverted back to *Office 97*. Any anti-virus scanner should be capable of automatically dealing with all differences which appear as a result of multiple up/down-conversions.

Many viruses do not survive this (the convertor adds an empty line for every downconversion which breaks most common viruses). However, even among broken viruses a scanner should be able to find and clean the non-viable, damaged viruses.

VBA6 macros do not differ much from VBA5 ones. In fact, VBA6 includes VBA5 as a subset. Macros written for *Office 97* usually work under *Office 2000* while the opposite is not always true (i.e. *Office 2000* is downwardly compatible with *Office 97*). Decent scanners are not able to distinguish between *Office 97* and *Office 2000* incarnations of the same virus, performing internal mapping to cover up/down-conversions automatically.

Typical Life-cycle of a Macro Virus

The life cycle of a great majority of *WinWord* macro viruses is as follows. A macro virus in a document which is being loaded gets control (for example via so-called auto macros, those which get executed automatically at certain moments; such macros are – AutoOpen, AutoClose, etc). The corresponding macro copies all the viral macros to the global template (NORMAL.DOT on a PC). NORMAL.DOT is used automatically when *WinWord* starts. It contains user settings (fonts, etc) and shortcuts (key redefinitions) and it can also contain macros.

If NORMAL.DOT contains the AutoExec macro it will be executed when *WinWord* is started. If NORMAL.DOT contains AutoClose it will be executed every time any document is closed.

Macro viruses do not necessarily have to infect the global template (NORMAL.DOT). Some infect files directly, searching for a victim on a disk and infecting it that way. WM/Snickers and WM/Ordo use the MRU list (most-recently-used list at the bottom of the File menu usually consisting of four items) to get the names of files to infect. Others (like the W97M/Groov family and WM/Eraser) drop their own template in *WinWord's* template directory and may or may not also infect NORMAL.DOT. This additional template (if it is registered as an add-in) will work in exactly the same way as NORMAL.DOT and *WinWord* will pick it up automatically.

In many ways, *Excel* infectors are similar to *Word* infectors. However, instead of infecting NORMAL.DOT, viruses written for *Excel* usually drop a new startup file in the XLSTART folder. *Excel* automatically picks up such dropped files when it starts. The most common name is PERSONAL.XLS (which is *Excel's* default name for a startup file; Laroux.A uses it). However, the name can be anything – Laroux.e uses PLDT.XLS. *Excel* does not bother to check the extension so many viruses drop a file with no extension like 'BOOK1.' (e.g. O97M/Tristate.A).

Auto Macros

Most macro viruses operate using auto macros. *Word 6.0* had just a few of them (AutoOpen, AutoClose, AutoExec, AutoExit and AutoNew). VBA5/6 have many more. Apart from the old ones kept for compatibility, there are also 'event handlers' (Document_Open, Document_Close, Workbook_Open, Workbook_BeforeClose, etc.).

Event handlers should be put in special 'class' modules to be able to work (hence the name of the first W97M virus to use event handlers – W97M/Class). Event handlers exist only in VBA5/6 and are not present in VBA3 (*Excel 5.0*). That is why many *Excel 97* viruses (X97M/Hopper, for example) cannot be downconverted to *Excel 5.0* (following downconversion there are no modules in the file).

In fact, probably every single field virus makes use of the 'Auto' macros. Viruses which do not use them do exist but their chances of spreading are simply below the threshold that enables them to survive in the wild.

Menu Items Interception and Key Shortcuts

It is easy to modify the functionality of any *Office* application by associating its menu item with a macro. For example, many viruses have the macro called FileSaveAs (in VBA it would be a function with the same name and it can be defined in any module). If this menu item is activated by a user it is the macro which gets control, pretending to be a real menu option while it copies additional viral macros to the destination file. Viruses can also remove and modify menu items (many remove the Tools/Macro item to make it impossible to check for the presence of viral macros) using the Tools/Customize functionality.

Macro viruses can attach a macro to a particular keyboard key. For example, WM/Gangsterz and WM/DLK1.a link their viral macros to frequently used keys (like space, 'e', 'a') and activate when this key is pressed. This is one of the ways macro viruses can avoid using auto-macros or menu-linked macros to get control.

Polymorphic Macro Viruses

There are many known polymorphic macro viruses in existence at the time of writing. A few examples include WM/FutureN, WM/Outlaw, WM/Slow, WM/Minimorph, W97M/Class.a and W97M/STP.

They all use *WinWord's* editing abilities to modify their own macros (like replace function) before copying them. This has the effect of making the virus body variable. Another approach to hiding parts of the virus is to use document variables which are stored in a file (for example, a WordBasic program can assign a string variable A$ and then save it in a file along with macros). Such variables can contain various bits and pieces of viral code/data which are used by viral macros.

Stealth and Encryption

Stealth for macro viruses involves some measures to prevent the easy viewing of a virus' source code. Normally, the host program is capable of displaying the source code of any macro or module. To prevent easy viewing, some viruses remove the Tools/Macro, File/Templates/Organizer and Visual Basic Editor menu items. Some viruses present the user with artificial, empty dialog boxes instead of real ones or produce fake errors.

Macros can be encrypted. Encrypted macros are simply stored in scrambled form (however, note that the encryption of macros does not affect the text in a document – it still is easily readable). So, there are encrypted and not encrypted macro viruses. When an encrypted macro is shown in Tools/Macro the option to edit is not available. The encryption is easy to overcome – the key to decrypt macros is in the file, so it is not a problem to scan encrypted macros for viruses. Macro encryption is also called 'macros are read-only' because the macro editor does not allow the editing of encrypted macros.

Macros encrypted in *Word 6.0* cannot be converted to either *Office 97* or *Office 2000* because the latter insists on protection being enabled on modules' projects, not individual macros (under both applications several functions/ macros can be defined in one module).

Office 97/2000 can also use 'read-only' macros (or, as *Office* puts it, 'lock project for viewing'). Such macros have a special flag set and cannot be edited in the Visual Basic Editor. Macros, however, are not actually encrypted in any way and macro bodies are easily accessible by any tool except the built-in macro editor.

Password Protection

Entire *WinWord 6.0/7.0* documents can be password-protected. This means that the whole file is scrambled and access to the text and macros is not possible without deciphering the file. The password is needed to access the macros and check them for viruses. However, the protection is weak and there are many shareware and freeware *WinWord* password crackers around. Many contemporary scanners are able to do on-the-fly cracking of password protected documents to check them for viruses.

Under *Office 97/2000* macros are not password-protected so scanning for viruses is not only possible but easy, even when the text is protected by a password. Repair, however, might be difficult as the area protected by the password may contain references to viral macro bodies. Furthermore, encryption in *Office 97/2000* is far more advanced and cannot be cracked on-the-fly.

Corruptions and Manual Editions

WinWord 6.0 has buggy routines that are responsible for macro copying. As was discovered by the author, any I/O error during macro copying produces unpredicted results on the destination macro. For example, if macro virus was in a document on a floppy disk and the disk was removed when the macro copying was being performed, parts of the written copy will be corrupted without any warnings.

In fact, there are about 200 different variants of WM/Npad around and all of them (except the original virus) are the result of the natural corruption described above. Between 1997 and 1998 this natural corruption was the main source of macro viruses because many of them are able to replicate even if they are seriously corrupted (the WordBasic statement 'On Error Goto Next' helps a lot).

In *Office 97* corruptions are very rare so they are not responsible for creating new viruses. However, the availability of the Visual Basic Editor (say, via Alt+F11) makes it very easy to modify the source of any virus manually (if it is not a 'read-only' macro). Users' modifications to field viruses are the most common source of new variants because the vast majority of manual modifications are perfectly able to travel even when the user is attempting to 'disable' the virus.

Remnants

Some macro viruses have just one macro (like WM/Wazzu, W97M/Ethan, WM/MDMA). However, many macro viruses consist of multiple macros (for example, WM/Rapi, WM/Concept and W97M/Aleja). It could and does happen that some of the macros belonging to a virus go missing.

This might happen because somebody deleted alien macros using the Tools/Macro/Delete command or some sloppy anti-virus tool that did not perform the disinfection correctly or because *WinWord* hung in the middle of the macro copying operation. Whatever the reason, we get a document with the remnants of the original macro virus. Remnants occur very frequently in *Word 6.0* viruses.

In most cases, remnants no longer constitute a viable virus because some part of the original is missing. However, in some cases remnants can still be viral. Say, WM/DZT has two macros (AutoOpen, FileSaveAs in documents) – what is left can replicate if either of the two is missing.

Under *Office 97/2000* remnants are rare. The reason is that VBA5/VBA6 allows several functions (like AutoOpen) and event handlers (like Document_Close) to be present in one single module. So, there is no necessity for a virus writer to place a virus in more than one module. *Office 97* viruses spanning several modules do exist but they are rare. That is why viable remnants of native W97M and X97M viruses have yet to be reported.

[*Next month's final instalment covers mating, devolving, naming and prevalence. Ed.*]

PRODUCT REVIEW

RAV v7.0 for Windows 98

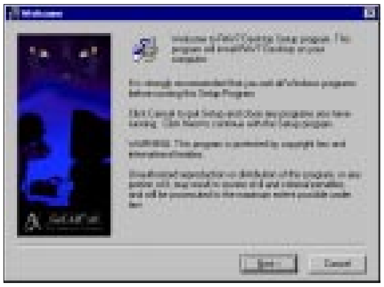
Martyn Perry

This is the first time I have had the opportunity to review the anti-virus product from the developers at the Romanian *GeCAD* – 'Romanian Anti-Virus', *RAV*. A regular participant in the *VB* Comparative Reviews, this is the first look at the freshly revamped *RAV 7* – will it be a good experience?

RAV 7's licence covers the installation of the software on a single PC, with the dispensation to have a copy on a separate, portable PC or home computer. The software may be installed on a network provided that it is only used by the specific licensee. However, separate licence packs can be obtained to increase the usage entitlement.

The product is serialized both with a five digit code that is printed on the outer packaging and registration form and with a 16-character code printed on the licence certificate. This is used as part of the registration process.

Presentation and Installation



The product is supplied on CD, which autoloads directly into the installation program. The first dialog prompts to choose language – English or Romanian. Choosing English produces the 'Welcome' screen. The next click presents the Licence Agreement, which is followed by the choice of installation, either Quick Setup (recommended) or Custom Setup (for advanced users). It would have been instructive to give the installer a summary of the advanced options available – there is certainly room on the screen.

The initial installation used the 'Quick Setup' option. This moves to the next screen which chooses the installation folder (the default is C:\Program Files\GeCAD\RAV7 Desktop). This is followed by naming the group for the program (default being RAV7 Desktop). The next step allows for a recheck of the selections before beginning the file copy process. When this has completed, the final screen gives the option to scan all fixed drives after restart. This was left deselected to allow control of the scan process.

After restarting, if the 'RAV Monitor' option is selected from the task bar, the first screen prompts for registration. The product has a default evaluation period of 30 days, but the 16-character code can now be used to register the software as a full licence.

If the Custom Setup is selected, then there are additional configuration options. Firstly, an option to install a DOS version of the scanner – essential for dealing with viruses which cannot be cleaned from within *Windows*. Other options provide links to enable easy access to *RAV 7*, for example with shortcuts on the desktop and Start menu, or the use of shell extensions.

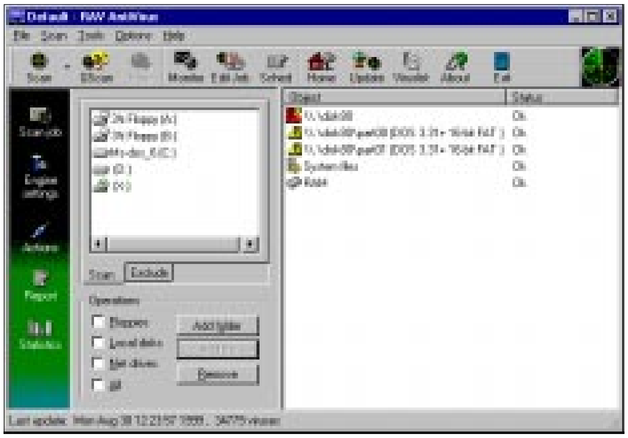
Finally, there is an option to start the real-time monitor by default at each *Windows* startup. If this option is enabled, the installation screen finally prompts for a restart of *Windows*. If the installation is not configured to start the real-time scanner, then a restart is not prompted for at the completion of the installation.

The CD contains a number of folders, which give options such as the ability to create a floppy disk set for deployment. Within the DOCS folder there are folders for English and Romanian language sets. The English documentation is available in HTML format, making it readable in *Windows 98* without additional file readers being required.

A hard-copy of the documentation is also shipped as part of the *RAV* package – something that is often missing from other products. Electronic documentation certainly has its advantages, but there is no doubt that, at times, referring to a manual is vastly preferable. The manual is concise and well presented, and, where appropriate, is adorned with the relevant screen shots.

RAV 7 for Windows 98

The main screen provides access to all the menu selections. The various scan objects along with required actions can be incorporated into a 'Job'. This Job can be created, edited and saved as part of the main application. The Job files have the extension .RJB. However, during testing a problem was noticed if when saving a new Job, the RJB extension was not entered. In such a case, a fatal exception was observed – a minor niggle for the developers to correct.



The scanner can have specific drives and directories for scanning. In addition, an exclusion list can be created to prevent certain areas being scanned. This I found useful while loading some of the clean test samples without having to wait for them to be scanned.

RAV 7 offers three levels of scanning – Safe mode, with heuristic scanning deactivated; Standard mode, with minimum heuristic analysis for unknown virus detection; Adaptive mode, with the highest level of heuristic analysis.

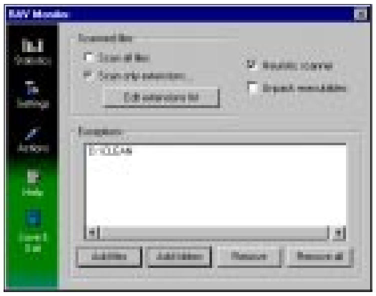
Manual/On-demand Scan

Setting up the scan engine involves selecting the heuristic level, choosing whether to unpack compressed executables and whether to scan inside archive files. When determining which files to scan, the choice includes: all files, typical extensions or custom extensions. In the case of both the typical and custom extension lists, there is the ability to edit the default entries.

The default set of file extensions to include in a scan are: 386, ARJ, BAT, BIN, BOO, COM, DO?, DLL, DRV, EXE, GZ, HT*, IMG, JS, LHA, LZH, MDB, OVL, OVR, POT, PPT, PRG, RAR, SAM, SCR, TAR, TD0, VBS, VXD, XL* and ZIP.

The actions one can take on detecting a virus are grouped depending on the nature of the infection. With ‘Infected’ objects one can Clean, Ignore, Rename, or Delete. With ‘Suspicious’ objects the options are to Copy to Quarantine, Ignore, Rename, Delete or Validate. Finally, ‘Infected and Uncleaned’ objects allow the user to Copy to Quarantine, Ignore, Rename or Delete. These actions can be set to activate automatically or to wait until the first occurrence for manual intervention.

Real-time Scan



Scanned files
○ Scan all files
● Scan only extensions
☑ Heuristic scanner
☐ Unpack executables
Exceptions
D:\CLEAN

This facility is provided by ‘RAV Monitor’. Normally, this is included at startup and remains running the whole time the PC is in use. If it is necessary to stop the monitor temporarily, then a right click on the *RAV* icon in the system tray gives access to the options which include ‘disable monitor’.

The actions available for on-access scan again apply to the three categories of object mentioned above. For ‘Infected’ objects, the options are to Block, Clean or Ignore, while for ‘Suspicious’ objects one can Block or Ignore. The same two options are available in the case of ‘Infected and Uncleaned’ objects. In this case, Block is used instead of Copy to Quarantine. This is to deny access by applications to the infected files, hence limiting potential viral spread.

As is customary, a log file for infected or suspicious files can be created to record the monitor’s activity. The default file is …\GeCAD\RAV7 Desktop\RAV7MON.TXT.

Scheduled Scan

Scheduled scans make use of the Job files by scheduling their start time and repetition rate. A one-off scheduled scan can have its start time and date selected. However, if one is running a scheduled scan on a daily basis, the interval between each run can be defined. The day of the week is selectable for weekly scans and finally, for monthly scans, the month together with the date or the occurrence of a day within the month can be chosen.

The scheduler can be configured to run other tasks. These include Run Live Update and Run Custom Task. The former can be used to download updates from a *RAV* Advanced Server that has been updated with the latest virus signatures from the *RAV* Web site. Alternatively, the Run Custom Task option looks for a program file or batch file to execute. This can be used to schedule backups or other data administration activities.

Administration

While many anti-virus products currently on the market have a quarantine folder to store infected or suspicious files, *RAV* goes a step further by allowing manual modification of the quarantine list. A file can be added either by physically moving it to the quarantine folder, or by placing a link to it there, leaving the original file unmoved. This is useful as it saves having to move key system or program files which may impact applications. Files can be removed from the quarantine folder in the situation where they have been incorrectly identified by the heuristic engine as infected, when in fact they are not.

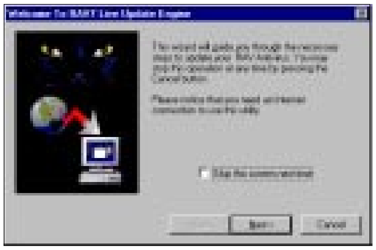
Another option is to mark files in the quarantine folder for shipping to *GeCAD*. This leads to the further options of either using HTTP transfer via the Internet, attaching files to an email or copying the files to a floppy. The problem with this is that if an attempt is made to copy to a write-protected floppy, there is no checking that the media can be written to. This is compounded by the fact that the status in the quarantine directory changes even though no file transfer has occurred.

Web Presence and Support

The *RAV* Web site is well-organized with a combination of facilities that users have come to expect. On the commercial side, there is a profile of the Romanian company, its products and its software services, including consultancy and software installation, as well as the facility to purchase the software on-line.

On the technical side, there is a good section on virus information which not only applies to the local Romanian market but has general relevance. This information supplements the virus description list which is available with the product itself. An email facility allows problems to be sent to *RAV* tech support and a download facility dispenses products and virus signature updates which can be linked to the live update wizard.

Updates



Welcome To RAV7 Live Update Engine

The wizard will guide you through the necessary steps to update your RAV AntiVirus. You may stop the operation at any time by pressing the Cancel button.

Please notice that you need an Internet connection to use this utility.

□ Skip this screen next time

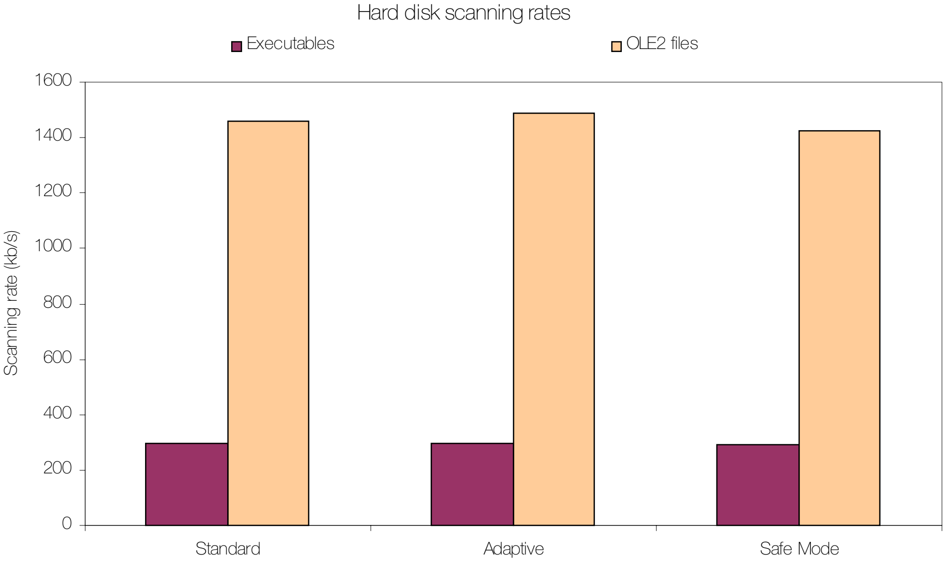
The version tested for this review was 7.2.357.5.1057. *GeCAD* includes an update wizard in *RAV* called Live Update. This can source updates from an Internet connection, *NT* Server connection hosting *RAV* Advanced Server or from a floppy or email update.

On-demand Scan Rates

To measure the extra work performed in detecting a virus, a diskette comprising 26 EXE and 17 COM files was scanned. The scan was repeated with the files infected with Natas.4744 virus. During the test it became clear that while scanning floppy disks, it was necessary to select the scan option twice to get the screen to update with the virus infection warning dialog. If the floppy was removed, and the next floppy inserted, this worked correctly. The overhead in scanning the floppy containing the infected files was approximately 24%.

The hard disk scanning rates were then investigated for each level of heuristics – firstly, using a set of 5,500 clean files, and then using a set of OLE2 files. The scan results are summarised in the graph below. The settings used were Scan All Files in the clean folder, Unpack Executables and Scan Inside Archives. The time between the quickest and the slowest made only about 1.5% difference to the measured overhead, leading the reviewer to question exactly what differences actually exist between the levels of heuristics offered by *RAV* – genuine functionality or mere window dressing?

Consultation with the developers at *GeCAD* identified why the different levels of heuristics make little difference in terms of scanning speed and overhead. Whatever the setting, the heuristic engine collects a particular set of flags. The heuristic engine's settings are only relevant when it comes to computing the probability factor from these flags.



Against the executable clean set, two false positives were registered throughout both the heuristics modes – obviously the probability factor was beyond the 'suspicious' threshold in the least paranoid heuristics setting.

Detection Rates

The scanner was checked using the standard *Virus Bulletin* test-sets. Importantly, the ItW set was aligned to a June 1999 WildList.

Initially, the tests were conducted using the default list of file extensions supplied with the scanner. The option to delete infected files was selected. The residual file count was then used to determine the detection rate (verified by cross-checking against the scan log). It appears that the scanner does not handle files with the read-only attribute set very well. Upon detecting an infection, instead of deleting the scanned item, it reports that it is copying it (to the quarantine folder). However, only the file name is created in the quarantine folder – the infected file remains *in situ*. When the read-only attribute was removed from the test-set files, the scanner handled the infections correctly. Such problems originate from the way in which *RAV* handles delete or rename operations compared to the disinfect operation. The latter is handled internally by the virus engine – indeed, brief tests verified that *RAV* successfully initiated disinfection of infected read-only files (the success or not of the disinfection process itself was not assessed). However, both the rename or delete operations are left to the user in the current product version.

As would be expected for any quality anti-virus product given that the tests were performed against a June 1999 WildList, detection of viruses in the ItW boot and file test-sets was 100%. In the Polymorphic tests *RAV* only missed three out of the 174 samples of ACG.A. Elsewhere, only one sample was missed from the other sets, namely an executable file infected with AIDS-II. The complete detection of all the samples in the Macro test-set was due partly to *RAV's* heuristics, which managed to flag 13 of the samples as suspicious. These included the A and C variants of W97M/Carrier, W97M/Boom.A and the recently introduced W97M/Flitnic.

It should be noted that the high detection rates observed here are only what should be expected from any quality anti-virus product, and are not be taken complacently. The *VB* standalone reviews are concerned with general features of the overall product, not simply the detection rates – this is an area best left to the regular comparative reviews.

Real-time Scanning Overhead

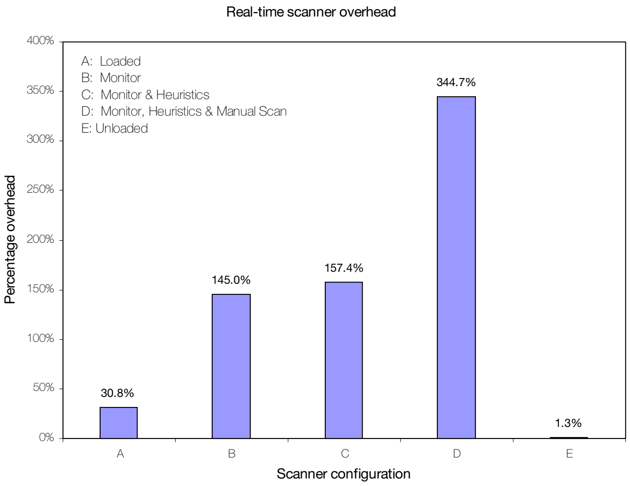
To determine the impact of the scanner on the Workstation when it is running, the following test was executed. The basis of the test was to time the following activity: 200 files totalling 23 MB (a mixture of DOC, DOT, XLS, XLT, XLA, EXE and COM files to reflect typical file types being moved) were copied from one folder to another using XCOPY. For the tests where the overhead was measured whilst an additional manual scan was performed, the folders used for the source and target files were excluded from this manual scan, so as to avoid the risk of a file being scanned while waiting to be copied.

The default setting of Maximum Boost for Foreground Application was used for consistency in all cases. The test PC was disconnected from the network throughout the tests, and the final results were averaged from ten iterations of the copying procedure.

The configurations used for the tests were as follows:

- Program not loaded: establishes the baseline time for copying the files on the PC.
- Program unloaded: run after the PC tests to check how well the PC is returned to its former state.
- Program installed, monitor off: tests the impact of the application in its quiescent state.
- Program installed, monitor on, heuristics off: shows the impact of having the monitor running but not using the heuristic tests.
- Program installed, monitor on, heuristics on: shows the impact of having the heuristic tests running.
- Program installed, monitor on, heuristics on, scanner running: tests the impact of the application scanning files when running a separate scan on the PC.

As can be seen from the graph presented, only a slight overhead is imposed with the program loaded, but in a quiescent state. The overhead measured with the monitor on increased as expected when the heuristics (set to 'standard' mode) were initiated, to a little over 150%. Running the on-access and on-demand scanners simultaneously resulted in a large overhead of approximately 350%, again as expected. The overheads observed during testing are in line with those seen with other *Windows 98* anti-virus products during recent tests (see, for example, the May issue, p.20).



Summary

On the whole *RAV 7* did its job effectively, the program is simple to use and it delivers high detection rates. The overhaul the product has received since the *RAV 6* days was long overdue, and it is nice to see that as well as changes to the engine, the developers have also significantly improved the GUI, which was becoming a little dated.

RAV 7 comes packaged in a relatively small shroud of marketing hyperbole. One particular comment on the box which drew attention to itself was as follows 'Over 80,000 lines of carefully written C, C++ and ASM code to guard your work'. An easier means for *VB* to test and compare products might be simply to compare the code lengths?!

The few operational problems that were encountered during testing, though relatively minor, really should not be occurring. The poor handling of infected files with the read-only attribute is the main gripe – an issue which is currently being resolved by the *GeCAD* developers. The other area of concern is with the handling of floppy disks, whether it be scanning, write-protect media checking or the apparent copying of files from the quarantine directory. All of these issues should be easily resolved by the use of even more carefully written code!

Putting the minor moans to one side, my overall first impression is that this could be a product worthy of a much wider audience.

Technical Details

Product: *RAV v.7.0* (30/08/99)

Developer: GeCAD srl, Gheorghe Patrascu Str, bl PM 53 ap 8, Bucharest, Romania. Tel +4 01 6476309, Fax +4 01 3248409, email office@gecadsoftware.com,
WWW http://www.gecadsoftware.com/.

Price: $29 – electronic version, $52 – full boxed product.

Hardware Used: Workstation: *Compaq* Prolinea 590, 80 MB of RAM, 2 GB hard disk, running *Windows 98.*

[1]**Virus Test-sets:** Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NT/199909/test_sets.html.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park,
Abingdon, Oxfordshire, OX14 3YP, England

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

US subscriptions only:

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

Tel (781) 9377768, Fax (781) 9320251

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

END NOTES AND NEWS

Internet management company *intY* has introduced a pay-as-you-use system of anti-virus software through its management service. For as little as £1 a month per user, a machine can be protected by *Sophos Anti-Virus* (*SAV*). Centrally held software, managed by *intY*, is upgraded immediately and automatically without additional cost to the user. For more information contact Leah May; Tel +44 117 9272444 or email leah@bbpr.com.

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

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

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

This month the Internet Security Conference (TISC) takes place at the Boston World Trade Center from 11–15 October. The four-day intensive curriculum is dedicated to secure computing and networking with an emphasis on safeguarding corporate Internet connections. The program includes workshops, product showcases and security symposiums. For more details contact Paul Kent; Tel +1 408 3542500, email paul@mactivity.com or see http://www.tisc.corecom.com/.

A two-day course Investigating Computer Crime and Misuse will be run by *Sophos* on 10 and 11 November 1999 at the organization's training suite in Abingdon, Oxfordshire, UK. For further information, or to reserve your place, please contact Daniel Trotman at *Sophos*; Tel +44 1235 559933, fax +44 1235 559935, visit the company Web site http://www.sophos.com/, or email courses@sophos.com.

***Computer Fraud and Security's* fifth annual conference takes place from 29 November–1 December 1999 at the Copthorne Tara Hotel, Kensington, London.** Day 1 is devoted to the subject of the Internet with Day 2 dealing with 'Who and Where and Recovery'. Day 3 is an all-day *NT* Security and Audit Workshop. Delegates may register for one, two or all three days of the conference. For further details contact *Audit Conferences Europe Ltd*; Tel +44 1892 526099.

***Data Fellows' F-Secure Anti-Virus 5.0* comprises multiple scanning engines including *Data Fellows'* new *Orion* scanning engine.** The product can be deployed, updated and monitored through a centralized Java-based console. Virus definitions are handled automatically using *BackWeb* technology. *F-Secure Anti-Virus 5.0* is available as a point application or as part of the *F-Secure Workstation Suite*. It is priced at US$ 18 per user for a 100 user licence. For more information contact Pirkka Palomaki; Tel +1 408 9386700, fax +1 408 9386701 or email Pirkka.Palomaki@DataFellows.com.

Virus Bulletin often exposes free magazine CDs and the like as virus-infected. **London-based *Reflex Magnetics* has added CD Authorisation to its *Reflex Disknet Data Security Suite for Windows NT*.** The new addition consists of three components: a low-level device driver/filter which checksums each CD; a control program that verifies the checksums for CDs inserted into any drive on the network and a database holding valid CD checksums. CDs are scanned for viruses *before* they can be used on the network. The Suite is currently available at £249 +VAT for the Administrator version and £28 +VAT for the Client version. For more information contact Phillip Benge; Tel +44 171 3726666, email phillip.benge@reflex-magnetics.com or visit the Web site http://www.reflex-magnetics.com/.

***Network Associates Inc* (*NAI*) recently announced new strategic initiatives with *Novell Inc* to protect *NetWare* customers.** The initiatives include the bundling of *Dr Solomon's VirusScan* and *NetShield* point products with *Novell's ZENworks* and *NetWare for Small Business*. For details visit the Web site http://www.nai.com/.

VB'99 takes place at the Hotel Vancouver, Vancouver, BC from 30 September–1 October. Messages to editorial@virusbtn.com or to our voicemail service on +44 1235 555139 will be answered as soon as possible after the event. Thank you for your support.