ISSN 0956-9979 DECEMBER 1999

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Ring out the old:** Our final three analyses this year keep up with two of the newest viruses around and look back at a lasting Legacy.

• **Marker my words:** One large US corporation decided to trace its virus problems back to the source. Find out how it was done in our Case Study on p.12.

• **Jobs for the boys:** Twenty-four/seven is a way of life for Costin Raiu and David Ensign. Two very different professions come under the spotlight on p.14 and p.18.

CONTENTS

COMMENT
Out of the Mouths of Babes and Hackers 2

VIRUS PREVALENCE TABLE 3

NEWS
1. FunLoving Criminal? 3
2. Practise Safe Computing! 3

LETTERS 4

VIRUS ANALYSES
1. Bursting the Bubble 6
2. One Sharp Corner 8
3. Merry MMXmas! 10

CASE STUDY
Following the Breadcrumbs 12

INSIGHT
Counting the Costin of AV 14

FEATURE
French Connection II 16

A DAY IN THE LIFE
The Politics of Anti-Virus 18

OPINION
Happy New Year… 20

PRODUCT REVIEW
Grisoft AVG v6.0 21

END NOTES AND NEWS 24

COMMENT

“ Hacking: it’s illegal – but is it immoral? ”

Out of the Mouths of Babes and Hackers

Computer hacking has always been sensationalist, front page material, fit for Hollywood consumption in films such as *Sneakers* and *The Net*. When I received an invitation to attend a debate, featuring two infamous hackers, entitled ‘Hacking: it’s illegal – but is it immoral?’ at last month’s *Compsec’99* it was too tempting to refuse.

And so, in a session moderated by Winn Schwartau (President of *Interpact, Inc.*), Kevin Poulson and Sir Dystic took to the podium, to face what I expected to be a barrage of questions from vexed network administrators concerning the ethics and morals of the hacking trade. Both speakers have achieved notoriety (of a sort) for interestingly different things: Poulson through getting convicted of espionage following a long tangle with the *FBI*, and Dystic as the writer of the original *Back Orifice* remote administration tool. Both appeared honest and surprisingly coherent in their opinions – all this despite jetlag; they had just hopped off a plane from the US.

Pleasingly perhaps, they both satisfy certain stereotypes about hackers, and no doubt they learned their trade whilst performing minor hacks through their teenage years. Poulson acknowledges ‘the start of criminality’ as when he began hacking for financial profit. A distinction between benevolent and profit-related hackers is drawn by Dystic – he clearly (publicly) opposes any hacking activity which is illegal.

Both hackers claim to be operating on the legal side of the line nowadays – Poulson pitting his wits in the journalistic arena with a weekly column for *ZDTV*, and Dystic performing ‘on-line investigation’ work for various companies. It was interesting to hear how each justified their current status. Even more fascinating was the apparent acceptance of the two individuals and their kind shown by the audience (mostly Information Security professionals). True, many questions were asked, but not one seriously probed either individual’s conscience. As a member of the *Cult of the Dead Cow* (*cDc*), Dystic cites the ‘attitude of denial’ displayed by some software vendors as his, and his organization’s, driving force. Dystic’s justification of his *Back Orifice* activities is that he is striving to force improvement into the security of the *Windows 9x* operating systems; a noble cause indeed. Noble enough, it would seem, to satisfy the audience who, in part at least, expressed gratitude to the *cDc* team for their ‘assistance’ in revealing security weaknesses.

The ‘exposing weaknesses’ justification is an argument that could be misused in many other areas to similar effect. For example, by the writers of viruses that target the VBA functionality of certain applications. How would the same audience react to the smiling face of the Melissa author for example? One would suspect none too favourably – many may have lost a weekend thanks to the incident. Yet, it cannot be denied that Melissa exposed fundamental security weaknesses. Consider also, the victims of data-diddling or email-propagating macro viruses who have had to deal with the potentially embarrassing consequences. They might not be so receptive to the ‘Ah, but I was only exposing weaknesses in products supporting VBA’ claim of the virus writer.

Perhaps the discrepancies in the apparent treatment of hackers as opposed to virus writers is due to the differing perceptions of viruses and overall computer security? Perhaps users feel a few steps removed from the consequences of hackers gaining unauthorized access to the company network, whereas viruses can, and do, inconvenience them with real-time problems at their desktop. If this attitude does exist, then it is slightly concerning.

Further discrepancies in the perception of hackers and virus-writers can be seen by examining their post-teen activities. Reformed hackers are readily employed in network security roles – in fact it would appear they are prize assets for certain companies. The rehabilitation practice is totally accepted. Anti-virus companies on the other hand do not seek to employ those who have spent their formative years writing viruses. It is peculiar therefore, that it is the anti-virus companies who are often unjustifiably accused of perpetuating their industry by writing the viruses themselves.

Fraser Howard

NEWS

FunLoving Criminal?

A new virus which infects *Windows 9x* and *NT* may be in the wild. Reported to several anti-virus companies early in November 1999, Win32/FunLove.4099 has been found at a few sites in Europe and the United States. The original release and distribution mechanism of Funlove is unknown. Add to this the fact that anti-virus updates that detect it are shipping as this issue of *VB* goes to press, and it is hardly surprising that it is not known whether the virus may have spread further.

This virus is also known as FLCSS and FLC. FunLove is of most concern because of its security-lowering payload under *Windows NT* and its deliberate attempts to spread via network shares on *Intel* Win32 platforms.

The virus infects Win32 PE files that are executables (EXE), screen savers (SCR) and ActiveX controls (OCX). It appends its code to the end of the last section in the file, modifies the PE header to reflect this change, and patches code at the original PE entry point to transfer control to the virus. It also alters the characteristics of the last PE section of its hosts.

In this way, FunLove gains control when an infected program is run. It copies its code from the end of the infected host into the file FLCSS.EXE in the *Windows* system directory, then executes that program. With suitable permissions under *NT*, FLCSS.EXE installs itself as a service, set to start automatically. This shows up in the services list as FLC while FLCSS.EXE shows up in the process list. Under *Windows 9x*, FLCSS.EXE runs as a hidden process and is not visible in the task list.

Each drive from C: through Z:, and other accessible network shares, are searched for suitable hosts. These are periodically rechecked for further hosts. Thus, FunLove can spread rapidly through networks with lax file sharing security policies.

The FunLove virus also borrows a trick from some of the *NT* infecting Win32/Bolzano variants (see *VB*, September 1999, p.10). It patches NTOSKRNL.EXE to bypass file permissions security and NTLDR to allow the patched kernel to load. Once a machine is restarted with these patches in place, any user will have full access to all of the files on the machine. These files will have to be restored or re-installed along with files infected by the virus.

Let us all hope that FunLove's appearance at a couple of disparate sites remains something of an oddity. If it has been broadbanded and the source has not been detected yet, Win32/FunLove.4099 could potentially become an additional pain for system administrators with the Millennium rollover pressing down on them. Watch this space for further information on this virus ▮

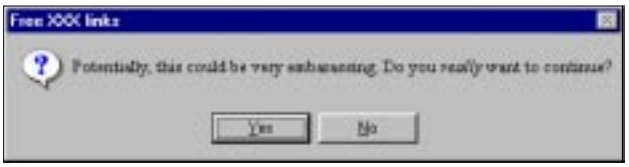
Prevalence Table – October 1999

Virus	Type	Incidents	Reports
ColdApe	Macro	890	40.9%
Ethan	Macro	212	9.7%
Win32/Ska	File	209	9.6%
Laroux	Macro	132	6.1%
Thus	Macro	121	5.6%
Marker	Macro	90	4.1%
Cap	Macro	86	3.9%
Class	Macro	81	3.7%
Win32/Pretty	File	59	2.7%
Tristate	Macro	39	1.8%
Npad	Macro	31	1.4%
Win95/CIH	File	22	1.0%
Concept	Macro	20	0.9%
Freelinks	Script	19	0.9%
Melissa	Macro	17	0.8%
Temple	Macro	12	0.6%
Goldfish	Macro	11	0.5%
Shuffle	Macro	11	0.5%
Nottice	Macro	8	0.4%
Parity	Boot	8	0.4%
Others [1]		100	4.6%
Total		2178	100%

[1] The Prevalence Table includes a total of 100 reports across 36 other viruses. A complete summary can be found at http://www.virusbtn.com/Prevalence/.

Practise Safe Computing!

Following its initial appearance in early July 1999, VBS/Freelinks (for a full analysis, see *VB*, November 1999, p.6) has been enjoying something of a second honeymoon over recent months.



Recently, *Data Fellows* became the latest high profile firm to fall victim to this virus. *ZDTV*, itself a recipient of a Freelinks-bearing email from an infected *Data Fellows* machine, reported that the Scandinavian anti-virus company became infected when IT staff momentarily disabled their anti-virus protection while performing routine network maintenance ▮

LETTERS

Dear Virus Bulletin

In with the New?

If you have strong feelings either in favour of or against Eric Chien's suggestion that False Positive testing should be added to the VB 100% criteria (see *VB*, November 1999, p.5), we would like to hear from you as soon as possible.

Taking into account the tide of popular opinion so far, *Virus Bulletin* is currently considering implementing this new and important condition in the year 2000 Comparative Reviews. Please send your opinions, be they for or against this additional test criterion, to editorial@virusbtn.com.

Fraser Howard
Virus Bulletin
UK

Y2iKes!

The end is near, the end is near! We're rapidly approaching the Y2K rollover. And all this hype about Y2K will soon be over! Indeed, we'll deal with actual Y2K problems soon enough. So, I thought I'd jot down some notes and projections for everyone.

First of all, come the New Year, unless you're in the Far East or Down Under, don't think that you're the first one to encounter the Y2K problem that will end the world as we know it. So let us pay homage to our friends and compatriots there as we sacrifice them for the sake of humanity. This illustrates the point that Y2K is not a one-off event, but rather 24 mini-events which will go off separately at different points around the globe.

Next, some random operating system problem is not going to be a Y2K problem. You'd think that all those people testing *Microsoft* OSes these past few years should have encountered any potential scenario with vanilla installations by now. Extend this thought to all the major *Microsoft* packages, like *Office*.

So, what kind of problems will most likely be Y2K-related? How about data transfer processes between two competing packages? Something like *Lotus 1-2-3* reading VISICALC files, especially if a process has been adapted locally.

Oh, you've forgotten about VISICALC? Well, what other out-of-date packages are still floating around your establishment? That's where the problems will be coming from.

One of my favorite comic strips about Y2K says: 'What is Y2K?' – 'It's a problem caused by using only 2 numbers to represent what should have been represented by 4 numbers.' 'Why is it called Y2K?' – 'Because we shortened it.'

So, as people continue to do things like that, we will continue always to have a job. And as the end of the Y2K issue approaches, it just means we'll have other issues to work on, like… viruses! Won't that be nice?

Jimmy Kuo
Network Associates Inc
USA

Net Results

One security magazine named 1999 'the Year of the Virus'. It has been a great year for anti-virus vendors – a dreary one for users. The good news is that virtually every new computer shipping today comes with AV software. In the US, 96% of major corporations have deployed anti-virus products and have an anti-virus policy in place. Every major anti-virus product has been certified by one or more independent organization to detect 100% of in-the-wild viruses. The war is over, we can all get real jobs, right?

Sadly, no. According to the *ICSA*, *Computer Economics* and others, the number of virus incidents, and the cumulative cost of those incidents, keep rising dramatically. Computer viruses, long regarded by the mainstream as little more then a nuisance, cyber-graffiti if you will, are now seen as a major threat – cyber-terrorism in fact.

Frankly, it is amazing that we are not all being sued for false advertising, or worse. It is high time that the anti-virus industry stood back and asked, what's gone wrong? We are fighting the wrong war on the wrong battlefield. 80% of anti-virus budgets are spent on desktop virus protection. However, perhaps 80%–90% of virus incidents originate as files attached to email. The failure of the desktop model shouldn't surprise anyone. Many, perhaps most, users either turn off AV software or fail to update frequently enough. It is almost a cliché among security professionals that any security system that relies on users is doomed to fail.

What's the solution? Some experts and vendors feel that we need a technological fix – either better scan engines or faster methods of distributing pattern updates. This is all great but it fails to address the fundamental issue – users. Besides which, even a cursory glance at the virus prevalence tables shows that the overwhelming majority of virus incidents are due to viruses that have been in the wild and widely detected for months or even years.

Other experts focus on the need to educate the public in safe computing practices. This year alone, thousands of articles have been published about computer viruses and about how to protect yourself. Despite all the press, the virus deluge continues unabated. Again, we shouldn't be surprised. Most people still eat too much, drive too fast and smoke too much even after vast public education efforts.

If technological and behavioural solutions fall short, what can we do? We can think in terms of structural solutions. Currently, for most organizations the first and only line of anti-virus defence is the desktop. It is time for every network to have virus protection at the email server and at the Internet gateway. Ask any military man where the defence perimeter should be – 'as far from me as possible'.

Viruses should be stopped before they get to the desktop, before they get to the email server, before they even get through the firewall. It is time to weave virus protection into the fabric of the Internet. I'm suggesting that virus protection should be a value-added Internet service offered by ISPs and ASPs.

A few innovative ISPs such as *UUNET*, *US West* and *Sprint* in the US and *SEEDnet Linkage* in Asia have started to offer virus free Internet access. More will soon follow suit. Once the majority of ISPs offer virus protection we will have the structure in place to contain virus outbreaks. Viruses may never go away but ISP-based protection will dramatically reduce the number of virus incidents.

For server and ISP-based virus protection to become the norm, two changes need to occur. First, more vendors need to offer carrier class, Unix-based products. Most major ISPs run on *Solaris* and need products that will integrate with their directory and billing systems. Few anti-virus vendors are focusing on what's really needed to run in such demanding environments.

Second, organizations such as *Virus Bulletin* need to change their focus. As far as I know, *VB* has never reviewed email server or firewall-based anti-virus products. Assuming that these products have the same virus detection characteristics as their *Windows 95/98* counterparts is a mistake.

Also, evaluation criteria for email and gateway-based products are dramatically different from those for desktop products. The growth of the Internet has offered virus writers endless opportunities for perfecting and distributing their wares – it also offers us new ways to fight back.

Daniel Schrader
Trend Micro Inc
USA

[*Both your main points are certainly true – we can all agree that users are often the weakest point in any anti-virus shield, and protection at the gateway is becoming more and more important. Despite these facts, the importance of adequate protection at the desktop should not be underestimated. The same military man would surely advocate weapons deployment within the defence peri-meter as an additional level of protection which could ultimately be life-saving.*

As this issue goes to press, Virus Bulletin *is working on initiating the testing of Groupware products in Comparative Reviews. To this end, developers and vendors are invited to contact us to discuss participation. With attention firmly placed upon perimeter-based scanners, and bearing in mind that* Trend Micro Inc *has not submitted a desktop product for some nine months now, I trust that* Trend *will be willing to submit its Groupware products for testing when the time arises.*

Finally, a couple of howlers for the holiday season! Ed.]

Logo-ing Mad?

Has anyone noticed the incredible similarity between the *Data Fellows F-PROT* logo (www.datafellows.com/pics/leftbarlogo.gif) and the logo used by Dr Evil in the block-buster movie 'Austin Powers, The Spy Who Shagged Me'(www.austinpowers.com/EVIL/EVILINDEX.HTML). Check it out.

Concerned readers want to know!

Name and Address withheld by request
USA

Little Green Viruses

I am interested in learning more about Top Secret projects governments might have in place for using viruses as weapons. Ever since watching 'Independence Day', I have been wondering how it might be possible to use a computer virus written on one type of machine to defend us from hordes of invading aliens.

Can you please research the issue and tell us more about what our government is doing to prepare for this eventuality? I thought that it was a pretty good idea… why couldn't the aliens have done the same to us though?

Name and Address withheld by request
USA

VIRUS ANALYSIS 1

Bursting the Bubble

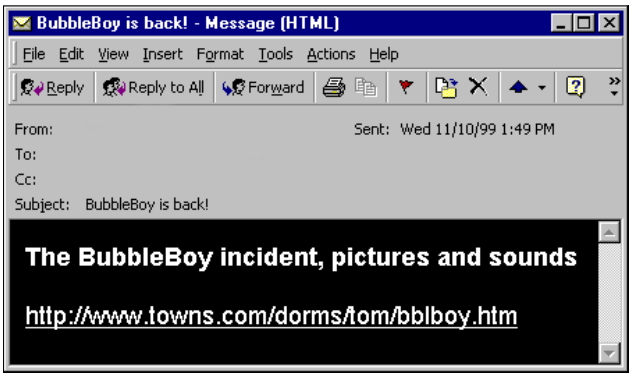
Ian Whalley
IBM Research, USA

In computer security terms, many moons have come and gone since the events surrounding Melissa in March and April of this year. Something new was certainly overdue, and in the week of 8 November 1999, something new arrived in the shape of BubbleBoy. This virus, in the same way as Melissa, exploits *MS Outlook*. However, unlike Melissa, BubbleBoy does not require the user to open a document to run. Just reading an email message is enough.

Yes, you read that correctly. You do not need to open an attachment to get infected.

Arrival

BubbleBoy arrives in what appears to be a standard HTML-enhanced *Outlook* email message, see below.



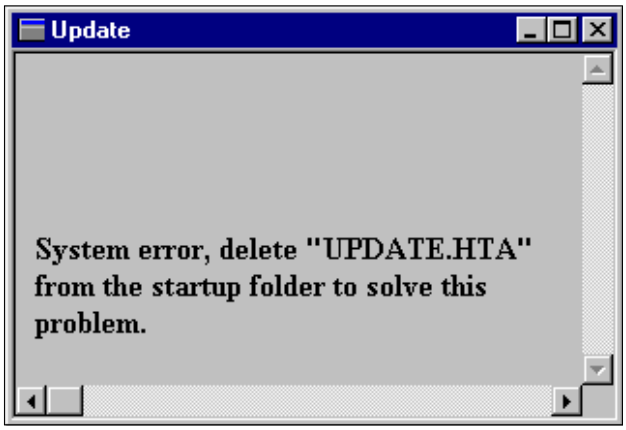
When the user views this message (more on this later), a file called 'UPDATE.HTA' is written into the user's Startup folder C:\Windows\Start menu\Programs\Startup. As one might expect, this will be executed the next time the computer boots, or the user logs on.

It turns out that the extension 'HTA' signifies a 'HyperText Application' file. What this means is that it contains standard HTML code, wrapped up in a proprietary *Microsoft* binary format. There is a program called MSHTA.EXE, which resides in C:\Windows\System, that knows how to deal with these files, and is associated with them by default. That is to say, when an HTA file is executed, MSHTA is invoked to handle the excecution.

In the case of UPDATE.HTA, the HTML embedded within the HTA file in turn contains a section which reads <script language="VBScript"> … </script>. This section will execute when the file is run. Due to the fact that the UPDATE.HTA file was dropped into the startup folder, it is executed as the user logs on. Thus, the VBScript section receives control as the user's session starts, and it immediately creates a WScript.Shell object. The first thing the script does is to set the registered owner and organization values to 'BubbleBoy' and 'Vandelay Industries', respectively. HKLM\Software\OUTLOOK.BubbleBoy is then checked – if it is set to 'OUTLOOK.BubbleBoy 1.0 by Zulu', the payload does not fire.

If this value is not set, BubbleBoy sets it and then creates an Outlook.Application object, and uses it to walk through all available *Outlook* address books. For each address book, it copies all the email addresses into the BCC (Blind Carbon Copy) field of a new email message – when it reaches the end of the address book, it carefully constructs a modified copy of its own code and places it into the message body, and sends the message.

Thus, one message will be sent per address book – but that message will be addressed to everyone in that address book. In addition, this message is displayed to the user.



Clearly the author of BubbleBoy is attempting to pass his creation off as a legitimate system extension, but it is less than convincing. The reason the virus displays anything is that it has to, as it is combined within what is basically an HTML file.

At a high level, that is all there is to it – the virus makes a copy of itself, and sends it to your contacts. However, if it really was this easy, it would have been done before, so where's the hard bit?

The Hard Bit

The difficulty is getting control without warning the user. *Outlook* is supposed to prevent email messages from performing 'dangerous' operations – this is, of course, correct behaviour. However, VBScript in email messages *is* permitted to perform certain operations, including accessing ActiveX controls marked 'safe for scripting'. 'Safe for scripting' is a *Microsoft* euphemism for 'this control can't do anything dangerous'. Controls marked 'safe for script-

ing' may (in general) be used even by remote scripts. The issue is further complicated by the existence of the Internet Options/Security settings under Control Panel.

All levels in here are customizable, but the default security level for the Internet Zone is 'Medium', which allows use of controls marked 'safe for scripting'. If the Internet Zone is set to 'High' security, then such controls cannot be used.

Unfortunately, at the end of August 1999, a vulnerability was discovered. An ActiveX control – 'scriptlet.typelib' – was erroneously tagged 'safe for scripting'. In fact, this control allows the caller to create, modify or delete files on the local filesystem. The fact that it was marked 'safe for scripting' meant that any remote script could do those things as well, simply by using the scriptlet.typlib control.

This was clearly a 'Bad Thing' and so a patch was released that removed the 'safe for scripting' tag from this control. Http://www.microsoft.com/security/bulletins/ms99-032.asp has more information.

However, most users have far better things to do than check *Microsoft*'s site for updates, and do not allow *Windows Update* to automatically check *Microsoft*'s site for updates either. The percentage of affected users that download patches for security flaws is something which we cannot know – however, what is certain is that if a computer is not patched against this attack, and the Internet Zone security setting is at 'Medium' or lower, that computer is vulnerable to BubbleBoy.

Complications

There are, in fact, two versions of the BubbleBoy virus known at the time of writing – 1.0 and 1.1. The second is distributed in scrambled format, via 'VBScript.Encode'. Aside from using a different identification string in the registry, (perhaps unsurprisingly, 'OUTLOOK.BubbleBoy 1.1 by Zulu'), it appears identical.

The reason BubbleBoy drops UPDATE.HTA into the local Startup group and allows it to be run at next logon is simple – it has no option. The scriptlet.typelib flaw only allows the script running within the email to modify the filesystem, it cannot do anything else. If it attempted to send the email messages immediately, it would fail due to *Outlook*'s security checks.

Instead, the virus drops a file containing that code into a location where it will be executed later by the system – namely, the user's Startup folder. When it runs from that folder, the system (as one would expect) runs it as a local file, and it has a much wider degree of freedom to do dangerous things.

Are You at Risk?

It is important to note that at the time of writing, this virus has not been seen in the wild, so certainly the risk at the moment appears to be low. In addition, BubbleBoy will not work unless you have *Windows 95/98* with *Windows Scripting Host* installed – *WSH* is built into *Windows 98*, but is available for *Windows 95*. In addition, *Internet Explorer v.5* and *Outlook* are required.

The virus does not run under *Windows NT* (due to the fact that it uses a hardcoded path to the Startup folder). For the same reason, it will not work on *Windows 95/98* when configured for multiple users. Also, the virus cannot run if you have installed the *Microsoft* patch against the vulnerability BubbleBoy exploits, as described above.

BubbleBoy will not work correctly with *Outlook Express*. While it will be able to drop UPDATE.HTA in this environment, and this file will execute, it will not be able to send the email messages with *Outlook Express* – for this, it requires the full version of *Outlook*, just as Melissa does.

Having said that, it is worth noting that under *Outlook Express*, UPDATE.HTA is dropped when the message is previewed in the 'Preview Pane'. Under the full version of *Outlook*, the message has to be viewed in its own window (you have to double-click on the message in the subject-line view). BubbleBoy requires *Outlook 98* or *Outlook 2000* to work correctly.

The Morals of the Story

What lessons does BubbleBoy teach us, the users? The most obvious one is that we should check *Microsoft*'s Web site and/or security-related mailing lists for fixes, and install them in a timely fashion. How you do this is up to you – you can either check manually, or use *Windows Update* to check for you.

Unfortunately, it seems dangerously inevitable that more bugs along the same lines as the one exploited by BubbleBoy will come to light. Of all the hundreds of ActiveX controls installed on the average modern *Windows* machine, how many of them are incorrectly considered by the system to be 'safe for scripting'?

VBS/BubbleBoy	
Aliases:	None known.
Type:	Worm.
Spread:	Via email. Expoits known security flaw in IE 5 to gain control when email is viewed.
Self-recognition:	Registry values (see text).
Payload:	Sends infected email to all members of all available *Outlook* address books.
Removal:	Delete infected mail without viewing. Remove UPDATE.HTA in Startup folder (see text for details).

VIRUS ANALYSIS 2

One Sharp Corner

Katrin Tocheva
Data Fellows, Finland

After unsuccessful attempts at cross-application by macro viruses (e.g.Cross and Teocatl), Shiver (see *VB*, October 1998, p.9) became the first macro virus able to infect both *Word 97* and *Excel 97* successfully. At the beginning of this year Tristate – capable of infecting *Word 97*, *Excel 97* and *PowerPoint 97* was discovered (*VB*, March 1999, p.10).

Unlike Shiver, which uses Dynamic Data Exchange (DDE), Tristate uses the Component Object Model (COM) feature also known as ActiveX. This feature allows one application to be accessed from another and is used in many other viruses. Most known mass-mailers like Melissa (*VB*, May 1999, p.5) and Freelinks (November 1999, p.6) use it to open *Outlook*. Many VBScript viruses and droppers like Loud, Hopper and Break (*VB*, March 1999, p.6) use it to infect *Word 97* from Visual Basic Scripts.

The First MS Project Virus

At the end of October 1999, *Data Fellows* received Corner, the first virus capable of infecting *MS Project 98*. This application supports Visual Basic for Applications like *Word 97/2000* and other *MS Office* applications. That is why anti-virus researchers anticipated the appearance of this virus and were not too surprised. Some products already implement heuristics for it.

The creation of new kinds of viruses is always a proof of concept for a virus writer. Creating a virus for an application like *Project 98*, which is not particularly widely used, means that this virus will not often be seen in the wild. Maybe that is why Corner was made as a cross application infector. It infects *Word 97/2000*, the most popular *Office* application of all. *Project 98* after all is not even included in the *Office* package.

The sample we received is a *Word 97* document. As a *Word* infector, the virus code inside is a simple class virus but not only that, it also contains additional code that infects *Project 98* files. Like Tristate and some other viruses, Corner uses ActiveX technology to gain access to objects from an application.

The Corner virus consists of one module that contains two subroutines – one for each application. *Word's* subroutine is named differently in the global template and in documents. The name of the *Project* subroutine depends on where the virus infects from (either *Word* or *Project*) and is the infected file document, global template or project. Each subroutine executes when an infected file is accessed in the appropriate application.

Infection via Documents

In infected *Word* documents, Corner resides in the first class module (usually named ThisDocument), which consists of two subroutines – one per application. The *Word* subroutine uses the Document_Close event handler, and so when an infected document is closed, infection of the *Word* and *Project* environments is attempted.

Upon closing a document, Corner first checks the registry and lowers the security settings of *Word 2000* and *Excel 2000*. The virus is not able to infect *Word 2000* from an infected *Word* document if the security setting is 'High' (default mode) because *Word* does not execute macros in that case. If the security setting is 'Medium' or *Word 97* is being used, then the virus code executes. Corner removes the Tools/Macro menu to hide its code. After that, it disables the built-in macro-content warning. Following this, Corner executes its infection routine which is in two parts – the first infects *Word* and the second infects *Project 98*.

Infection of Word's Global Template

As mentioned above, the *Word* part of the virus is a simple class infector. It checks if the global template is infected by searching for a marker (a single apostrophe-style comment) in the second line of the ThisDocument module. If this marker is not found, the virus deletes the entire contents of the module. Then, using the InsertLines command, Corner transfers the virus code from the active document's module to the global template's one.

At that time the first and the 41st line, which contain the names of *Word's* and *Project's* subroutines, are changed. The virus inserts a string into the first line to change the name of *Word's* subroutine in the infected global template to 'Document_Open'. Similarly, *Project's* subroutine in the 41st line, which is named 'Sub projcloser' in the ThisDocument module of infected documents, is renamed to 'Sub wrdcloser'. Later, when the virus infects *Word* documents, it changes the name of both subroutines back.

The names of the subroutines in the global template and infected documents, is summarised as follows:

Infected documents:

Module: ThisDocument

Word's subroutine: Document_Close

Project's subroutine: projcloser

Word global template infected from a document:

Module: ThisDocument

Word's subroutine: Document_Open

Project's subroutine: wrdcloser

Infection of MS Project

The next part of the code is that which infects *Project 98* from *Word*, using the GetObject ('MSProject.Application') function. In order for infection to occur, this function requires that *Project* is running at that time. If it is running, Corner adds a blank project. After that, again using the InsertLines command, it transfers the contents of *Word's* ThisDocument class module to the blank project's ThisProject module (this module exists in every project). Then Corner changes the 41st line (the name of *Project's* subroutine) to 'Sub Project_Deactivate'. Thereafter, all deactivated projects during the current *Project 98* session will become infected. The subroutines in infected projects are named as follows:

Infected projects:

Module: ThisProject

Word's subroutine: Document_Open

Project's subroutine: Project_Deactivate

Infection via Projects

In infected projects, Corner resides in the first class module which is usually named ThisProject, and contains two subroutines. The *Project 98*-relevant code is stored in the subroutine named Project_Deactivate. Thus, when an infected project is opened, all subsequently accessed projects will be infected during deactivation.

When the virus activates from an infected project, it first disables the macro virus protection. Then it executes the infection routine in two steps – to infect projects and to infect *Word* (if it is not infected yet).

Infection of Projects

When a project is deactivated Corner checks for the infection marker in its ThisProject class module. If it is not found, the virus deletes the contents of the module, and inserts its code there. During this infection process Corner does not change any subroutine names and so it looks the same in all infected projects.

It does not infect GLOBAL.MPT (*Project 98's* global template) and therefore has no form of 'residency' in the *Project* environment – only opened projects will be infected. This infection mechanism in *Project* does not make this virus very viable and it is unlikely that it will ever become widespread in the wild.

Infection of Word

The next subroutine is that which infects *Word* – relevant if the virus enters the system via an infected project. To infect *Word*, Corner checks that *Word* is running and if it is not, opens it using the CreateObject ('Word.Application') function. So, unlike the corresponding *Word* to *Project* infection channel where *Project* had to be open, in this case the virus does not need *Word* to be running in order to infect it. The initial infection mechanism of *Word* from *Project* is similar to that described above but the names of the *Word* and *Project* subroutines are 'Document_Open' and 'Closer'. The infected *Word's* global template looks different to that infected via a *Word* document:

Word global template infected from Project 98:

Module: ThisDocument

Word's subroutine: Document_Open

Project's subroutine: Closer

Importantly, Corner can infect *Word 2000* from projects even if the security settings are 'High'. This is because the virus opens *Word* using the CreateObject function and transfers the virus code from the ThisProject class module to the global template's ThisDocument class module. With this 'user-invisible' operation Corner infects the global template irrespective of the security settings. Subsequently, upon loading *Word 2000* (with its now-infected global template), the first thing the virus code will do is lower the security settings. Thereafter, all opened documents will be infected, the user still under the illusion that *Word 2000* is set to deny the execution of any macros.

These lyrics from a Joy Division song on their album 'Closer' are included in the virus code but never shown:

'I never realized the lengths I'd have to go
'All the darkest corners of a sense
'I didn't know
'Just for one moment
'Hearing someone call
'Looked beyond the day in hand
'There's nothing there at all
'Project98/Word97-2k Closer

Corner is the first virus to infect *MS Project 98*, and is able to cross-infect between this application and *Word*. The availability of the source code coupled with the fact that *Project 98* supports VBA, suggests that we may see more such viruses soon.

P98/Corner

Aliases:	Closer.
Type:	Cross-application macro virus.
Infects:	*Word 97/2000*, *Project 98*.
Removal:	*Word 97/2000* – delete Normal.dot to clean the environment; delete the contents of the ThisDocument module from infected documents. *Project 98* – delete the contents of the ThisProject module from infected projects. Restore security settings of *Word 2000* and *Excel 2000* using Tools/Macro menu.

VIRUS ANALYSIS 3

Merry MMXmas!

Snorre Fagerland
Norman ASA, Norway

In August we saw the first polymorphic virus which used MMX instructions – W95/Prizzy. Some time later came a Win32 virus with the same functionality, only with considerably fewer bugs. This is W32/Legacy.

This virus contains a heap of tricks, many of which have been used before in other viruses. It is clearly another example of virus authors exchanging information.

Polymorphism

Legacy contains some simple entry-point-obscuring code; not very advanced compared to the mid-infecting capabilities of a virus like W32/Bolzano. The entry point of infected files points to the original code section of the infected executables, where the original code has been replaced with a polymorphic piece of code intended to throw off emulators.

Inside this code piece is a structured exception handler (SEH) trap. The idea is to set up an exception handler, and then raise an exception that will transfer control to this handler. This technique (or something similar) has been used in DOS viruses for a long time; it was a natural development for it to appear in 32-bit code as well. Unfortunately, many emulators fail to emulate an exception stack in Win32 properly, so the trick will work in many cases. After the SEH trick the virus jumps to its main virus code.

Most of the virus itself is located at the end of infected executables. It is either placed in the .reloc section, which is overwritten and given a random name, or placed at the end of the last section if no .reloc section is present. Legacy avoids any problems connected with overwriting the .reloc section by nullifying the fields in the header defining any fixups. This code is doubly encrypted. The outer layer is polymorphic, while the inner layer is static. Not far into the decryptor comes a cpuid opcode. The return from this opcode (bit 17) tells the virus whether the CPU supports MMX or not.

MMX, or Multimedia Extensions, is an architecture that was first introduced in the so-called 'Pentium with MMX technology' and included as standard from the *Pentium-II* processor. It consists of 57 extra opcodes specially directed at graphics handling. In addition, MMX defines eight new 64-bit registers named mm0 to mm7, that can be manipulated by these opcodes. As was the case with the Prizzy virus, Legacy tests for the presence of MMX before attempting to use it. If MMX is not present, it will default to a simple XOR loop.

The MMX instructions are not only used as garbage, they are used in the loop for the actual decryption by PXOR-ing two MMX registers. This is a 64-bit XOR operation but only the low DWORD is used in the decryption. It is easy to imagine these instructions, so directed towards bit manipulation, involved in polymorphic viruses in many other ways in the future.

An interesting additional feature is that every subroutine in this virus is encrypted, and is decrypted before first use. The encryption in these routines is different than the start-up polymorphism. Instead of a linear decryption with a fixed key, the decryptor attempts to decrypt with new keys until it is successful. Since the encryption key is only a byte, and maximum tries before success is 255, there is no notable delay.

Initialization

The main functionality of this virus is divided into five (or rather six) threads, but before starting them up, it has to set up the API calls it will be using. It does this by locating the image of KERNEL32.DLL in memory and doing a search for the APIs it wishes to use. As is the case with several other viruses, it does not search for the routine names themselves, but instead for routine name CRCs. The Kernel32 base address is found either by static offsets for the *Win9x*/*WinNT*/*Win2k* kernels or preferably by searching for the file in the default exception handler address space.

The virus then increases its own execution priority for a short time while it runs its main thread. The main thread sets up and controls the six sub-threads. The first five threads are initialized at the same time, while the sixth is not started until the fifth has finished.

The sub-threads, described below, perform the infective action. However, the main thread still has some things to do. After the sub-threads have finished their tasks, the main thread is free to continue. The first thing it does is to turn its priority back to the original level since the work-intensive actions are finished. Then it checks if it is time to activate the payload. If not, it copies the 256 host bytes back to the original code section and returns to the host. The payload is described later.

Legacy's Threads

One thread is dedicated to anti-debugger tricks. Several methods are used to determine if a debugger is present. One is to try to open SICE or NTICE on root; a request that will be successful if *SoftIce for Win95* or *NT* is running. An alternative method is to see if there is a debugger context at FS:[20h], and the third method is to use the API call IsDebuggerPresent. This function was included from

Windows 98, so it does not work under *Windows 95*. If any of these tricks returns true, the thread leads directly into a loop that will freeze the machine.

The anti-monitor thread is set up to close some resident anti-virus monitors ('AVP Monitor' and 'Amon Antivirus Monitor') by posting a WM_QUIT message to them. The FindWindowA function is declared out of USER32.DLL for this purpose. Additionally, integrity data files from numerous anti-virus products are deleted by the anti-integritycheck thread.

The per-process residence thread redirects several Win32 file oriented functions (see below) to the virus by patching the KERNEL32 exports in memory. Valid executables (*.EXE, *.CPL, *.SCR) that are accessed with these APIs are infected. If the file accessed is a *.RAR or *.ARJ archive, a virus dropper will be inserted into it. The dropper will be uncompressed – only the archive header is updated to show the new addition.

The infection thread needs some initial information, which is provided by a pre-infection thread. Why this thread is singled out as a separate one, and not kept as part of the initialization code probably had to do with timesaving considerations. The functionality that this thread is supposed to provide is plainly to get to the *Windows* directory, system directory, and the current directory. These, and subdirectories, will be targeted for direct-action infection.

As soon as the previous thread has completed, the infection thread is initialized. It will search through the aforementioned directories hunting for eligible files and infecting them. The files are mapped to memory before infection. As mentioned before, 256 bytes from the start of the code section is replaced with some virus code. The original code is stored inside the main virus body. The rest of the virus is written at the end of the host, and the infected file's CRC is recalculated and inserted in the header.

Legacy does not preserve the original datetime stamp, and it will infect read-only files. The virus checks for its own presence in infected files by looking for the string LGCY in the reserved field at PE+4Ch. The Legacy author's earlier creation, W32/Thorin, has the string THRN in the same area. This seems to be his standard marking technique.

Just as with the per-process resident thread, *.RAR and *.ARJ files are searched for and 'infected' by a dropper. It consists of (LEGACY.TMP), a semi-compressed dummy host file which is decompressed, dumped to disk, infected and stored in the archive.

The infected dummy is erased afterwards. The compression used in the dummy host is not very advanced; it basically has multiple consecutive zeros removed. It contains the text 'PRON – XXX SeaRCHer [FaTaL eRRoR!!!] Unable to initialize search engine' and 'Unknown error at address BFF79463'. The Legacy virus avoids 'reinfecting' archives by looking for the string LG in the header entries.

On 31 July the virus activates its payload. It consists of the dialog box shown here. In addition, it will enter the following text into the registry key. This renames the 'My Computer' zone to the virus name.


```
…\Microsoft\Windows\Currentversion\Internet
Settings\Zones\0\Displayname =
[Win32.Legacy.15707 v1.00].
```

Conclusion

Win32/Legacy is not spectacular but it is one of a series of rather highly developed viruses released lately. The extensive use of MMX and floating point opcodes is something we are going to have to get used to, so emulators that are not ready yet have to hurry. The code also shows a certain degree of understanding of Win32 architecture, sufficient to be able to locate important structures in memory without relying on hardcoded offsets and undocumented calls and structures.

The modular architecture, where every function has its own exception handler, shows virus authors intend to make their creations more stable than before. Some functions will not work everywhere – not all function calls are defined for all Win32 platforms. This virus handled such occasions gracefully; and it did not crash on the test platforms.

Win32/Legacy

Aliases:	None known.
Type:	*Win9x/NT/Win2k* PE infector.
Intercepts:	Per-process residency hooks the following APIs; MoveFileA, CopyFileA, GetFullPathNameA, DeleteFileA, WinExec, GetFileAttributesA, CreateFileA, CreateProcessA, SetFileAttributesA, _lopen, OpenFile, MoveFileExA and CopyFileExA.
Infects:	*Windows* PE executables with EXE, SCR and CPL extensions. Inserts infected droppers into RAR and ARJ archives.
Removal:	Restore infected files from backup. Delete infected droppers from archives.

CASE STUDY

Following the Breadcrumbs

Christine M Orshesky
i-secure corporation, USA

Many people and organizations are still puzzled about where viruses come from and how they get into their systems. Traditional wisdom about not opening suspicious messages from unknown individuals seems to have given the impression that viruses come from strangers – either those who unknowingly forward them or those who actually want to do harm.

Is it really the strangers that bring or send the viruses and other suspicious things into your organization or is it your trusted employees, co-workers, and professional acquaintances? This has to be a key question when trying to determine effective ways to protect your organization and its resources.

One such organization, *Anycompany.com*, decided to find out how and where the viruses in their organization were being obtained so they could do more to protect their environment. 'Anycompany.com' is a pseudonym given for purposes of this report to an actual organization where the following research was performed. *Anycompany.com* is a very large organization with over 40,000 systems in its decentralized computing environment and a diverse population of over 20,000 employees spread over various departments and networks. This article is a corporate case study on the process *Anycompany.com* used to trace its virus problem and from where it hailed.

Anycompany.com started by researching some of their virus infections to see if there was any way to trace back to the first introduction of a particular virus into their environment. They reviewed their incident tracking information and discovered that they had experienced several instances of the W97M.Marker virus, which maintains a log of all systems that it infects – a travel itinerary of sorts for the virus. So, they began to follow the breadcrumbs that Marker left for them.

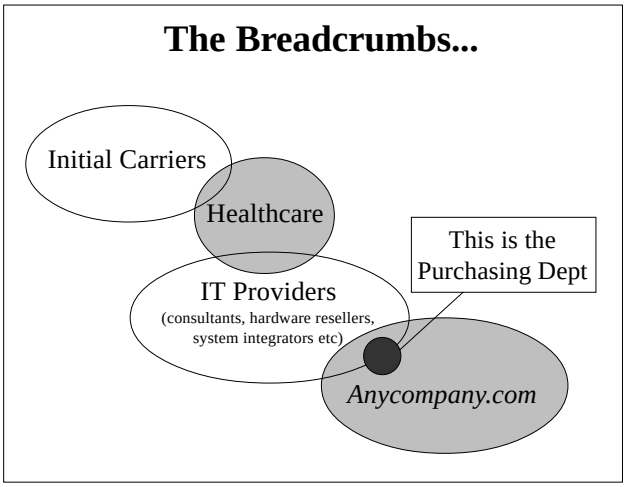
Collecting and Analysing the Breadcrumbs

Anycompany.com had been finding it difficult preventing the Marker virus from infecting their organization. As in many organizations, *Anycompany.com* had found it hard maintaining and updating the anti-virus protection on the desktops across multiple networks. It should be noted that the basis of the information and samples for this case study were taken in February and March 1999 prior to the W97M.Melissa virus infection that opened many organizations' eyes to the need for anti-virus protection for their email servers. As a result, the desktop was the primary defence or protection point.

To help determine the scope of their infections and to track down systems which were infected, *Anycompany.com* specifically blocked all outgoing traffic to the FTP site to which the Marker virus attempted to send the log file. *Anycompany.com* then monitored for systems attempting to access that FTP site. Even though the FTP connection was never successful, *Anycompany.com* did not want its systems attempting to make the connection since it could alert the maintainer of the FTP site to their repeated infections and expose a vulnerability.

A daily check of all systems attempting the FTP connection was performed and the 'offending' systems were identified. For each system identified, *Anycompany.com* alerted the appropriate system administrator and the system was reviewed, the anti-virus protection updated, and the system cleaned. The administrator was also asked to submit a copy of the log file that the virus created to *Anycompany.com*'s information security staff before removing it from the system. At the end of one month, *Anycompany.com* had collected fifteen distinct samples and began its analysis.

Anycompany.com took the samples and began breaking down the entries. The logs provided names, organizations, and dates for each infection captured in the log. While some entries were not complete, the entries were sufficient, based on the information in the system registry that was captured by the virus, to begin establishing patterns of travel. *Anycompany.com* was able to construct diagrams of each infection path and show intersections between the paths, including types of organizations that had been infected. The diagram below is a graphical representation of the relationships between the various entities that became infected by the Marker virus in *Anycompany.com's* samples. This kind of exercise, and graphical tool, helped *Anycompany.com* to identify their relationship with the others in the log and the path the virus had taken to reach their organization.



Following the Breadcrumbs

Anycompany.com found during its analysis that, while many of the log files had a variety of individuals listed, there were some very clear patterns in the travel of the virus and the types of organizations the virus had infected before it arrived in their systems.

The following points illustrate some of the observations consistent in all of *Anycompany.com's* samples as suggested by the relationship representations shown in the diagram.

- The virus had infected the same four initial systems in each sample – captured in the "Initial Carriers" set in the diagram.
- The virus travelled from the 'Initial Carriers' to at least three other systems before reaching a system in the healthcare industry.
- Once the virus infected a system in the healthcare industry, it moved throughout the organization and infected at least one other system in the same organization but at a different geographical location.
- The virus travelled from the healthcare industry to various information technology (IT) providers, including consultants, system integrators, and hardware resellers.
- The virus infected various systems throughout the IT providers and then infected individuals related to both the purchasing departments and vendors for *Anycompany.com.*
- The relationship between *Anycompany.com* and its purchasing departments, vendors, and its own IT providers was the point of entry of the virus into *Anycompany.com* systems.
- Once inside *Anycompany.com*, the virus was able to infect multiple systems throughout various departments and networks.

Finally, *Anycompany.com* were able to determine through their analysis that the virus did not in fact arrive from unknown individuals or individuals with malicious intent. Rather, their analysis clearly demonstrated that the infected items arrived through their electronic mail communications with and between other organizations that they had 'trusted' relationships with such as their employees, consultants, IT providers, and vendors.

It should be noted this company had no direct relationship to the healthcare industry. Nevertheless, *Anycompany.com* was able to ascertain that this recurring development was an important discovery, since it suggested that the health-care industry's connection with their IT providers was the point of entry for the virus. This also served as a poignant reminder to *Anycompany.com* that they were exposed to the same things as their direct connections. This was especially true if the direct connection, in this case the IT provider, did not have adequate anti-virus protection and acted as an unknowing conduit.

Learning Lessons

While the analysis *Anycompany.com* performed was with the W97M.Marker virus, they reviewed their findings and conclusions in conjunction with many of their other experiences with virus infections, such as the viruses and worms that perform mass mailings. These mailings provided similar results when looking at where the email attachment or message originated. Again, the infected items were not received from unknown entities but rather individuals and organizations where there was an existing and trusted relationship.

Given that the viruses and other 'Bad Things' can be shown to be coming from 'trusted' people and organizations, how does an organization or individual protect against infection? Like many organizations, *Anycompany.com* exchanges documents and other files frequently over its email systems and it has become a familiar way to share and disseminate corporate information.

As a result, traditional practices and guidance provided to computer users, such as being wary of things arriving from unknown sources or blocking incoming traffic from sites and organizations known to be spreading infected items, are obviously no longer sufficient or acceptable.

If the patterns that *Anycompany.com* saw in its environment are representative for organizations at large, digital signatures and encryption will not offer the level of protection currently purported. Digital signatures and encryption are a good way for an organization to establish a level of assurance that the item received is from a specified individual or organization and that the item has not been tampered with since it was signed or encrypted.

Neither of these technologies, however, do anything to establish that the item was not infected before it was signed or encrypted. So, these technologies will simply make it easier for the infected item to infiltrate your organization and more difficult for you to educate your users and your applications about when to trust and when not to trust.

In essence, for a corporate organization to protect itself, a variety of anti-virus protection strategies must be implemented and maintained at the points of entry to the business, thereby providing layers of defence. In addition to this, user education becomes even more crucial in order to keep users aware of the changing threats and ways that the infected items will arrive. Remind your users that items received from so-called 'trusted' individuals should also be treated as suspicious until they are found to be free from any infections. 'Trust' should no longer be viewed as the default answer.

Based on *Anycompany.com's* real-life experiences and analysis, the moral of this corporate case study seems to be that the ones you 'trust' in your business actually pose your biggest threat. Something to think about finally is the fact that once you trust one entity, you inadvertently inherit everyone they trust.

INSIGHT

Counting the Costin of AV

Costin Raiu was born in the city of Bucharest, Romania, in 1977. He is representative of the hundreds of younger generation anti-virus researchers who are still relatively unknown, despite working for internationally recognised AV companies, and developing countless anti-virus software packages.

Getting Started

His first computer was a Z80 clone, with 4 KB of ROM and 64 KB RAM. 'With this computer I started learning Basic, then when Basic became too slow, I started hacking into the mysteries of Z80 machine code. It was very exciting to discover how programs worked, and to learn new optimization tricks,' he enthuses. 'At that time, many Z80 machines were real masterpieces with only 48 KB but soon the Z80 clone, with most of its mysteries uncovered, became uninteresting. So then I got hold of a book of 8086 assembler from a friend and I was amazed at the possibilities opened up by 16-bit processing compared to the old 8-bit architecture of the Z80.'

It was at this time that Raiu's school received a donation of five computers, more precisely four 286s and one 486SX server running *Novell NetWare*. One day, problems started to show up in the school network. It soon proved to be a computer virus – one that we now know by the name BadSectors.3428. Using his assembler skills, Raiu spent a day and half a night analysing the virus and writing a small disinfection program for it, both in files and memory.

'Soon I become interested in this problem and I started getting more viruses from my friends,' he recalls. 'I continued developing disinfection programs for other common viruses we had in Romania, such as Jabb.1000, Jinx.846, _1963 and Eddie.1800, on my home computer which by then was a 16MHz 286 with 40MB HDD.'

When he had managed to write around 17 different virus disinfection programs, he realised that writing a program for each virus was inefficient. The solution seemed plain, 'I merged them together in a larger program, which I named Main Scan (MSCAN), and made it freely available for download on some Romanian BBSs. The program gained popularity, and users all over the country started to call me on the phone to ask questions or provide suggestions for new versions.'

In the summer of 1994, a small, local company named *GeCAD srl* called Raiu and asked for a commercial version of his program to sell. The new version was named *Romanian AntiVirus (RAV)* and started at version 3.0, as *Main Scan* was already at version 2.0.

Getting Hired

Raiu thinks that in 1996 boot viruses were the main problem; 'In fact, back then I wrote a large number of utilities and tools to detect and remove stealth boot viruses, as well as heuristics dedicated to boot viruses. However, boot viruses soon started to become less widespread as floppy disks became obsolete amid the exponential growth of networks and the Internet.'

In 1997, everyone was faced with a new problem: macro viruses. Raiu could have missed out on this opportunity; 'At first, we nearly assigned someone else to do macro virus research at GeCAD, since I was too busy with my work, but that didn't happen. I had a certain feeling about macro viruses, and decided to research them for myself, and to add detection and disinfection for them to my product.' He began this task somewhat isolated from the other AV researchers around the world because he did not know of too many people from other companies that were doing the same thing at that time. 'The only one who used to reply to my messages was Eugene Kaspersky from AVP.'

The first macro engine was ready in the summer of 1997 and even Raiu admits it was fairly primitive. However, he continued to develop it, first using *Microsoft* OLE32 as an interface to the OLE2 compound files – 'I didn't have any information on OLE2 files back then, so I had to put the engine into production somehow'. Unfortunately, *Microsoft's* implementation of OLE32.DLL was rather unstable and buggy. It was not able to handle damaged documents, crashing several files, and was not portable on DOS.

His early work with macro viruses was the start to an interest in this subject which ended with Raiu making it his primary field of activity. He remembers it well, 'Thus, I witnessed the first Excel virus, the first Word 97 virus, the first Excel formula virus and the first Access virus, and wondered to myself when will this ever end?'

Getting Known

During 1998, Raiu also wrote an analysis of the first Java class infector, Java/StrangeBrew, which was published in *Virus Bulletin* (see *VB*, September 1998, p.11). 'Eugene Kaspersky brought this virus with him on his trip to Romania, and I spent a day and a night after he left analysing it and writing the description' says Raiu.

'I was so afraid someone else would do it first, and would publish the analysis before me! I think Symantec had a description ready by the time my analysis was complete, but it was not as detailed as mine.' StrangeBrew was Raiu's second article to be printed in *VB*, after his overview of the Romanian virus scene entitled 'Between East and West' (June 1998, p.8). He is now a regular contributor.

Raiu returned to his macro virus research. Meanwhile, his interest in Java development grew and he wrote 'The Gatherer' project to monitor IRC for Worms and malware. 'The Gatherer' was a Java program, designed as an 'automotive' IRC client, which collected files sent to it via the DCC protocol. The project was up and running for around two months, but it had to be terminated because it 'ate up' too much bandwidth, from the already limited available Internet connection. 'Right now, it's just another sub-directory on my work HDD, among hundreds of other projects I wrote' says Raiu.

In 1999 he started working on the new architecture of *RAV 8*. This architecture finally ended up as the subject matter of a paper that was scheduled for presentation by the *RAV* team member Adrian Marinescu at the 1999 *Virus Bulletin* conference in Vancouver. Taking some of the time dedicated to macro virus research, Raiu designed the Unified File System component of the *RAV 8* architecture, which allows handling of all kind of file systems, such as native disks and archives in a common, simple way through plug-ins. 'Flexibility was what I had in mind as my primary focus', he recalls.

At the same time, he began work on his own VB'99 conference paper. His focus was Win32 BackDoors and their detection by heuristic methods. Much of his Unix experience can be seen reflected in the context of that paper, which is featured on the conference CD (see p.3 of the November issue).

Getting Ahead

Outside the office, Raiu is just like any other 22 year-old guy and enjoys the same things they do. 'I like to spend my time among my friends, reading books or listening to high beat music styles, such as rave music'. He also considers himself a fan of Frank Herbert's work, especially the *Dune* cycle, which he regards as 'a true masterpiece in the history of Science Fiction'.

He also confesses himself a fan of *Beck's* dark beer, but laughs that he does not nearly match the drinking ability of some of his Russian friends! Among his other interests are chess, large number arithmetics and good, old movies.

Raiu is vague about his future plans. 'It's too early to make a prediction. However, I will probably stick to anti-virus research for some while, it is still the thing I know how to do best.' He is also interested in data security and data recovery but they are not his primary fields. 'They are not as interesting as anti-virus research, but everyone eventually gets bored doing the same thing every day, so maybe in 10 years from now I'll be working as a computer security consultant, or recovering data, which is a problem unlikely ever to end.

'The future looks cloudy for anti-virus companies as well,' considers Raiu. 'The anti-virus industry has become very powerful in the past few years. Soon, small companies will have a hard time surviving the aggressive market standards pushed to the fore by large vendors. The number of companies that develop anti-virus products will become small, then the large ones will split into small ones in a perfect cycle.'

Raiu's view of the AV product of the future is primarily focused on a global solution for the customer, ranging from desktop protection through server protection to technical support and interaction between the developer and users. 'Although still young,' he says, 'the anti-virus industry is starting to become old. What we really need are new ideas. Unfortunately, in this respect we are not doing very well…'

Not surprisingly, he has strong views on virus writing too, but his opinions reflect his age – he believes that young virus writers will burn out their obsession in time. 'I personally regard virus writing as something bad for everyone. Look at it as some kind of disease, but one which eventually gets cured by the organism itself. No one writes viruses for a lifetime, or at least I hope no one will. Unfortunately, this disease is hard to cure, and unlikely ever to cease to exist.'

Unfortunately, the Romanian laws against virus writing are woefully inadequate. 'We can't do a thing about anyone who writes a virus and distributes it widely through email or shareware programs. The problem is not understood by the authorities. The only lawful case we can bring against virus writers is when they cause data damage or loss directly as a result of their program.'

Currently, no virus writer has ever been convicted in Romania for either writing viruses, damaging data or causing data loss. There are plans for that to change, but it is not likely to happen very soon.

Until then, he is happy to continue doing his work, and remains challenged on a daily basis. 'There are enough viruses left to analyse, elegant solutions to replace the current implementations and dark areas to uncover in this field,' reflects Raiu. 'I consider myself a very privileged person to be able to work in the front line of such an interesting area of research. The battle we're fighting here will probably never end – it is, however, our duty to keep the advantage on our side.'

FEATURE

French Connection II

François Paget
NAI, France

Back in August 1997 I gave you an overview of the viruses in France from the end of 1992 to the beginning of 1997. Now I would like to discuss developments since then.

In 1995, before the macro virus infestation, we estimated that 2% of all French microcomputers had undergone a viral attack during the year. In 1995, the most common alerts only affected between one and three machines. Computer viruses have become so familiar that we no longer bother to count them.

That said, the virus phenomenon is taking a worrying turn for all heads of French businesses. *RECIF* (*Recherches et Études sur la Criminalité Informatique Française*) reports that one machine in 12 was infected in 1997 (one in 19 the year before). By July 1997 most big companies had been infected with CAP: it was not exceptional to find more than 8,000 infected files in one company. More surprisingly, in August 1998, a PE file infector (WIN32/HLLP.DeTroie.A) spread like wildfire.

264 different viruses have circulated in France over the last seven years. (The figure I gave previously in *VB* of 450 was rather too pessimistic). Some viruses have come and gone within a year, others have been with us since 1992. By now this figure is stable at about 100.

The levelling out of ItW viruses since 1995 may come as a surprise in view of the common perception that the number of alerts is ever increasing. However, I believe this can be easily explained. Viruses are now so common that many are no longer reported. Readers will have noticed that the number of viruses on the Main List of Joe Wells' WildList has also remained constant since November 1997.

There is another possible explanation for this; macro viruses are more and more prevalent and some anti-virus products do not mark the difference between the numerous variants. In the same way, users often indicate only the generic name (e.g. CAP, MDMA, Npad or Wazzu) without the variant designation. Only the French variants are occasionally distinguished. 15 variants had been noted in France by mid-November 1998 at a time when 832 variants were known for the four virus families named above. As stated in the August 1997 issue, the number of viruses that had been written in France up to then was nearly 70; now, it is nearer to 120.

WM/Appder.A was the first French macro virus to become wild. It was initially named NTTHNTA (the virus writer's name), or FUNYOUR (subroutine name) when it was found in December of 1996. The name Appder derives from the name of one of the virus' macros. This virus was designed to be destructive after 20 documents had been opened, but a typo (intentional or not) rendered it practically harmless.

WM/Inexist.A was found in the wild in October 1997. It contains no trigger and no payload. This name is derived from message boxes (never displayed) indicating the possible presence of the virus. The next one to come along, WM/Wazzu.DO is of unique interest in that it contains its origin and birth date, and the author's name, in its code:

```
' VirusMacroWord du Bureau Informatique du
SIRPA
' Virus Anti Virus du 14 juillet 1997
' v0.1b – Sgt THERY – 18/07/97
```

WM/Wazzu.DV is a variant and WM/Wazzu.EC is one of the numerous corruptions generated by *Word* itself. These three Wazzu viruses were the most widespread viruses in France during this period.

In January 1998, a new and interesting kind of macro virus for *Excel* appeared in the wild in France – XF/Paix.A. Of most interest was that it did not contain VBA modules, but rather, it is implemented entirely as cell formulas (see *VB*, April 1998, p.16). Since no anti-virus products were looking within the formula boxes, the virus was not easily noticed. Its payload (invoked with a probability of 1%) allowed its discovery – an *Excel* window would appear with a title filling up the screen:

```
Enfin la Paix… (Peace at last…)
```

In 1997 and 1998, a virus writer with the nickname of ZeMacroKiller created a series of stupid and destructive *Word 97* viruses. One of them was in the news on the occasion of the last World Cup. W97M/ZMK.J (alias WorldCup98) displayed various dialog boxes. It contained the names of nine of the favoured football teams and the user had to choose the champion. If the choice did not match that of the virus (random selection), any of a number of destructive payloads could be triggered.

These viruses were rarely encountered in the wild and without the publicity provided by some anti-virus companies, they would have remained unknown. ZeMacroKiller wrote a *Word* macro virus generator (ZMK98MVCK) in September 1998 and now tries his hand at *Windows* viruses without success…

In France, 1998 marked the beginning of a new period with the resurgence of file infectors and the appearance of a new breed of even more sophisticated viruses. During the summer of 1998, WIN32/HLLP.DeTroie.A (alias Win32/Cheval) appeared. Let me suggest simply that one should consider this as a combination of BackOrifice and a

virus. With Remote Tools like BO, the victim must initiate the installation of the server component. To bypass this requirement, a Frenchman who does not appear to be linked with the underground created and released something called SOCKET23 (Sockets de Troie v2.5). In this version, the server portion was propagated via a virus capable of infecting *Windows 95* or *NT*. The viral component is named W32/Cheval, alias W32/HLLP.DeTroie.

I received the first sample of this virus on 19 July, 1998. Immediately, the author became known and warnings about the dangers of this malware were issued. Unfortunately, after the French summer holidays, this 'tool' was readily available and never had a virus so disturbed French industry. All the big French companies were infected between August and October of 1998.

Later, the author of the virus apologized on his own Web site. He gave specific virus removal tools and procedures. According to his confession, he did it for amusement and to learn about programming on the Internet. He did not foresee the consequences of his creation falling into hacker hands. He finished his text with this single sentence:

```
'Error is human and I am very sorry.'
```

One year later, the French authorities decided to take the matter seriously. They arrested this person in June of 1999 when his tools were found inside the 'Vitale' card management system (in the future the card is to be used in France to obtain medical treatment and prescriptions). In my opinion, this story is a perfect example of misfire. If the authorities had not waited for a formal complaint but, as I proposed, brought this person around to reason immediately, the consequences for him and for the French industry would have been less serious.

This article would be incomplete without talking about Spanska. Behind this pseudonym hides the best-known French virus writer. Before 1999 some of his viruses were on the WildList but are no longer widely encountered. They were disturbing, but not dangerous. Depending on the system time, they displayed texts and graphic effects.

Before W97M/Melissa, what I call 'The Return of the Worm Spirit' was born in France with the last Spanska creation. The first virus to be propagated on a large scale via email was W32/Ska (alias Happy99). It is still around and spreading via an attached file named HAPPY99.EXE, which is immediately sent to all your correspondents in a follow-up email after the one you meant to send. The (voluntary) execution of the program (HAPPY99.EXE) leads to an animated cartoon of a fireworks display… and the infection of the machine. Fortunately, the impact of W97M/Melissa has given this particular author food for thought. Before he left France, he said goodbye to the alt.comp.virus community:

```
Times are getting bad for virus/worm
writers…. So, i will take some holidays far
from this newsgroup and internet in general
for some weeks/months, "le temps que ça se
calme"…
```

But the war continues. When BackOrifice2K was announced amidst great publicity in America, a young Frenchman nicknamed Jaguar proposed his own tools called 'Armageddon'. Furthermore, and similar to W32/Ska, which is the most common virus in our country, the latest on the scene now is Win32/Pretty (PrettyPark) which appeared in France in May 1999.

This Worm was first available through a user Web page created via a French ISP. I received my first alert on 19 May but the spreading really began during the French Infosec conference (1 June, 1999) when an infected contribution was posted in the AFUNT (French *Windows NT* user Group) mailing list speaking about interoperability between *NT* and *Linux*.

Despite rapid reaction, this Worm is still regularly encountered around the world. Just as with W32/Cheval, it was not difficult to discover its author immediately but I do not understand why nothing was done by the authorities…

As previously noted, the French virus creators did not seem well organized. This situation does not seem to be changing. The title of my first paper, The French Connection, gave the impression of a criminal organization. I know that this idea did not please some virus writers who would say that the underground community is not organized in France. They are right! I agree.

To conclude this second paper, I would like to encourage other researchers to carry out similar studies on their local virus scene. The 'local' virus situation may be very different from the 'worldwide' one. This fact has many implications for Systems Experts and Researchers all around the world.

Users in France, or those who conduct business with company offices in France, need to look especially for the viruses listed or discussed in this paper, even if they are not as prevalent on the worldwide 'in-the-wild' list. As I said before, we have our own 'Top Ten' virus list in France. What surprised me when I prepared the 1998 list was that seven of the ten viruses were of native origin. Without similar studies, we cannot know if this fact is unique. But the preponderance of 'local' viruses (from the field) in a particular country must be seriously considered.

Also, France is now a more prominent participant in the worldwide 'virus' scene. Previously, France had been isolated and more immune. That is no longer true. Notice that more and more companies are placing researchers in 'local' places. More companies are going to need an active French researcher, to handle both French viruses and the 'extra business' due to the increased infection rate.

Acknowledgements to Simon Zambra, MA and to Jimmy Kuo (Director, AV Research, Network Associates, USA) who initially edited this paper.

A DAY IN THE LIFE

The Politics of Anti-Virus

David Ensign
ACS Government Solutions Group, USA

Unfortunately, protecting against computer viruses and other malware is a standard part of any computer support operation today. *Affiliated Computer Services Government Solutions Group* (*ACS GSG*) provides full service computer support to US Federal government agencies across the United States, each with differing missions, operating environments and requirements.

In 1991 most Federal agencies, like many organizations, hoped the computer virus problem would remain isolated. *ACS GSG* received reports of fewer than ten virus encounters in the four years after the first virus appeared in the wild in 1987. Nevertheless, in late 1991, we determined that the threat from viruses was growing and began formulating a protection plan for our customers.

The timing was fortuitous, because the Michelangelo scare arose in February 1992. While exaggerated, it created an atmosphere in which organizations were receptive to anti-virus protection implementation. With draft procedures already available, widespread distribution was accomplished in a matter of weeks. No instances of Michelangelo were found but numerous other infections were, justifying the heightened focus on viruses. As a result, *ACS GSG* devoted a small but dedicated staff to the issue.

Our virus protection philosophy is based on the following tenets. Regardless of how successful the program is, viruses will continue to try to infiltrate from outside sources. Therefore, all plans must be long-term. The key to virus protection is early detection and removal. The only effective early warning system is the use of anti-virus software with active monitoring (on-access scanning). User restrictions (no Internet access, mandatory diskette prescanning) should be avoided, as these usually face resistance, resulting in non-compliance and a negative attitude. Protections should be automatic and invisible to the user (unless a virus is detected).

In the case of an infection, the most important element is an investigation by trained staff – the cornerstone of a successful and proactive virus protection process. This assures that all infections are properly cleared of the virus, there is no continuing threat to internal workstations and media, and the source and all recipients are alerted to potential problems, preventing accidents and hopefully eliminating the source for additional virus incursions.

In order to trigger a support request and ensure an investigation, users should be discouraged from clearing their own systems. Where possible, AV software should be set up to disallow automatic eradications of viruses that reach the desktop. (Unfortunately, few major packages provide options to prevent user-initiated or prompted eradication.)

Data collection associated with virus incidents is essential for tracking containment and gathering critical information to analyse for trends and trouble spots. Where available, automated encounter reporting should be activated.

Recently, we implemented email gateway firewalls with virus scanning. We now have a centralized detector at a single point of entry for email and an automated mechanism for tracking a large majority of incidents, providing a more accurate picture of the virus situation. Due to automated reporting, the firewall can clean and forward the attachment to the user, eliminating any impact on productivity and alleviating the need for an on-site investigation.

With macro viruses the greatest threat today, the email gateway has become the most important weapon in our arsenal (although desktop protection remains the fundamental component of a complete solution), providing a line of defence for everyone, even in organizations that do not have proper or up-to-date anti-virus software at the desktop. In the face of Melissa, it provides a tool for the rapid identification of an outbreak and a viable platform for countermeasures. If you can afford it, implement one; if you cannot, find a way.

Daily Activity

Due to the various reporting mechanisms, we log hundreds of virus encounters every month. Several times a year, we provide one-day orientation courses on viruses and virus response procedures to organizational computer support staff, who take that expertise back to their groups. These trained staff provide the first line of support to users, with *ACS GSG* available to help and to review each incident in order to ensure proper containment (quality control).

Optimally, users who encounter a virus notify their support staff or a central Help Desk. A trained computer support staff member responds to the user and determines the source and propagation of the virus and eradicates it appropriately. Investigation sheets (outlining all applicable questions for each type of virus) are completed and faxed to us for review and logging.

An organization must escalate notification of an incident to *ACS GSG* immediately in four cases: when the virus is new to the organization, the virus is destructive, the virus has infected a shared resource such as a network server, or the virus is network-aware. Our involvement is directed in order to ensure complete containment and prevent loss of data or services, and to analyse any new threat that may require a procedural change.

We also monitor the email gateway scanners, which log every intercepted message. Usually, the infected attachment is cleared and transmitted to the user. The original message lists the source and all recipients, so a complete history of the virus is detailed, simplifying the 'investigation'. The gateway notifies the sender automatically, alerting them to problems and so eliminating one of the computer support staff's duties. If the message is outbound (originating from within a site), the appropriate support staff are notified.

Recipient lists are reviewed to see if associated organizational users have received the file, unaware of, or possibly unprotected from, the danger. The gateways have been a boon, eliminating the need to dispatch staff to the many desktop computers that used to be infected each month, drawing precious resources away from other tasks. This alone has recouped any costs for the gateways themselves.

Gateways also have policy checkers that allow them to intercept hoax messages based on the message's text. We review all of these immediately upon receipt, as they are sometimes legitimate messages that need to be passed on. Hoax messages are not transmitted, greatly reducing the number of calls from users, which used to occur daily, again distracting resources from more important efforts.

Virus reports are monitored constantly for immediate identification of any threats. Summary reports are printed monthly and examined for dangerous trends. A written analysis is provided for corporate computer security and management staff each quarter summarizing the current situation and recommending preventative improvements.

The Internet provides an easily accessible informational avenue to staff and users. We use it to distribute AV software and updates. Most anti-virus software business licences allow for home use and employees can install and maintain the software at home much more efficiently through Internet access. We maintain Web sites at customer locations for education, hoax information, and analyses.

A Recipe For Virus Protection

1) Complete protection means an anti-virus program on each and every desktop computer. With today's propagation volume, any chink in the armour is susceptible to virus invasion. The desktop is the single focal point for the major virus types and must be the place to centralize protection.

2) Active monitoring (on-access scanning) is the only viable desktop implementation. Periodic scanning, even daily, provides a large window of opportunity for viruses to spread in an interconnected environment. Melissa, arguably the most dangerous virus to date, attacks upon infection, so periodic scanning will never provide protection.

3) Procedures and mechanisms to roll out the latest AV updates must be in place. Melissa spread worldwide in three days. You must be able to update signatures as soon as they are released. Monthly (or even weekly) updates are no longer sufficient. For timeliness and resource efficiency, network distribution is the only viable avenue in any large company. Do not rely on users to pull the updates; push them out whenever possible.

4) With macro viruses now to the fore, email is the primary propagation vehicle. While beneficial to virus writers, it also provides centralized points of control that can be used for protection. An email gateway can provide significant protections that will soon recoup any costs.

5) With viruses spreading through so many avenues, a multi-tiered protection architecture is beneficial, and protections should be placed wherever possible (e.g. servers, post offices, gateways, firewalls). We have found that no single package is perfect all the time – implement different programs at different points, both horizontally and vertically, especially in a large enterprise. This increases the possibility that a virus slipping through a gateway, for instance, will be caught at the post office or desktop.

6) Install or activate any reporting mechanism you can, especially automated ones. With viruses, there is no substitute for the awareness that statistics can provide. Many virus problems occur only because they are not seen, and many disasters could easily be avoided with a minimal amount of information.

7) Make sure you have two-way lines of communication with your user community. With the recent publicity about viruses, users are more attuned than ever to the situation, and they can be excellent sources on new threats. Encouraging users to send all alarmist messages to a clearing house will forestall any widespread panic and allow you to update your email policy checker to stop them. Users must be partners in any anti-virus campaign. Use every means to educate them and keep them informed, and be sure to have an emergency broadcast mechanism for use in a crisis.

8) Think about the big picture. Worrying only about your small community will not help in the long run; you will continue to be bombarded by viruses from outside, often from the same source over and again. Notifying originators can alert them to problems they can correct, reducing them as potential sources in the future. Every office and user that implements proper protections decreases the number of avenues through which viruses can propagate.

All this has a cost, but the price of not doing it could be astronomical. Incidents occur daily that justify a high-control approach. Our recipe minimizes virus encounters, and we have the expertise to respond to situations with skill and speed. The virus problem is growing, and new techniques in propagation and payload create a world where dedicated staff are no longer a luxury – they are a necessity.

OPINION

Happy New Year…

Nick FitzGerald
Computer Virus Consulting Ltd, New Zealand

I missed the presentation of Graham Cluley's *Millennium Madness* paper at VB'99. However, those of us attending the technical stream paper presented concurrently to Graham's heard his audience enjoying it in the adjoining room. From a subsequent viewing of the slideshow, it is apparent that Graham covered the important issues of what the real Y2K 'virus concerns' are, and that these bear little relationship to the media event some are encouraging.

Essentially, the message should be 'business as usual'. That's all well and good, but with the dawning of a new millennium in just over twelve months, I'd like to suggest that now may be the time to ask whether we have not put the cart before the horse? Maybe now is a good time to reconsider our whole approach to anti-virus issues?

Scan You See What I See?

What have been the most worrying virus developments over the last few months? Perhaps as far back as a year or so? Mass-emailing viruses such as Melissa and/or the email 'worminess' introduced by Ska seem to be the answers, at least from the informal virus-related chat at the conferences I've attended. The speed of distribution of both is certainly a concern, but such distribution mechanisms do not really cause any virus *detection* problems.

I am more concerned at the increasing number of non-structured files that can now carry malicious code, and therefore may require scanning. We are approaching the day where most of the content of most new files arriving on a machine will have to be scanned. Without the advantage of pointers in the header of an HTML file to tell a scanner where to find the code in the file, locating the JavaScript, VBS or whatever's next, will become a slow, old, grunt scan. In the last year we've seen VBS script files, HTML files (with various related extensions, such as HTT and HTA), *Corel* Script (CSC) and *mIRC* and *Pirch* script files (INI), amongst others, added to the 'watch this space' list.

It seems unlikely that *Microsoft* will abandon its current path to world domination – sell 23 copies of *Visual Basic* per machine, but give each a name that makes it sound both completely different from all the others and completely indispensable. More depressing for those with strong security leanings; it seems equally unlikely that many of *Microsoft*'s customers will revolt against this.

So, anti-virus developers, faced with users who choose unsafe applications, will be scanning much more of your hard drive for many more viruses of many more types.

Use IT, Don't Abuse IT

Corporate IT managers should aim to take their users out of the problem. Neither Melissa nor Ska could have become corporate problems if they were *never* run (as opposed to 'never run once our anti-virus software was updated'). That they *were* run following a conscious user choice suggests we should address that choice point. Perhaps the best method of controlling the impact of such user decisions is an integrity management system that only allows 'sanctioned' code to be executed. This is not the same as code-signing, although the two could be used in combination, allowing more flexibility but reducing security.

The basic idea has been around for at least as long as the term 'computer virus'. Dr Fred Cohen advocated the approach and even developed a DOS product implementing his ideas. In the resource-strapped, single-tasking, single-user world of DOS, with no memory protection or file system security and little networking, his product was not successful. Now, with 128 MB of RAM and 400+ MHz CPUs common, and mobile LAN connectivity, these problems should be easily overcome.

One of the best-known scanners reports, as I write, that it detects '48081 viruses, trojans and variants'. That strikes me as almost absurd – should we be scanning the hundreds to a few thousand potential virus hosts on each PC on the network for 50,000 viruses – few of which have ever been seen outside anti-virus test-sets – or is it more sensible to validate that code that is about to run is one of the couple of thousand programs allowed on our network?

The integrity management approach may take a little more initial effort than a traditional virus scanning approach, but that should not be an impediment to its use. Conscientious system administrators should be able to produce a listing of the software in use in their organization anyway, *and* be able to say who is licensed to use what. Once each 'program' (remember scripts, macros and the like) has been 'fingerprinted' and its user profile defined, the pay-offs are potentially tremendous. Mobile or home-based users, and those at remote offices which may not have a continuous connection to the main LAN, are always 'up to date'. When they install a sanctioned software addition or update, their access profiles will also be updated, so they remain current. You are no longer dependent on regular software updates and the nuisance value of obtaining them and ensuring they are rolled out where they are needed, and so on.

If IT is an integral part of your business, should maintaining its total integrity not be given at least the same priority as maintaining the integrity of the company accounts? Neither the CEO nor the janitor should be able to threaten the integrity of your IT infrastructure because they want to install their favourite screen saver. The choice is yours…

PRODUCT REVIEW

Grisoft AVG v6.0

Martyn Perry

This month's review product is in the process of having a marketing makeover in terms of product functionality supplied in different flavours. Traditionally, *AVG* has offered the users two 'products' in one package – Standard and Advanced user interfaces. This is all to change shortly, with the Standard product being made available free of charge, and the Advanced product being rebadged as *AVG Professional Edition*. This review is concerned predominantly with the Standard product version, although certain configuration options available through the Advanced interface are also described, since these will most likely be present in the forthcoming professional product.

The licence allows for just one copy of the software to be installed on one PC at a time. A separate network licence is required for each PC that can *access* the network (not 'assess' the network as documented in the licence – pretty sloppy for a legal document). However, the licence does allow the primary user to have the software installed on more than one PC, provided that the software only runs one machine at a time. This allows for laptop and home use.

Presentation and Installation

AVG 6.0 is supplied on CD which autoloads and calls up a Web browser to view the opening HTML screen. This gives five language options; UK English, US English, German, Czech and Slovakian. Selecting the appropriate flag brings up the next screen, which provides options for installation, update and information.

Product installation offers a choice of the commercial version, Soho version or the previous version of *AVG*. Choosing the commercial version leads to File download and an option to run SETUP from the CD or copy down onto the PC. The Default is save to PC, which copies SETUP.EXE down to a selected directory (AVG) and returns to the browser.

Curiously, when SETUP was attempted from the disk drive rather than from the CD, an error occurred saying 'Cannot init language file'. However, if the setup is run again directly from CD, the warning does not appear. Initially, there is a security warning from the browser because it is set up to be recognized by the Trust Provider. This is because the software is classed as an unsigned program.

Assuming that the source is trusted, clicking OK shows the Welcome screen with the usual recommendation for installations to exit *Windows* before running the Setup program. From this point, the installation follows a predictable sequence, displaying the Software Licence Agreement for acceptance and personalizing options of User Name and Company. Using the serial number determines the version of the product installed. The default installation location can be set to C:\Program Files\Grisoft\AVG6. The last option selects the program folder (default AVG 6.0 Anti-Virus System).

A final confirmation screen is for location and user details. Once the files have been copied, another screen selects options to configure the level of protection. These include: *AVG* Resident Shield, which works in the background to protect all files and *AVG* E-mail scanner which works with *MS Exchange Client*, *Outlook* and *Eudora*. *AVG* Control Center is the main management facility to allow scheduled scans and modify configuration, while *AVG* Boot-Up scanner tests key parts of the operating system at startup. The default has all options selected.

Selecting the user interface window offers a choice between Standard (predefined, ready to run scanner) and Advanced, the default which configures the various aspects of *AVG*. The Automatic Update window allows either automatic download from www.grisoft.cz or www.grisoft.com, or the option to handle the update manually. The Readme file is displayed, followed by the final screen which can restart the system immediately following the end of installation.

After restart, a boot scan test is performed and the configuration files adjusted. This is followed by the *AVG* First Run Wizard which allows the user to update to the latest version, create a Rescue Disk, and scan the PC. After the Wizard has completed, there is the option to exit the program or continue running the software.

Using AVG

Upon running the shortcut that is placed on the desktop, the user is presented with the main program console. Here, two buttons to start the two pre-defined tests are provided.

Information concerning *AVG*, its licence information, the *AVG* virus database and system configuration is accessible from the 'Info' button on the console. Similar buttons are provided to access the help files, view test results, or access the scheduler.

The Standard version comes with two pre-configured tests; a Removable Media Test to scan floppies, CD-ROMS etc, and a Complete Test to check all hard drives. These tests can be started from buttons on the main program console, or from the Tests drop down menu. The menu also enables the selection of a Custom Test, which allows the selection of drives and directories at scan time.

Though pre-defined, the settings of the Complete Test can be adjusted from the drop down menus. The drives and directories and file extensions to be scanned can be defined. Scans can include All Files, Program files and/or Documents and Sheets. If desired, additional user defined file extensions can also be tested.

There is another option called 'Smart Scan' – *AVG's* answer to file type recognition. This is enabled by default, and as such was enabled throughout tests. There is an extra selection to determine whether or not archives and compressed files are included in the file scan.

Configuration via the Control Center

Accessing the *AVG* Control Center from the drop down menus enables configuration of the Resident Shield, E-mail scanner, Update Manager and Scheduler. The on-access scanner apparently monitors all computer operations and begins by checking all files and removable media. Options to check boot viruses, executable viruses and macro viruses are available. There is a further option whether or not to use Heuristics. The last option disables the Resident Shield.

By default the scheduler is configured to run the Complete Test every 24 hours. From the Control Center it is possible to disable this scan. When using the Advanced interface (and so probably of relevance to the forthcoming professional product version since the feature is likely to stay) the scheduler is far more powerful. The creation of customized tests is permitted, which can be added to a Plan List. Each plan contains details as to when the scheduled scan will run. Parameters to determine whether the scan will run with low, normal or high priority in the background or foreground, and with or without user input can be added to each plan.

Updates to the program and virus database files can be obtained either from the Internet, from a folder (local or network), or from a CD. The Update Manager, accessed from the Control Center, allows the user to configure whether or not to enable scheduled updating from the Internet. The frequency of automatic updates can be adjusted from daily through to every 99 days. Users can choose to download the updates from either the American or Czech sites (http://www.grisoft.com or www.grisoft.cz). There is an override which allows for an immediate update to be performed.

The Virus Vault

When *AVG* detects a virus that cannot be removed by AVG's 'automatic healing' (disinfection), it deletes the infected file via a quarantine – the *AVG* Virus Vault. This is simply a directory that stores the infected files.

The file names are changed and the content is encrypted so that the files cannot accidentally be executed. The vault also provides the ability to restore these files if necessary. If the product is de-installed while there are files still in the vault, the user receives a prompt to deal with any remaining files before closing down.

Product Help

The 'Help' section comprises three groups of information. Program Info repeats the serial number, version information and licence details. Virus Info displays a list of virus descriptions that *Grisoft* considers to be widespread or interesting. For those keen to verify the correct operation of their product, this list also includes the *EICAR* test string. For additional help, there are contact addresses for *Grisoft Software*, and computer configuration details which can help with problems.

There is a Bug report facility which is a notepad to allow logging of problems. The utility tries to send email of the report. If it cannot, it creates a file C:\AVGBUG.TXT which contains the text of the problem plus automatically appended user details and version information for the main program as well as key supporting DLLs. Finally, the 'Help' text itself was limited on this version although there was good background information about the anatomy of a PC and virus types.

Web Presence and Support

The company Web site contains a good balance between technical and commercial information. On the commercial side there are news, press releases and a company profile. In addition, there is the facility to download a 30-day evaluation product as well as an option to purchase a full licence on-line. Technical facilities include virus alerts and

signature updates that are expected on all AV company sites. The Web page also provides links to virus-related sites such as newsgroups and even a link to *Virus Bulletin*.

Detection Rates

The scanner was checked against the standard *Virus Bulletin* test-sets – In the Wild, Standard, Polymorphic, and Macro sets. Importantly, the ItW (file and boot) set was aligned to the August 1999 WildList. The tests were conducted using the default scanner file extensions supplied. Detection rates were determined from the log files.

As would be expected from any decent AV product, *AVG* kicked off proceedings by detecting 100% of the ItW boot sector viruses. Pleasingly, the same result was achieved against the ItW file set. Seven samples were missed in the Standard test-set – one VBS/First.C, three Goldbug samples and three Win32/Kriz samples. In the Macro set it missed all the *Access* files infected with the A and B variants of A97M/AccessiV, and a variety of *Word* and *Excel* samples. Interestingly, a few infected *Word* document templates were missed despite the correspondingly infected documents being detected.

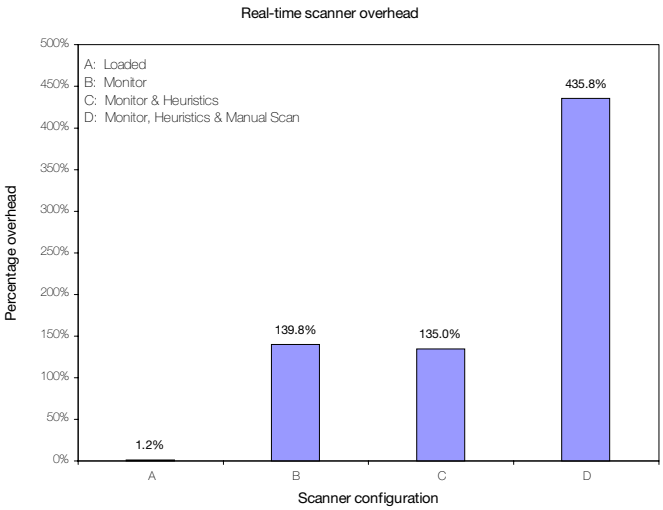
Against the Polymorphic test-set, the scanner missed six samples of ACG.A and all 90 samples of ACG.B. When the tests were re-run with all file extensions being checked, the results were identical.

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. The folders used for the source and target were excluded from the virus scan so as to avoid the risk of a file being scanned while waiting to be copied.

Due to the different processes which occur within the machine, the time tests were run ten times for each setting and an average taken.

- Program not loaded: establishes the baseline time for copying the files on the PC.
- Program installed, shield off and heuristics off: this tests the impact of the application in its quiescent state.
- Program installed, shield on, heuristics off: this shows the impact of having the program running with shield enabled but without heuristics.
- Program installed, shield on, heuristics on: this shows the impact of having the shield running with the heuristics tests.
- Program installed, shield on, heuristics on and running scanner: this tests the impact of the application scanning files when running a separate scan on the PC.



As can be seen, the real-time scanner enforces an overhead of almost 140% when enabled. This compares well to other products, and is consistent with the observations *VB* has made in recent Comparative Reviews (see for example *VB*, November 1999, p.23). When a manual scan is performed as well, the overhead climbs sharply to well over 400% as would be expected.

Summary

Operation of *AVG 6.0* through the Standard interface is an extremely straightforward affair. The lack of configuration options is only as might be expected for a product that is available for free download. To access the full functionality of the product, it is necessary to upgrade to *AVG Professional Edition* (soon to be released).

One small quibble is the number of files which are written to the root of the C: drive when the integrity checker is run. I would prefer to see these default to the *AVG* directory. Although the databases can be redirected, some of the text files remain fixed. Another gripe is that these databases were not removed during an uninstallation of the product.

In summary, *AVG 6.0* presents the user with a convenient and simple scanner, exhibiting good scanning speed and competitive detection rates. Available free of charge, the standard product certainly looks to be an effective product for both home and business use.

Technical Details

Product: *AVG 6.0* (build 87, 01/11/99).

Developer: Grisoft Software, Lidicka 81, 602 00 Brno, Czech Republic. Tel +420 5 41243867, Fax +420 5 41211432, email info@grisoft.cz, WWW http://www.grisoft.com/ or http://www.grisoft.cz/.

Price: Standard – free, *AVG Professional Edition* – $39.95.

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/Win98/199911/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:

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

Tel (781) 2139066, Fax (781) 2139067

END NOTES AND NEWS

***Microsoft's* seasonal dance card is filled by the unlikely double-booking of rivals *Symantec* and *Network Associates Inc*.** Both anti-virus giants have announced independent, 'strategic' partnerships with *Microsoft* in a bid to help corporate clients combat Y2K-related viruses and malware attacks. Http://www.microsoft.com/y2k houses free, fully functional trial software for download from both *Symantec* and *Network Associates*.

A two-day course entitled Practical Anti-Virus will be run by *Sophos* on 25 and 26 January 2000 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.

***IIR Training* is hosting a practical two-day foundation course called 'How Do Networks Work?' on 13 and 14 December 1999 in central London.** An optional workshop will be running on 15 December. For more information about location and prices contact; Tel +44 171 9155055, or email information@iirtraining.co.uk.

Network Associates Inc* announce the release of *Gauntlet v5.5, allegedly the first Firewall with integrated virus scanning, VPN and content screening management from a central console. The product is available now for *NT* and Unix and starts at $6 per seat for 1,000 users. For more information contact; Tel +1 408 9883832 or visit the Web site http://www.nai.com/.

Content Technologies Ltd* announce the release of *e-Sweeper*.** Powered by the *MIMEsweeper* engine, the product enables Service Providers to check all incoming and outgoing email and attachments for content threats including viruses and virus hoaxes which can then be quarantined. For more details on this product and its three levels of deployment contact Catherine Jamieson; Tel +44 118 9301300 or see http://www.mimesweeper.com/. ***Content Technologies Ltd has also established **The Threat Lab** at its UK headquarters. All aspects of content security threat shall be analysed, monitored and examined here. The Threat Lab will also act as a centre for information on unfavourable email content such as hoaxes, active content, macros and viruses. The company intends to publish its findings regularly at http://www.mimesweeper.com/threatlab/.

The fourteenth annual *Vanguard Enterprise Security Expo 2000* will be held at the Atlanta Hilton and Towers, Atlanta, Georgia, USA on 15 and 16 May 2000. Details can be found at the Web site http://www.vipexpo.com or contact *Vanguard*; Tel +1 714 9 390377.

Walt Disney Co, one of the world's largest corporations, avoided a potentially disastrous PR blunder when an internal memo infected with Melissa was inadvertently sent out to a list of press members in the corporate address book. Fortunately, the document did not contain sensitive information. *ZDNet*, who broke the story, points out how significant this incident really is – the future is a virus that not only has the potential to damage data but can publicize it as well.

Symantec's* new *Norton AntiVirus Corporate Edition v7.0* together with *Symantec System Centre focuses on the easy management of anti-virus policies inside the enterprise. The product combines technology from *Symantec*, *IBM* and *Intel*. For more information contact Lucy Bunker; Tel +44 1628 592222 or visit the company Web site http://www.symantec.com/.

The second *AVAR* (*Association of Anti-Virus Asia Researchers*) Conference was held in Seoul, Korea on the 28 and 29 October 1999. Subjects included *AVAR's* role in the prevention of the spread and potential damage of computer viruses by exchanging information around the Asian Pacific region. Also discussed were the topics of technical and legal standards necessary to facilitate this, and the Governments willing to help and co-operate with *AVAR*, namely those of Korea and Japan. There was a presentation on the generation gap in computer knowledge, which could lead to children getting into inappropriate computer activities, including virus writing. Another paper covered the spread of Worms and discussed the changes in policies necessary to handle a new Worm every week or every minute. The *AVAR* 2000 conference will be held in Japan. More information on *AVAR* can be found at http://www.aavar.org/.

Virus Bulletin
wishes all our subscribers a very
Merry Christmas & Happy New Year