ISSN 0956-9979 DECEMBER 2000

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Creating monsters:** forget polymorphism, Péter Ször reckons that metamorphism is the new standard to which virus writers aspire. His technical feature starts on p.8.

• **Coming a cropper:** Andreas Marx of Magdeburg University starts his Feature Series on p.12 with a look at some simple, common problems thrown up during AV product testing.

• **User friendly:** *EICAR's* Chairman of the Board outlines a new and ambitious project to get to the bottom of what users really, really want. See p.16.

CONTENTS

COMMENT

Another Fine MSN!

The virus problem is over. *Microsoft Network* (*MSN*) has basically declared viruses to be no longer a problem, closed down their AV area, and said to move on. And *Microsoft* knows what it's saying and it is never wrong, right?

Ignore the breach in *Microsoft's* security giving Bad Guy access to future product source code in October 2000 by QAZ, or even the near harmless but annoying and buggy Navidad worm hitting in November this year. QAZ is not a problem, *provided* you keep your AV signature databases up to date, and actually use the AV products you've installed to protect your assets. *Microsoft* obviously did not do so, hence the infection and subsequent breach of security giving access to the crown jewels. Whoops!

"Microsoft has its head in the sand…"

Now, I have a vested interest in what you're about to read, but I think you do, too. Until November of 2000, I ran the Safe Computing forum for *MSNs ComputingCentral*, an unbiased anti-virus support area for all anti-viral products, with up-to-date threat assessment offering support, support message boards, and free assistance to members.

Then, declaring this area no longer of interest to its members, *MSN's* management closed the forum down. An area's success or failure on the Web is ascertained by how many unique user hits are generated in a given time frame. During a viral outbreak, such as Melissa or the LoveBug, monthly hits for the forum by unique users rose to over 200,000, making it one of *ComputingCentral's* more popular forums (forums for downloading virus-checked shareware and those ubiquitous screen-savers are always popular, of course).

However, management at *MSN* declared virus control no longer an issue for concern and summarily closed down the forum. My vested interest? I got paid to run the area, to answer questions, and to keep things up to date and accurate. It was my job to run a forum reflecting on a fast changing arena. Your vested interest? Think of all the time, effort, and money you've been wasting over the years on something *Microsoft* has now declared a non-problem. You feel foolish that you've done so for years, right?

I am the author of the world's first true anti-virus product (*Flu_Shot*), a longtime *Computer Anti-virus Research Organization* (*CARO*) member, and the one guy in the world running on-line anti-virus support forums on multiple services for over a decade. Heck, I was even a member of *Virus Bulletin's* initial Editorial Board. I feel I've got, and have earned, the credibility in the AV field *MSN* management so obviously lacks. *MSN* management has foolishly fallen for more than a few of the virus hoaxes over the years. Moreover, it gathers its own feel of the significance of a viral incident by what is published on sites themselves reported frequently on Rob Rosenberger's Virus Myths page (http://www.virusmyths.com) as the butt of many a joke.

So, consider this a warning: *Microsoft* cannot simply declare computer viruses a non-issue and hope people believe it. That's what the company has done, though. *Microsoft's* own record in the field shows that it doesn't take virus warnings seriously, that it doesn't follow the advice even of the world's most expert anti-virus researchers (generally other *CARO* members) and that *Microsoft* employees spend more time and effort pro-actively dealing with their image and public relations issues than fixing well-known security holes in their products.

Now, rather than dealing with these issues publicly and in a public forum, *Microsoft* simply closes that forum down and hopes nobody notices. My task is to make sure you do. *Microsoft* has its head in the sand on your behalf, and doesn't consider what that leaves exposed.

For shame, *Microsoft*!

Ross Greenberg, Software Concepts, USA

NEWS

AVP is KAV

Russian-based anti-virus company *Kaspersky Lab* is changing not only the name of its flagship anti-virus product but also its logo from mid-November 2000. Official reasons are being given as diversification of the product range and the establishment of a 'clear relationship between product and company name.' Is it pure coincidence that *Virus Bulletin* has often remarked on increasingly bitter problems with *KL's* US resellers *Central Command* and the rights to the *AVP* name? Users should be aware that until February 2001, *AntiViral Toolkit Pro* (*AVP*) from *Kaspersky Lab* will co-exist with the newly named *Kaspersky Anti-Virus*. Up to that time, the old name will be phased out. The company trademark will be simply *Kaspersky* ▮

AVX-tra! AVX-tra!

It never rains but it pours. Further to the confusion surrounding *Central Command's* release and boosting of *AVX* to its former *AVP* users, it appears that *Central Command* may have more problems than it bargained for. Aside from early user reports of system instability following installation of *AVX*, some of the installer applications *Central Command* has been distributing appear to have been infected with CIH.1024 at some point, then disinfected.

Both the *AVX for ICQ* installer, AVX4ICQ.EXE, and the installer for the trial version of the full desktop product, SETUPAVXPRO.EXE, downloaded from www.avx.com on 22 November contain the 'body' of CIH.1024. In attempting to discover whether this 'contamination' occurred at developer *SoftWin's* end of the process or after it shipped the EXE files to distributor *Central Command*, the equivalent files were also downloaded from www.avx.ro.

As the desktop trial version of *AVX* available from the Romanian site is simply packed in a ZIP file, rather than a self-extracting and installing archive, that file was not of much use. However, although the AVX4ICQ.EXE from the Romanian site was a slightly different version, it also showed signs of previous infection from the same virus.

It would be devastating for both *Central Command* and *SoftWin* had these files been carrying active infections. However, the fact that the disinfection process seems not to have removed any of the virus' code and just fixed the PE entry point in the header is what allowed the discovery that the files had been infected at some point. It must be sufficiently worrying for a potential *AVX* user to know that the developer or distributor allowed an active virus near enough to any of its shipping code that could become infected, let alone that this actually happened! ▮

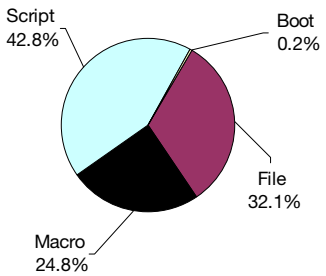
Prevalence Table – October 2000

Virus	Type	Incidents	Reports
LoveLetter	Script	553	22.2%
Win32/MTX	File	513	20.6%
Kak	Script	328	13.2%
Stages	Script	120	4.8%
Divi	Macro	113	4.5%
Win32/QAZ	File	99	4.0%
Marker	Macro	78	3.1%
Win32/Ska	File	70	2.8%
Laroux	Macro	61	2.4%
Barisadas	Macro	50	2.0%
Ethan	Macro	46	1.8%
Win32/Pretty	File	43	1.7%
Tristate	Macro	39	1.6%
Thus	Macro	38	1.5%
Freelinks	Script	37	1.5%
Win32/Funlove	File	37	1.5%
Class	Macro	29	1.2%
Netlog	Script	27	1.1%
Melissa	Macro	26	1.0%
Sat	Macro	17	0.7%
Story	Macro	15	0.6%
Win32/Kriz	File	15	0.6%
Myna	Macro	14	0.6%
Cap	Macro	12	0.5%
Others[1]		112	4.5%
Total		2492	100%

[1] The Prevalence Table includes a total of 112 reports across 41 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

In order to avoid a distortion of the figures, data for the 'self-reporting' W97M/ColdApe virus (totalling 482 reports in October) have been omitted from the table this month.

Distribution of virus types in repor



LETTERS

Goodbye Y2K, Hello 2001

[*Major AV vendors from around the world ring out the old and bring in the new. Ed.*]

Alwil Software, Czech Republic

One week after the infamous LoveLetter incident in May, the secretary in one company asked her IT Manager, 'When the LoveLetter danger is over, can I click on my attachments without any fear again?' This year demonstrated the danger coming with new technologies and new virus threats. While the biggest risk of the last year came from infected documents sent by email, the biggest hazards of today are the mass-mailing and fast spreading email worms and viruses. And we have seen that most PC users do not learn their lesson from the way things are today. The only way forward is to change the security design of the applications. Next year we will see the first intelligent mobile phones containing 'real' computers. Will the designers be aware of all the associated problems and take their chance? I wildly hope the answer is yes … Feliz navidad everyone.

Pavel Baudis
Vice President

Computer Associates Inc, Australia

There is one thing about the year 2000 which immediately springs to my mind – it's gone faster than any previous year in my whole life. It's probably just a sign of me getting old. On the other hand, what could possibly attract our attention after the long anticipated beginning of this year turned into a meaningless anti-climax? The world as we know it still crawls towards its doomsday, but at its usual pace. If it wasn't for the LoveLetter worm, this year would be no more exciting than the three zeros in its number. The only excitement we can feel at the end of these passing dozen months is the uncertainty about the next President Elect of the free world. As far as viruses are concerned, for the anti-virus industry its business as usual.

Jakub Kaminski
Virus Research Manager

Eset Ltd, Slovak Republic

In their effort to grab the 'palm of victory', the virus authors (being very vain) will do their best to conquer any available platform. Worms, macro viruses, script viruses, Win32 and *NT* viruses, and others will follow the pattern set by 'classic' file viruses, increasing the complexity of their detection. Further virus development will, without doubt, be linked to the wide exploitation of Internet possibilities. Using hacking techniques, exploits and backdoors, viruses will actively seek proper infection targets, enter the networks, modify and improve their code via different plug-ins found on the Internet (not on static Web pages but via USENET, IRC and so on). They can even 'make money' by misusing advertisement systems. Our imaginations are too limited to house the vision of the scope of emerging opportunities.

Miroslav Trnka
Technical Director

Gecad Srl, Romania

As I write these lines, I find out that it's easier for me to research some complex virus than it is to make a prognosis about next year. For sure, email will continue to be the most exploited method of malware spread. Features like metamorphism and self-upgradeable code will be used more and more in the next 12 months. Macro viruses will reduce their spread. Security holes will continue to help viruses to spread. However, the biggest problem I foresee is high-level language malware. We still need to find reliable ways to analyse this kind of program fully. One could say that we're already doing that, but what about the times when we'll receive ten times more of them? Got to go – Hybris is playing me some 'Music about Love'. Happy New Year!

Adrian Marinescu
Head of Research and Development

Grisoft Inc, Czech Republic

We live in unhappy times. Interesting questions like 'How many angels could fit on the tip of a needle?' are forgotten and replaced with questions like 'How many new worms we will see this week (day, hour …)?' or 'How fast can this new mass-mailing virus spread?'. The use (or misuse?) of the Internet is a typical attribute of today's malware, and I worry that the year 2000 only provided us with a 'demo version' of nice things we will meet in the near future. I'm still sure that the AV industry will be able to create solutions for all new malware, but I can only hope that we can deliver it to customers fast enough. We will see.

Petr Odehnal
Virus Researcher

Kaspersky Lab, Russia

While making predictions don't forget about the story of Cassandra – don't scare people to death. I'm sure there are no nervous subscribers reading *VB* so let's get to the next computer millennium, starting with its first year. What can we expect? I recall my forecast at the end of 1998, when the first HTML virus was discovered (the Rabbit virus). When I did an analysis of that virus, I warned about 'ten lines that will shake the world'. That happened a year and a half later. Looking at today's virus innovations, can we predict what we will face next year? Probably, yes.

First of all, viruses will utilize all possible features of the Internet. They will continue their integration into the 'Net, and probably bring lethal problems to some of its components. Secondly, non-*Windows* viruses will come. *Linux* is in the process of going global – well, count to ten and we'll see native *Linux* viruses commensurate with the number of *Linux* users. Thirdly, switch off your mobile phone and never connect your handy toys to the 'Net. Don't worry, just joking! But the Russians do like to say 'every joke is only partly a joke!'.

Eugene Kaspersky
Senior Virus Researcher

McAfee, USA

So another year passes and we are a little the worse for wear. We saw VBS/LoveLetter, and the love from its brothers, sisters, aunts, uncles, third cousins etc. And we were walked through the IRC/Stages of life, got a prank phone call from VBS/Timofonica@MM, and were blown to X97M/Oblivion by JS/Winbomb. It was also a miracle them finding us in one W97M/Piece mailed all over the place.

The W32/Southpark kids JS/Spawn-ed a whole new W97M/Generic set of sayings and doings. That was soon followed by a new W97M/Class of script kiddies who W97M/Marker-ed their territory, and said don't DUNpws.* from us. And so we say Goodbyee (W97M/Marker) and WM/Goodnight. Happy Holidays.

Vincent Gullotto
AVERT Labs Director

Panda Software, Spain

Despite widespread alarm generated by predictions related to the 'Y2K effect', the long-awaited date change hardly caused a ripple. A positive effect of the pandemonium is that it did bring about a greater awareness of the need to protect IT resources. Many companies, however, are yet to implement adequate complementary security policies. Given the dramatic increase in the virulence of malicious code, it is essential, now more than ever, to take these measures seriously.

In the past 12 months, email infections have risen sharply (up 87% this year, in the US, from 56% in 1999 according to *ICSA's* Computer Virus Prevalence Survey 2000) in comparison with other means of infection such as floppy disks or, as common myths would have us believe, Internet browsing. Other current forms of infection such as those posed by applications that support VBA, together with some security holes in ActiveX controls, lead me to foresee more self-propagating massive infections in the future.

Carlos Ardanza
Software Engineer

Sophos, UK

The anti-virus marketroid types will continue to delude us with their message of 'buy our software, install it and don't worry about amending your behaviour because we've been so clever on your behalf'. The products with the most tick-boxes will continue to win awards, and there will be lots of rumours about 'Immune Systems'. Some people will even try to deploy these 'new' technologies, but the most prevalent viruses will continue to be those which every anti-virus product on earth has been able to detect for ages, and against which even *Microsoft* published a fix last year. Go figure.

Paul Ducklin
Head of Global Support

Symantec, USA

The year 2000 was punctuated by the LoveLetter explosion, but held few other surprises. Let's call Y2K 'The Year of the Script/Win32 Worm'. On the technology front, this year delivered no major advances on either side of the fence. There were, however, two very significant developments in the industry. First, AV companies started building partnerships to embed anti-virus into the Internet infrastructure. Second, AV firms began tinkering with solutions for handheld devices. Both areas are very immature, but may some day impact our digital world profoundly.

How about 2001? On the virus/worm front, without any pervasive, connected and easily-programmed new platforms to target, next year will bring more of the same with perhaps a few new proof-of-concept viruses on the (handheld) device platforms. Win32 viruses will continue to evolve and cause major headaches for corporations. Finally, I've been wondering for quite a while when criminals, rather than kids, would start to build and exploit computer viruses. A trade traditionally dominated by pimply-faced adolescents may next year begin a shift to the mainstream criminal element. I hope not.

Carey Nachenberg
Chief SARC Researcher

Trend Micro, USA

So, we are finally here – 2001, the year we make contact with a mysterious, artificial artifact. At least, this was Stanley Kubrick's vision about 32 years ago. While all his visions may not have come true, we have certainly come a long way in the computing world. 2001 will certainly be an interesting year, as we will see the outcome of the *Microsoft* anti-trust suit, the war of the Web portals, an increase in mobile Web usage, and several new exploits in *Windows 2000*, *ME*, and *Whistler*. 2001 will also be a continuation in regards to macro, *Windows* 32-bit, and Script viruses/worms.

2001 will also be the year of improved anti-virus technology and better cooperation among anti-virus experts. In this regard, I wish everyone a happy holiday season and a 'guten Rutsch' (German for 'good slide') into 2001.

Joe Hartmann
Anti-virus Research Engineer

VIRUS ANALYSIS

Sonic Boom

Andy Nikishin
Kaspersky Lab, Russia

The era of network-aware viruses (more precisely known as network worms) began in the late 1980s. These included the Morris worm, Christmas Tree and WankWorm. They all made use of the erratic and undocumented functions of the global access networks of the time. These facilitated propagation, transferring copies of the worms from one network server to another, and starting their execution or at least 'pushing' a user to run the infected file.

The Internet is getting more and more powerful and penetrates almost all walks of life. At the same time, the Internet is a good basis for any kind of 'badware'. 'Internet worm', 'backdoor' – these words are relatively new to the Internet society but are already well-known.

Moreover, network viruses and worms prevail now in user reports – they are widely spread and because of that they are notorious. Everybody knows the names of LoveLetter and Melissa and fears them. Here we look at one of the latest network worms to have appeared in the wild. But before that, let us recall the near past.

One of the most well-known and widely spread Internet worms of recent times is Happy99 (Win95/Ska), which is still displaying a nice fireworks show and wishing users a 'Happy New Year!'. I wonder if there is a version which congratulates users on the new Millennium?

The history of backdoors is not as long as that of worms. Backdoors are network administration utilities that allow the remote control of computers on the network. One of the most infamous backdoors is BackOrifice (BO).

These days, it is possible to find this program on many hacker-related Web sites. One of the advertising banners on such a site says: 'Back Orifice is a remote administration system which allows a user to control a computer across a TCP/IP connection using a simple console or GUI application. On a local LAN or across the internet, BO gives its user more control of the remote Windows machine than the person at the keyboard of the remote machine has'. Realistically, BackOrifice Has become as powerful as any other commercially available remote administration tool. So what do you think would happen if you crossed Happy99 with BackOrifice? You would get the W32/Sonic worm.

W32/Sonic

This multi-component Internet worm was discovered in France and Germany at the end of October this year. W32/Sonic infects *Windows* machines (it works on *Windows 9x*) and spreads in email messages as an attached .EXE file. The Sonic worm is also able to 'upgrade' itself from an Internet Web site. There are two principal components to this worm: the 'loader' and the 'main' component.

Loader Component

The loader part is a *Windows* EXE file about 25 KB long (it is compressed by the UPX PE EXE file compression utility; when decompressed it is about 70 KB in size), written in Visual C++. When the loader is activated on a computer (i.e. launched from an email attachment) it first checks the type of operating system, registers itself as a hidden process (service) and displays a concealment message box to hide its activity (see picture below).

Next, it checks an infection mark and if the system is not already infected it copies itself to the *Windows* system directory under the name GDI32.EXE, and runs this file. Sonic uses the date of the WIN.INI file as an infection mark – setting the creation date and time to 31 December 1999, 23:59:59. (On a FAT file system it is impossible to set this time because of the file system's limitations and the file time will be set to 23:59:58. Nevertheless, the worm works properly on FAT drives.)

When Sonic is started from the GDI32.EXE file, it registers itself in the auto-run system Registry key ('%SystemDir%' is the *Windows* system directory name):

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Run
 GDI = %SystemDir%\GDI32.EXE
```

As a result, the worm loader will be executed on each *Windows* startup. Sonic sets dates and times for its files identical to those of the SCANDSKW.EXE file in the *Windows* directory. Usually, all *Windows* system files have the same creation/modification dates. This worm uses just such a technique to disguise itself in a standard *Windows* environment. It is important to note that there are standard *Windows* components in that directory, namely GDI.EXE and GDI32.DLL.

The Sonic worm then activates the main procedure that gets and executes its main component. To facilitate this, it enters the http://www.geocities.com/olivier1548/ Web page and down-loads several files from there. It deposits all the following files in the *Windows* temporary directory:

- LASTVERSION.TXT – contains a number of the latest versions of the worm. If there is no new version, the worm exits.
- nn.ZIP – contains the latest version of worm's main component, 'nn' is obviously the number defined in LASTVERSION.TXT.
- GATEWAY.ZIP – contains the latest version of the worm's loader component (only a few versions of the worm use this file).

The nn.ZIP and GATEWAY.ZIP files are not in fact archives, but encrypted *Windows* EXE files. The worm loader decrypts them and spawns. As a result of this, Sonic's main component is activated on the computer. To keep itself up to date the loader checks for the latest versions of the main component every 10 minutes and updates it if necessary.

Main Component

The main component is a *Windows* EXE file of about 40 KB (it is compressed by the UPX PE EXE file compression utility – decompressed it is approximately 120 KB long). When Sonic's main part is activated on a PC (i.e. run by the loader component), it registers itself as a hidden process (service). So, it is clear that both the component parts of Sonic are invisible in the *Windows* task list.

Then it copies itself to the *Windows* directory (from the temporary directory) under the name GDI32.EXE and registers in the system Registry in the same key as the worm loader before it. In this way, the main part will get control at the next system restart instead of the loader.

The worm's main component downloads an auxiliary EMSMTP.DLL which is necessary for the email spreading routine. This DLL is non-viral and can be deleted if desired. Then the worm, depending on various conditions and circumstances, opens the *Windows* address book, retrieves email addresses from there and sends out infected messages. In known worm versions, these messages include:

```
Subject: Choose your poison
Attached file name: GIRLS.EXE
```

or sometimes:

```
Subject: I'm your poison
Attached file name: LOVERS.EXE
```

The email does not contain any body text. The worm works with *Microsoft Outlook* (*Express*), and any other email clients which use the *Windows* address book. Furthermore, some versions of the Sonic worm appear to send email messages to e_flemming2000@yahoo.fr or olivier1548@yahoo.com.

The main component also has backdoor facilities capable of watching an infected computer and using its resources from a remote host machine. To do this, Sonic listens on port 1973 or 19703 (depending on the worm version).

Additional remote access features include – screen capturing, access and modification of the file system (creating, renaming, deleting, copying or removing files and directories), downloading and uploading files, arbitrary execution, obtaining computer system and user information (drive list, OS version), obtaining dial-up networking passwords, displaying message boxes, obtaining address book and process control (get list of running processes, kill process) etc. During its operation the worm may create temporary files such as SNAPSHOT1 or MYKEYS.SYS.

Fortunately, the W32/Sonic worm does not seem to have provoked a global epidemic, thanks to the fact that in this realization, the loader component has no ability to replicate itself. It requires an essential connection to its Web site to download the worm's components. As soon as the Web site is closed, the worm dies.

Conclusion

The Internet is getting increasingly wide and ever more speedy. This presents hackers with the chance to build more complicated and bigger Internet viruses, with the potential to carry complex functionalities. Unfortunately, the Sonic Internet worm is not the first virus or Trojan capable of self-updating via the Internet. Before Sonic, the Babylonia virus had the same capabilities (see *VB*, February 2000, p.6), not to mention the Resume worm and several others. So, this is hardly newsworthy at the moment.

The more disturbing issue is that network-awareness seems to have become a new standard for malicious programs, and now most of them seem to be able to update themselves via the Internet. This is potentially a very dangerous trend, as it allows hackers to extend their malware capabilities in real-time with direct connection to the infected computers. And one more thing – never run programs from email attachments, even from people you trust!

W32/Sonic

Aliases:	I-Worm.Sonic, Sonic, and variations thereon.
Type:	Multi-component Internet worm with backdoor characteristics.
Self-recognition:	The appearance of a GDI32.EXE file in *Windows* and (or) *Windows* system directories, new Registry values in Run Registry keys.
Payload:	Backdoor capabilities.
Removal:	Remove the Run Registry entry, reboot the computer and remove GDI32.EXE files from *Windows* and *Windows* system directories.

TECHNICAL FEATURE

The New 32-bit Medusa

Péter Ször
SARC, USA

I remember the first time I was faced with MtE (the Dark Avenger Mutation Engine). Initial research showed that most anti-virus products were unable to detect 100% of the MtE-based viruses. Product tests carried out by Vesselin Bontchev at VTC showed that most scanners missed a certain percentage (occasionally as high as 10%) of infected files. If infected files were replaced from backups, sooner or later this initial 10% miss could build up to 100% on a particular system. Everything could be infected but the scanner would not be able to detect a single infection!

As virus writers developed polymorphic engines, scanners became stronger in their ability to defend against them. A virus scanner which used a code emulator to detect viruses looked like it was on steroids compared to those without emulator (virtual machine)-based scanning engines.

Nowadays, most polymorphic viruses are considered boring. Even though they can be extremely hard to detect, most of today's products are able to deal with them relatively easily. These are the scanners that survived the DOS polymorphic days; for some others DOS polymorphic viruses signified the 'end of days'. Other scanners died with the macro virus problem. In my opinion, for most products the next challenge is 32-bit metamorphism.

32-bit Encrypted Viruses

Virus writers have always tried to implement virus code evolution from the very early days. One of the easiest ways to hide the functionality of virus code is encryption. Among the first DOS viruses that implemented encryption was Cascade, which starts with a constant decryptor followed by the encrypted virus body.

This simple 'code evolution' method appeared in 32-bit *Windows* viruses very early too. The viruses Win95/Mad and Win95/Zombie use exactly the same technique as Cascade, the only difference being the 32-bit implementation. The detection of such viruses is possible without the trial of having to decrypt the actual virus body; a pattern based on the decryptor is unique enough to identify these viruses. Obviously, such detection is not exact. However, the repair code can decrypt the encrypted virus body and deal with minor variants easily.

32-bit Oligomorphic Viruses

Unlike encrypted viruses, oligomorphic viruses change their decryptors in new generations. Win95/Memorial had the ability to build 96 different decryptors for itself. Thus, detection of this virus based on the decryptor's code, while possible, was not a practical solution. Most AV products tried to deal with Memorial by dynamic decryption of the encrypted code instead. So detection is still based on the decrypted virus body's constant code.

32-bit Polymorphic Viruses

Win95/Marburg and Win95/HPS were the first viruses to use real 32-bit polymorphic engines. Polymorphic viruses can create an endless number of new decryptors that use different encryption methods to encrypt the constant part of the virus body. Some polymorphic viruses, such as Win32/Coke, use multiple layers of encryption.

Some of the newer polymorphic engines generate a decryptor based on a random decryption algorithm (RDA). Such a decryptor implements a brute force attack against its constant but variable encrypted virus body.

At that time, most scanners already had a code emulator capable of emulating 32-bit executables. Some virus researchers only implemented dynamic decryption to deal with such viruses. That worked in the same way as before because the virus body was still constant under encryption. Next, virus writers used a combination of entry point-obscuring techniques along with 32-bit polymorphism to make the scanners' job even more difficult. In addition, they tried implementing anti-emulation techniques to challenge code emulators.

32-bit Metamorphic Viruses

Virus writers waste weeks or even months creating a new polymorphic virus that is unlikely to get into the wild due to bugs. However, a researcher might deal with the detection of such a virus in a few minutes or at most a few days.

Obviously, virus writers try to implement various new code evolution techniques to make the researcher's job more difficult. Win32/Apparition is the first known 32-bit virus that does not use polymorphic decryptors to evolve itself in new generations. Rather the virus carries its source and drops it whenever it can find a compiler installed on the machine. It inserts and removes junk code to its source and recompiles itself. Thus, a new generation of the virus looks completely different. It is fortunate that Apparition has not become a major problem. However, such a method would be more dangerous if implemented in a Win32 worm.

The Win32/Apparition virus' technique is not surprising. It is much simpler to evolve the code in source format instead of binary. Not surprisingly, many macro and script viruses use a junk insertion and removal technique to evolve themselves in new generations. Igor Muttik explained metamorphism very concisely: 'Metamorphics are body-

polymorphics.' Metamorphic viruses have neither a decryptor, nor a constant virus body. They do not use a constant data area filled with string constants, but have one single code body that carries data as code.

Although there are some DOS metamorphic viruses, such as ACG, they have not become a significant problem for users. In a few short months the number of metamorphic 32-bit Windows viruses will probably exceed that of metamorphic DOS viruses. The only difference between the two is their potential. The networked enterprise gives metamorphic binary worms the opportunity to cause major problems. And as a result we will not be able to close our eyes to them and say 'we do not need to handle them since they are not causing problems to our users.' They will.

In December 1998, the virus writer Vecna created the Win95/Regswap virus. Regswap implemented metamorphism via register usage exchange. Any part of the virus body will use different registers but the same code. Obviously this is not all that complex. Below is a sample code piece selected from two different generations of Regswap.

```
5A              pop   edx
BF04000000            mov   edi,0004h
8BF5            mov   esi,ebp
B80C000000            mov   eax,000Ch
81C288000000    add   edx,0088h
8B1A            mov   ebx,[edx]
899C8618110000  mov   [esi+eax*4+00001118],ebx

58              pop   eax
BB04000000            mov   ebx,0004h
8BD5            mov   edx,ebp
BF0C000000            mov   edi,000Ch
81C088000000    add   eax,0088h
8B30            mov   esi,[eax]
89B4BA18110000  mov   [edx+edi*4+00001118],esi
```

The bold areas show the common areas of the two code generations. Thus, a wildcard string could be useful in detecting this virus. Moreover, support for half-byte wildcards such as 5? B? (as described by Frans Veldman) could lead to even more accurate detection.

However, depending on the actual capability of the scanning engine, such a virus might need algorithmic detection due to the missing support of wildcard search strings. If algorithmic detection is not supported as a single database update, the product update might not come out for several weeks or months for all platforms!

Other virus writers have tried to recreate older permutation techniques. The Win32/Ghost virus can reorder its subroutines like the BadBoy family of DOS viruses. The order of the subroutines will be different from generation to generation, and this leads to n! different virus generations where n is the number of subroutines. BadBoy had 8 subroutines leading to (8! = 40,320) different generations. Discovered in May 2000, Win32/Ghost virus had 10 functions (10! = 3,628,800 combinations). However, both of them can be detected with search strings. Still, some scanners need to deal with this kind of virus algorithmically.

Two different variants of Win95/Zmorph appeared in January of 2000. The virus' polymorphic engine implements a build-and-execute code evolution. Zmorph rebuilds itself on the stack with push instructions. Blocks of code decrypt the virus from instruction to instruction and push them to the stack. The build routine is already metamorphic. The engine supports jump instructions and removal between any build code instructions. Regardless, code emulators can be used to deal with the virus easily. The virus' constant code area provides identification since the virus body is decrypted on the stack.

The Win32/Evol virus – which implements a metamorphic engine – appeared in early July. Evol is capable of running on any major Win32 platform. Below is a sample code piece mutated to a new form in a new generation of the same virus. Even the constant-looking double word values can change in the pattern in newer generations, since the virus can calculate them (e.g. Magic=A+B). Therefore, any wildcard strings based on them will not detect anything above the third generation of the virus. Evol's engine inserts garbage in between core instructions. Here is an early generation:

```
C7060F000055    mov [esi],5500000Fh
C746048BEC5151  mov [esi+0004],5151EC8Bh
```

and one of its later generations:

```
BF0F000055          mov edi,5500000Fh
893E            mov [esi],edi
5F              pop edi
52              push edx
B640            mov dh,40
BA8BEC5151          mov edx,5151EC8Bh
53              push ebx
8BDA            mov ebx,edx
895E04          mov [esi+0004],ebx
```

Members of the Win95/Zperm family appeared in June and September 2000. This virus employs the same infection method as the PLY DOS virus. It inserts jump instructions into its code. The jumps will be inserted to point to a new instruction of the virus. The virus body is built in a 64 KB buffer that is originally filled with zeros.

Zperm will not use decryption. In fact, it will not regenerate a constant virus body anywhere. Instead, it creates new mutations by the removal and addition of jump instructions as well as garbage instructions. Thus, there is no way to detect the virus with search strings in either files or memory. Most polymorphic viruses decrypt themselves to a single constant virus body in memory. However, metamorphic viruses do not. Therefore, the detection of the virus code in memory needs to be algorithmic. Figure 3 explains the code structure of Zperm-like viruses.

```
instruction 2.
JMP instruction 3.
instruction 1. < Entry point>
JMP instruction 2.
instruction 3.
JMP instruction n.
```

Sometimes the virus replaces certain instructions with other equivalent ones. For example, the instruction 'xor eax, eax' (which sets the eax register to zero) will be replaced by 'sub eax, eax' which also zeros the content of the eax register. The opcode of these two instructions will be different.

The core instruction set has the very same execution order; however, the jumps are inserted at random places. The B variant of the virus also uses garbage instruction insertion and removal such as 'nop' (the 'do nothing' instruction.). It is easy to see that the number of generations can be at least 'n!' where 'n' is the number of core set instructions in the virus body.

Zperm introduced the RPME (Real Permutating Engine). RPME is available for other virus writers to create new metamorphic viruses. In October 2000, two virus writers created a new metamorphic virus, Win95/Bistro, based on the sources of Zperm and the RPME engine. To complicate matters, this virus uses a random code block insertion engine. A randomly activated routine builds a 'do nothing' code block at the entry point of the virus body prior to any active virus instructions. When executed, the code block can generate millions of iterations.

Win95/Bistro not only mutates itself in new generations. It also mutates the code of its host by a randomly executed code morphing routine. The virus might generate new worms and viruses this way. Moreover, the repair of the virus cannot be done perfectly because the entry point code area of the application can differ. The code sequence at the entry point of the host application will be mutated for 480 bytes. Figure 4 shows an original and a permutated code sequence of a possible entry point code.

Original entry point code:

```
55        push  ebp
8BEC      mov   ebp, esp
8B7608    mov   esi, dword ptr [ebp + 08]
85F6      test  esi, esi
743B      je    401045
8B7E0C    mov   edi, dword ptr [ebp + 0c]
09FF      or    edi, edi
7434      je    401045
31D2      xor   edx, edx
```

Permutated entry point code:

```
55        push  ebp
54        push  esp
5D        pop   ebp
8B7608    mov   esi, dword ptr [ebp + 08]
09F6      or    esi, esi
743B      je    401045
8B7E0C    mov   edi, dword ptr [ebp + 0c]
85FF      test  edi, edi
7434      je    401045
28D2      sub   edx, edx
```

Thus an instruction such as 'test esi, esi' can be replaced by 'or esi, esi', its equivalent format. A 'push ebp, mov ebp, esp' sequence (very common in high level language applications) can be permutated to 'push ebp, push esp, pop ebp'. Obviously it would be more complicated to replace the code with different opcode sizes but it would be possible to shorten longer forms of some of the complex instructions and include 'do nothing' code as a filler.

This is a major problem for all AV scanners. Heuristic scanners typically cannot deal with high level language written worms yet. Obviously some of these worms could easily be morphed to a new format. In my VB2000 conference paper I already introduced the problem of new virus variants being generated accidentally as a result of Portable Executable file repair. While it is unfortunate that such mutations can appear, it is feasible to deal with the problem. On the other hand, code permutations of worms and viruses, as performed by Win95/Bistro, will be much more difficult to deal with.

If a virus or a 32-bit worm capable of implementing a similar morphing technique should appear, the problem could be major. New mutations of old viruses and worms would be morphed endlessly and a virtually endless number of not-yet-detectable viruses and worms would appear without any human intervention, leading to the ultimate virus generator.

At the end of 1999 the Win32/Smorph Trojan was developed. It implements a semi-metamorphic technique to install a backdoor to the system. The standalone executable is completely regenerated during the installation of the Trojan. Its PE header will also be new and will include new section names and section sizes.

The actual code at the entry point is metamorphically generated. This code will allocate memory, then decrypt its own resource that contains a set of other executables. The Trojan uses API calls to its own import address table. The import table is filled with a lot of non-essential API imports as well as some essential ones. Thus, everything in the standalone Trojan code will be different in new generations.

Conclusion

It is only a matter of time until we see in-the-wild Win32 worms using metamorphic engines. Unfortunately, metamorphic viruses such as Win95/Bistro often have a random replication mechanism. Since their code structure is much more obfuscated, they are more difficult to analyse than polymorphic viruses. Their random infection and spreading mechanism will make the job of automated analysers and advanced behaviour-blocking systems more challenging.

We need to support detection of such viruses regardless of their complexity. It seems that scanning technology has to go through a new evolution! It is clear that by the time meta-morphism in viruses becomes complex, scanning technology alone will be inefficient as a primary anti-virus defence solution. It is going to be extremely difficult to deal with the rising number of potential false positives. Therefore, we must start to develop new systems and defences to reduce the inevitable overload in the future.

ERRATA

NT Comparative Update

On-access tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Aladdin eSafe Desktop	0	100.00%	11	98.44%	98.48%	191	95.16%	1144	80.09%	122	93.58%
Alwil AVAST32	1	95.65%	n/t	n/t	n/t	n/t	n/t	n/t	n/t	n/t	n/t
CA InoculateIT	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	2	99.61%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	10	99.86%	768	91.10%	3	99.81%
Command AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.98%	9	99.22%
DialogueScience DrWeb	0	100.00%	n/t	n/t	n/t	n/t	n/t	n/t	n/t	n/t	n/t
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
F-Secure Anti-Virus	0	100.00%	1	99.93%	99.93%	0	100.00%	0	100.00%	21	99.71%
GDATA AntiVirusKit	23	0.00%	626	22.33%	21.71%	1488	60.82%	623	83.30%	34	98.26%
GeCAD RAV	0	100.00%	1	99.74%	99.75%	8	99.79%	0	100.00%	8	99.25%
Grisoft AVG	23	0.00%	3	99.60%	96.83%	12	99.74%	292	89.47%	46	97.22%
Kaspersky Lab AVP	23	0.00%	1	99.49%	96.72%	0	100.00%	0	100.00%	1	99.81%
NAI VirusScan	0	100.00%	1	99.93%	99.93%	0	100.00%	99	95.71%	8	99.85%
Norman Virus Control	0	100.00%	7	99.49%	99.50%	26	99.46%	300	90.40%	2	99.77%
Panda AntiVirus Platinum	0	100.00%	0	100.00%	100.00%	26	99.35%	889	89.69%	52	98.21%
SOFTWIN AVX	23	0.00%	2	99.68%	96.90%	2	99.99%	56	94.36%	77	96.59%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	13	99.66%	191	95.24%	37	99.15%
Symantec Norton AntiVirus	0	100.00%	0	100.00%	100.00%	17	99.53%	264	94.74%	18	99.44%
VirusBuster VirusBuster	1	95.65%	25	96.55%	96.53%	66	98.34%	292	93.77%	10	99.01%

On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Aladdin eSafe Desktop	0	100.00%	9	98.58%	98.62%	191	95.13%	1144	80.09%	117	93.92%
Alwil AVAST32	0	100.00%	0	100.00%	100.00%	31	99.21%	28	95.36%	13	98.93%
CA InoculateIT	0	100.00%	0	100.00%	100.00%	0	100.00%	9	98.87%	2	99.61%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	178	96.37%	0	100%
Command AntiVirus	0	100.00%	3	99.78%	99.79%	0	100.00%	1	99.98%	13	99.23%
DialogueScience DrWeb	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	21	99.71%
GDATA AntiVirusKit	0	100.00%	1	99.49%	99.50%	0	100.00%	0	100.00%	2	99.71%
GeCAD RAV	0	100.00%	1	99.74%	99.75%	8	99.79%	0	100.00%	8	99.25%
Grisoft AVG	0	100.00%	2	99.49%	99.50%	11	99.71%	124	92.01%	30	98.67%
Kaspersky Lab AVP	0	100.00%	1	99.49%	99.50%	0	100.00%	0	100.00%	1	99.81%
NAI VirusScan	0	100.00%	1	99.93%	99.93%	0	100.00%	17	97.87%	7	99.86%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	0	100.00%	286	91.23%	0	100.00%
Panda AntiVirus Platinum	0	100.00%	0	100.00%	100.00%	26	99.35%	889	89.69%	50	98.34%
SOFTWIN AVX	0	100.00%	2	99.68%	99.69%	2	99.95%	55	94.36%	63	97.07%
Sophos Anti-Virus	0	100.00%	1	99.93%	99.93%	13	99.65%	191	95.24%	14	99.55%
Symantec Norton AntiVirus	0	100.00%	0	100.00%	100.00%	17	99.53%	264	94.74%	16	99.46%
VirusBuster VirusBuster	0	100.00%	25	96.55%	96.65%	66	98.34%	292	93.77%	10	99.01%

Regrettably, last month's *NT* Comparative contained a number of minor errors which, in turn, raised several issues regarding testing. The mistake which has the least effect upon the figures is, ironically, that which is in most urgent need of correction. Hawk-eyed developers at *Aladdin Knowledge Systems* pointed out that the ItW non-detection of Byway by *eSafe Desktop* showed a problem with the test-sets, since this virus should not have been on the WildList for September 2000.

The test-sets and WildLists were examined and the root of the problem found to be slight inconsistencies in the WildList relating to some of the viruses which, like Byway, had dropped out of the main WildList that month. This resulted in the incorrect version of data being used. This did not, in the majority of cases, affect detection rates by more than a fraction of a percent and virus collection upkeep has been safeguarded against future repetitions. This did not affect VB 100% award ratings, or any tests other than this. The charts here correct this matter and present the final results as they should have been.

There were also some problems while testing *DialogueScience's DrWeb* which affected the results here and raised important issues as to the *VB* testing protocol. Errors in testing resulted in *DrWeb* being erroneously declared to miss files which it did indeed detect. This leaves it with 100% detection of files, though this required a certain degree of tweaking. Under current protocol it is thus denied a VB100% award. The figures in these charts reflect results for default settings rather than detection capability, the same being the case for *AVAST32*.

Since the failure in these cases to gain a VB100% award is by design rather than inefficiency, it has been decided to implement new tools to provide testing of these products in default mode. Details of this change in protocol will be announced in the next Comparative.

FEATURE SERIES

The Usual Suspects – Part 1

Andreas Marx
University of Magdeburg, Germany

Most of today's anti-virus software detects nearly every known virus, even very complex polymorphic ones. However, to be good in the 'virus scanning' category there is much more out there to detect than simply all in-the-wild or zoo viruses.

This feature concentrates on virus-related problems in AV scanners, and how developers can avoid them. It is based both on results from our various tests (see http://www.av-test.org) and on comments received from IT representatives in large, international corporations. Most of the points raised look very simple, but they are all too often overlooked. This first part starts with trivial issues while the second instalment will reflect on more complex problems.

File Extensions

Most scanners do not scan all files by default – they use an extension list. Since new viruses have started to target 'new infectable' extensions, a program has to update this list with every scanner update. A better idea would be to scan all extensions by default to avoid this problem. However, there is usually an associated performance dip and, sometimes, additional heuristic false positives will be triggered.

Scanning everything on-access will cause huge performance problems if the scanner is dumb enough to scan everything every time, even if the file is unchanged or cannot be infected at all. Other problems will be caused with temporary files and very large files – it is not a solution, but it helps if the maximum size of the file to scan can be configured. However, at email gateway level 'scan all files' should be the default setting, since files can be renamed too easily to avoid detection.

Some scanners do not actually scan all files even when set to 'scan all files' or when the mask '*.*' is used. Most of the time at least some infected .BAT, .VBS and .COM files will be missed if they have non-standard extensions. This happens when the scanner checks the file extension, not the content, in order to scan solely for this kind of virus. It would be a good idea for vendors to make a 'smart' scan to find out the (hopefully) correct file format. If there is more than one possibility (like ASCII text or a .COM file), all possible supported formats should be scanned.

In most programs, the inclusion or exclusion extension list allows only 3-byte long strings. This is fine for .COM or .EXE files, but what about larger extensions like .CLASS – these have been found in the *Windows* world since at least *Windows 95*. Some scanners do not allow them to be scanned (unless in 'all files' mode), others look for extensions like '(*.)CLA(*)'. The latter is probably the best as there are often old volumes on file servers which cannot handle long file names. A user should be aware how the scanner handles such 3-byte extensions. Currently there are no known ItW viruses which infect files with more than a 3-byte extension but there are some zoo viruses which do. An interesting idea would be to export the 3-byte extension limit into the Unix world: some scanners under *Solaris*, *FreeBSD* or *Linux* show the same behaviour in this regard.

On-demand scanners usually use an extension list different from that of the on-access scanner (e.g. without archive file extensions). The on-access extension list cannot be configured in many programs, and in some scanners there is not even the option to scan all files on-access.

Another problem is caused by files with no extension at all. For example, many of the *Excel* macro viruses drop a file into the XLSTART directory. For this, many scanners have a special option on their default extension list – 'Scan files without an extension'. Unfortunately, not all of them handle extensionless files correctly – some do not scan for them, taking the real name as the extension – and often the file is left unscanned. If the option to scan all extensionless files does not exist, there is usually no way to add an empty value and all the files have to be scanned. A good point to make while discussing extensions is that no scanners seem to have a problem with double extensions like '.TXT.VBS'.

Scanning Options

Some scanners have really interesting default settings – usually they are optimized for speed, but not for security. Such settings start with a list of ten file extensions for the on-demand scanner to look for. No archives or packed programs will be scanned at all. Therefore, infected files could be missed, even if the virus scanner is capable of finding them. It would be better to scan all files by default, if not all archives too, in the first (automatic) scan of the whole system. If no virus is found, it can be switched back to an extension list until an infection is flagged.

Often, only one possible option exists for dealing with many types of infected files. Even on a desktop and especially on servers and mail servers, it is important to have different settings at least for macro and non-macro viruses. It would be better to divide them into boot, file, script viruses and other malware. For example, a user would be able to specify that script viruses and Trojans be deleted and macro viruses be cleaned.

Most of the time, there are different options for what to do with infected files – clean, copy, move (isolate), delete, rename, allow or deny access, print a page, beep and shut

down the computer, and so on. Occasionally these options can be used together (like rename and move) but, more dramatically, if the option fails, nothing will be done. It should be possible to have a second option in case the first fails. For example, 'try to clean, and if that fails, delete the file' is often used by customers. Some scanners, especially ones on mail servers or gateways, allow only one setting – if the cure of an attachment fails, it will be delivered …oops! An extra option to make a backup copy of the original file in a special quarantine folder before taking any action should be a standard setting.

An option to switch on or off the protection against back-doors and similar malware should be implemented. This would avoid legal issues and provide the user who requests it with real protection. It would be useful to add a switch for detecting jokes, too, since most home users want to have these programs while corporations do not. In some cases it would be helpful to exclude only some 'virus names' from the detection. This could not only be useful in the case of false positives, but also if the user wants to use NetBus (and only NetBus), for example.

Report Files

A standard report file should at least include information about the scanner, the version and date of both the program and the signature file(s), and the options used for scanning. The current date and time, the user and computer names should be included, too – at least once with a desktop product or with every entry if it is a server or mail server scanner. Some anti-virus scanners still do not include this essential information in their log files.

Every virus found and the action subsequently taken should be included in the report file, together with the full path, file name and why the action has been performed. With a mail server product, information about the sender and the recipient should be added. In our tests, we frequently came across unusable log files – the exact path or file names were truncated and replaced by '..'. In the case of archive files, only the file names could be found, but neither the archive name nor the path to the archive were located.

The log files should be exportable to at least text or comma-separated value (CSV) files. HTML-only log files are better to read if a browser is available, but should not be the standard or only setting, since they are infectable and harder to import into other programs. All entries should be separated by correct line feeds (e.g. 0x0d/0x0a for *Windows* programs) and the length of the report file should be unlimited. However, some anti-virus scanners currently have problems exporting log files with more than 1,000 entries. Really huge log files of several MBs will often be truncated at a random position without an error message.

In good documentation it should be possible to include all the files which have been scanned, not just the infected ones. For desktop products, it is useful to truncate the log files automatically if they are too big (1–2 MB rather than 50 KB), but on server software this option should be turned off by default. A short statistic or overview function of how many files have been scanned, how many are infected, how many have been deleted etc. is also useful.

Error Messages

Many AV scanners try to avoid displaying error messages and others' messages are incomprehensible, like 'PK-F-Init failed. Return code = 0x25628'. If an operation has failed, for example the removal or cleaning of a file on a write-protected drive, an error message must be displayed and included in the log files. Some programs do not do this – they look as if they are cleaning viruses correctly even if they cannot do this for physical reasons. So, the virus is still there, even when the program says it has been 'successfully' deleted or cleaned.

The same happens if a file cannot be opened, changed into a directory or scanned, if it is locked or if the user does not have the right to access it. Most scanners will skip such files without any notice. This is not acceptable, especially on *NT* or Unix systems with a user rights system. In the case of a password-protected (archive) file, a scanner should write a comment into the log file indicating which encrypted files cannot be scanned. Most of the time, the scanner will not report anything, or it will give a wrong 'OK' message or report internal errors, not specifying the real reason. Of course, the scan statistics should show the number of files which could not be scanned.

Translation

In some programs the translation of documentation is really ugly. This applies not only to error messages (some, translated verbatim, are nonsense), but also to the program itself and the on-line help. An example would be the use of the word 'exchangeable' instead of 'removable' in the case of a virus being cleaned. Others describe scanner options wrongly or are shortened – the English version is usually shorter than most other language versions. In this case, there should be enough space left for the translated strings.

Command Line vs GUI Versions

In many cases, command line scanners are much more powerful than GUI versions. Even if virus researchers and some companies choose this kind of program exclusively, it should be made clear that most, if not all, the additional functions are implemented in the GUI version, which is used more often in general practice. These functions include some speed-up or exact detection of viruses, and also recursive scanning for more types of compressed and archived files in memory. The GUI version only scans for certain files and then not recursively, with temporary extraction onto hard disk. Some complex polymorphic zoo viruses can only be detected with command line options, which are obviously not available in the GUI version.

Next month we will look at more complex problems.

TECHNICAL OPINION

Scriptobabble

Paul Baccas
Sophos Plc, UK

The rise of Script Viruses (I will use the word 'virus' for consistency but it can be changed to worm or Trojan in this article) over the last couple of years has presented some interesting problems as regards their detection. One AV company has recently introduced a special scripting module, and no doubt others will follow. The fact that the languages the scripts are written in are robust, powerful and pervasive has meant that they can, and do, spread over and through a myriad of different systems.

'God takes a text, and preacheth patience'

Whilst in traditional viruses (except polymorphic ones) the binary does not change, in scripts this is not necessarily true. I believe that there is some isomorphism between the problems presented by script viruses and those which were faced by the industry on the rise of macro viruses. This belief has been held for a while, and this article will attempt to formalise the arguments and hopefully convert readers.

Setting aside general virus detection problems such as exactness and co-ordinated nomenclature, we are left with several groups of problems:

- Manual manipulation, where user interaction changes the viral code;
- Artificial manipulation, where the system changes the code;
- Deliberate manipulation, where the virus changes itself.

These groups can be rejigged as:

- 'White space';
- Differences in file formats on different systems;
- Polymorphism;
- Padding operations.

'But thou read'st black where I read white'

Below, some of the problems listed are described more fully. However, this is not necessarily a complete list, for various reasons.

The **'white space'** category can consist of a number of variations – a user analysing viral code and inserting blank lines, tabs and/or spaces for 'readability'. If they execute the code, publish the code and it is downloaded, or leave code where the system will execute it, they have created a 'new' virus.

The editing may not be deliberate – their Text Viewer may interpret a tab (0x09) as 'n' spaces (0x20) or vice versa or possibly change other things. A poorly written Mail Program (either client or server) may add seemingly null characters to what is effectively a text file.

It is more likely, however, that the 'white space' has been removed to obfuscate the code, to see 'how good the scanner is', or by a programmatic error in viewing or transporting the virus.

There is an issue of where in the code these changes occur, as they may alter the actions of the virus. However, in a large proportion of cases, this is not so. These are all trivial changes and the average user would expect these not to cause any problems.

Look at this file snippet:

```
0000–094F6E20 4572726F 72205265 73756D65–
.On Error Resume
0010–204E6578 740D0A20 20202020 20202020–Next
```

Is this one really so different?

```
0000–204F6E20 4572726F 72205265 73756D65–
 On Error Resume
0010–204E6578 740D0A20 20202020 20202020–Next
```

Within the **differences in file format** category, there are two main problems. The End of Line (EOL) marker is different under different systems and can also differ between applications. So, the EOL marker, either 0x0d 0x0a, 0x0d or 0x0a, is not constant and can even change between replicants (if sent via a Mail or IRC server) or if opened on another system.

The second issue here is in the way that foreign characters are processed by intervening systems. For example, with JS/Kak a number of files run into trouble with the French directory name.

JS/Kak.A:

```
0490–7e315c5c 5c5c4490 4d415252 7e315c5c
;~1\\\\D MARR~1\\
04A0–5c5c6b61 6b2e6874 61273a6b 656e3b74
;\\kak.hta':ken;t
```

JS/Kak.B:

```
0490–7e315c5c 5c5c443f 4d415252 7e315c5c
;~1\\\\D?MARR~1\\
04A0–5c5c6b61 6b2e6874 61273a6b 656e3b74
;\\kak.hta':ken;t
```

These two snippets show differences between JS/Kak.A and JS/Kak.B. All are to do with how the acute 'e' (é) is handled, except for one extra space (0x20) at 0x360 in the HTM part of JS/Kak.B.

Polymorphism is a general problem with all viruses. However, once the normal tricks have been looked at, script viruses have other issues which include:

- The insertion of padding operations;
- Changing variable and string names (these can either be from a fixed list or at random);
- Encrypting parts of the code (normally the decrypting part is still plain).

Also, due to the fact that most scripting languages are not case-sensitive, we have the problem that a variable can have many different incarnations depending on its case. While polymorphic viruses already employ most of these tricks, the script kiddies may also make use of them.

Padding operations can be a number of things. These include the insertion of null operations, from If, Do and For loops that do nothing, through assigning and setting variables that consequently are not used, to writing functions that are never called. A common padding operation is the random insertion of comments with, of course, VBS.

The troubles resulting from 'white space', polymorphism and padding operations are due to manual and deliberate manipulation. The concern with file formats is to do with artificial manipulation. The latter should be dealt with automatically, as all of the possible changes can be determined beforehand. Problems resulting from manual and deliberate manipulation also need to be solved, but different scripts need different solutions. In some cases, automatic solutions are needed, and in others manually crafted solutions work best. Often a combination of both is required to get the job done.

'Shall I compare thee to a script virus?'

Apologies to Shakespeare! What, then, is the isomorphism between script and macro viruses? Vesselin Bontchev's VB'97 paper entitled 'Macro Virus Identification Problems' described the above problems in relation to VBA5 macros. To summarise the relevant parts of that conference paper:

- Empty Lines – describes the possible prepending of blank lines when VBA3 and VBA5 macros are up- and downconverted.
- White Space – describes tabs being converted to spaces when WordBasic viruses are upconverted to VBA5, and other white space problems.
- Letter Case in the Identifiers – describes a way of ignoring the case of variables by a canonicalisation of variables.
- Insertion of Do Nothing Lines – describes some of the null operations discussed above and suggests that ways have to be found of ignoring them.
- Variable and String Modification – describes a possible way of canonicalising these elements to ignore trivial modification.
- Commenting and Uncommenting Lines – describes a possible encryption trick as well as a padding operation with the warning that you should not always ignore these lines of code.
- Encryption – describes some of the potential problems and solutions.

The paper goes on to describe other possible macro identification problems, some of which are specific to macros and some that are general in nature.

For each item in the groups of problems selected, with the exception of those associated with artificial manipulation, there seems to be a direct correlation to problems in macro detection. A good solution for macro viruses should be able to provide the basis for a good solution to script viruses. Some further pre-processing is needed to provide a direct correlation and to fix the artificial manipulation issue.

'The answer, my friend …'

There are, however, one or two more issues with script viruses that need to be considered in order to deal with their detection efficiently. The first is the result of another piece of artificial manipulation which was not included above because it invariably produces a non-working piece of code.

This problem was especially prevalent during May of this year. It is to do with the curious phenomenon of mail programs or gateways wrapping the code to 80 characters or, more bizarrely, removing nearly all EOL markers to give one continuous block of text. The resulting code will almost always be treated by the Script Interpreter as having errors. However, it is possible that it will not.

This is similar to yet another white space problem, that of line continuation characters – so, in theory, the majority of these corruptions may be caught. When the pre-processor is being developed consideration should be given to how to handle this kind of glitch.

The other problem that may be encountered is due to the potential complexity of the script viruses themselves. VBS/Newlove brought this issue to the fore with approximately 100 lines of actual code capable of creating several thousands of lines of code (we created a replicant of 810 KB before we got bored).

Newlove had a couple of bugs that meant it did not spread as widely as had been feared. The resulting rash of fixes for Newlove showed the need for a thorough solution for script problems, especially when the heuristics of one anti-virus product detected this virus in the HTML documentation of a *Windows* version of Perl.

To summarise, the problems of script virus detection are similar to those which have long been solved in relation to macro viruses. With a careful design of the scanning engine both macro and script viruses could be dealt with using the same core module despite having different pre-processors.

OPINION 1

EICAR Surveys the Scene

Rainer Fahs
EICAR, Belgium

I became aware of the *European Institute for Computer Anti-Virus Research* (*EICAR*) in 1992 when I was responsible for the implementation of AV defence in an international special project. I joined in 1993 and became a board member, responsible for the co-ordination of *EICAR* Working Groups, in 1995. In 1996 I was elected Chairman of the Board and was re-elected in 1999. When *EICAR* was founded in 1991, it was the intention of the pioneers involved to build a common platform. Here, computer users could articulate their requirements and developers of anti-virus products could listen, building products that would satisfy users' requirements.

When we look back faithfully, we have to acknowledge that this idea has materialised in a very limited way. How many discussions have we had about scanning for known viruses being only the second best solution? How many times have we heard the numerous questions about heuristics and their implementation, and the 'expert system'? In the meantime, anti-virus products have been adapted to new operation environments and scanning engines have been improved. However, developers and vendors of AV products – and, for the sake of this article, I include also products that scan for Trojan horses or other malicious malware – are forced to spend a lot of time getting samples of new viruses to include in their products.

So what's new? Nothing has really changed for years. At *EICAR* or *VB* conferences we hear presentations on ideas for improvement of the defence against malicious effects on our computers and networks. We listen to recommendations for the enhancement of products, proposals for a change in defence methodology, and requests for new or improved laws. We experienced a long period of discussions on the transition from anti-virus to anti-malware.

Even the consideration of testing facilities does not paint a brighter picture. The publication of results from various testers shows there are undoubtedly some good AV products on the market. But are these products the best technical solution to a system's engineering problem? As long as the methodology is not countered with a solid systems engineering approach, I have my doubts.

However, part of the problem seems to be a legacy issue. People like Alan Solomon – one of the first to write a scanning engine to find and clean viruses on PCs – have developed products with the best of intentions. They never dreamed about viruses appearing in exponential growth rates and PCs interconnected to networks or hooked up to the Internet – at least, not in the beginning. What these pioneers built was a product that was good at finding viruses, but was it responding to a real user's requirements?

For example, is finding, identifying and disinfecting the virus the solution? Put it this way, I have never seen a set of implementation-independent user requirements that point to a specific product which satisfies all that the user is looking for, maybe even verifiable by independent testing. But if we know that AV products are only the second best solution, why are they selling? Maybe they are the best commercially available products, leaving the user no choice?

There are many questions like these and there are no unanimously agreed-upon answers. To find the answers one must find out what the real user's requirements are. In a brainstorming meeting with Sarah Gordon at the last *VB* conference in Florida, we discussed these issues – again – but this time we agreed that there was a requirement for action – and an idea was born. Why not try to find out what the real users' requirements are?

It was and is *EICAR's* objective and its strength to unite efforts. As an independent non-profit organization with a large corporate membership, it is best equipped to initiate a global survey to discover what users' requirements are – and to draw conclusions for subsequent improvements. To take the first step – and there are many more to follow – *EICAR* has initiated the *EICAR* Anti-Virus Enhancement Programme (EAVEP) in the form of a research project with an ambitious objective.

The programme will consist of a data-gathering survey and a data analysis and interpretation endeavour, which should yield recommendations for improvement. This venture will be carried out with close cooperation with our members, in particular those from large corporations, academic and commercial researchers and AV developers and vendors.

At our last Board meeting on 21 October 2000, the Board approved the program and we are happy to announce that the University of Aalborg in Denmark have agreed to lead this operation, with Professor Urs Gattiker as the project leader . Currently, Sarah Gordon from the WildList Organization (*WLO*), Andreas Marx from Magdeburg University in Germany, and Robert Niedermeier, a German lawyer working for *Price Waterhouse Coopers* form the core of the survey team. The first task for the project operators is the development of questionnaires and more volunteers are welcome. The first milestone will be the *EICAR* annual conference from 3–6 March 2001 in Munich. There will be a dedicated session concerning the survey on the conference agenda and a special meeting with vendors and developers.

For more detailed information about the programme and conditions for participation, please check the *EICAR* Web page; http://www.eicar.org.

OPINION 2

Adjust Your Attitude!

James Wolfe
Lockheed Martin Corporation, USA

I sometimes feel like a weatherman with a part-time job as a car salesman. Just as the guy on the news predicts the weekly weather I get to predict what type of viruses might be coming, and the level of protection I will need to employ to protect my company. Those of us who do make these predictions have the near impossible task of selling them to the pivotal people in our corporations so that they don't think we are just crying wolf.

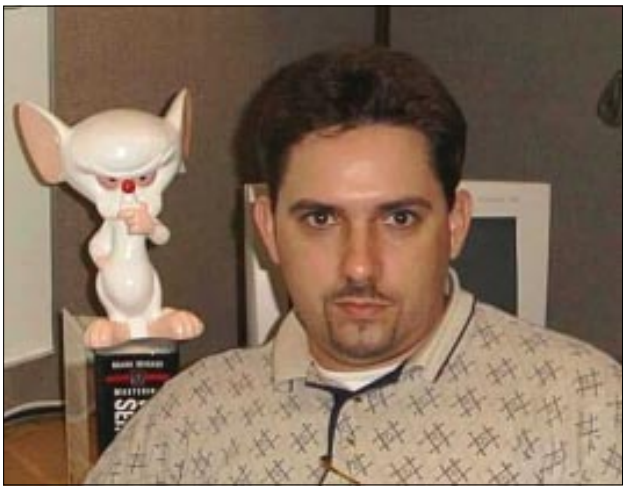
For example, many people in the industry believe we are on the cusp of a new wave of viruses that will be *Linux*/Unix-based and will cross over to that aloof group of Unix users which often says 'I use Unix so viruses can't touch me'.

I'm fortunate to have receptive server administrators who take my recommendations seriously but, how do you convince someone who doesn't respect your knowledge they should spend thousands of dollars on a solution for something that doesn't exist yet? How do you communicate to someone that the best you can expect from your solution is that nothing will happen? How do you get the policies implemented to make your job easier?

Recently, I attended a conference where I had the opportunity to meet many people who share the same daunting task that I do protecting their corporations from viruses. Most of these poor souls have the job of tracking viruses, testing products, pushing out updates of existing products, and reporting virus statistics. More often than not this is on a limited budget with limited manpower. Some are doing the job part-time and many of them have little of the actual organizational power that would allow them to do their jobs properly. Most of the conversations I had involved having to deal with 'stupid users' and not being able to get the resources needed to do the job.

As I listened to the various reasons (and, frankly, endless whining) being given for their problems, I thought to myself that the problems that AV people encounter are very often brought on by their own attitudes. Being something of a technical 'weenie', I sometimes forget that I perform a service to a customer.

Realizing that I serve a customer (albeit, in my case, an internal one) was an eye-opening experience for me. I had fallen into the trap that there were no grey areas when it came to my recommendations. Newsflash folks, a hard-line approach like this will cause people to go on the defensive, with the result that nothing you say will be heard. This is what causes many of us to lose our credibility and more importantly, funding.



So what is the answer? Instead of telling your audience what you want, *sell* them what you want. As with many of my peers, I work in the Information Security department. Most people see any department with the word 'security' in its name as the enemy. To overcome this my supervisor, who has a degree in Marketing, suggested that I should *sell* what I do instead of just telling what I do. So when I receive an inquiry from a client regarding a particular virus, instead of saying 'yes, we're protected' I'll say, 'We've been protected for 3 weeks, and here is a little information about it, thank you for your help'.

What's the difference? One is directly to the point and the other is user-friendly. Always let your customer know that you're happy to help and that it is all part of the service, even if the question is about an annoying two-year old hoax. When you attend inter-departmental meetings introduce yourself to employees you don't know and let them know that they can call you if there is ever anything that you can help them with. Many times this has put me in direct contact with Managers and Directors – the very people who will be approving the new software I need or the new policy I need support on.

The bottom line is 'get over yourself' and change the way you approach the whole process. Most of you reading this article have the technical skill but do you have the people skills? Sell yourself and your service. Not getting the funding and the staff you want might not be the problem. *You* might be the problem.

Before you can get people to provide the tools you need, they have to want to listen to you. You have a much better chance of getting funding, more people, or a new policy if the people who perform these services think of you fondly. I know it sounds campy and is terribly friendly but those 'stupid users' pay your salary, and let's face it a grumpy virus researcher without a job is an unemployed nerd with a bad attitude.

PRODUCT REVIEW

GDATA AntiVirusKit v10

Matt Ham

AntiVirusKit (*AVK*) is a familiar, well-respected product in another form – it is built around the engine of what used to be known as *AVP*, now known as *Kaspersky AntiVirus*. Readers of last month's *NT* Comparative Review will note that *AVK* was reviewed briefly there and seemed to show a distinct weakness in the field of on-access macro virus detection – a weakness due more to testing difficulties than any proven lack in detection capability.

It seems a little harsh to make such figures public without the redress of a fuller look at the product, and thus this review was born. *GDATA*, the German manufacturers of *AVK* – known as *AntiVirenKit* in its native land – controls interests in several categories of business, of which computer security, and thus anti-virus measures, is only one. Listed among *GDATA's* other manufactured products was another anti-virus program, marketed under the catchy name of *SmileWare*.

Like several other recently reviewed software manufacturers, the English-speaking market looks tempting to *GDATA*, and thus the *AVK* package is in the process of being converted into an English version. There has also been a recent new release of the software itself, and so not only is this review fresh with news for the English-speaking world, it should also be relevant to those who have got to know the German version of *AVK*.

Testing Protocol

Reviews were performed, in the main, on a machine running *Windows NT v4* with Service Pack 5, for direct comparison with results in the *NT* Comparative. Some tests which did not concern viruses were performed on a *Windows 95* machine with dial-up Internet access.

For security reasons, update testing could not be performed on lab machines directly from the Internet. All scan results included here and the discussions about *AVK's* interface refer to the *Windows NT*-tested version unless otherwise made clear.

Contents and Documentation

The review package received in the post contained no great surprises – a CD, a German manual, a registration card and the installation instructions on a card liberally sprinkled with the word 'Achtung!'. As this is about my limit as far as German is concerned, the manual was turned over for scrutiny by Bernadette, VB's Office Manager and resident linguist. This was declared to be, if not the most interesting read in the history of literature, filled with sufficient detail to provide users with a good, solid guide to installation. The manual is quite weighty and contains that most valuable of things – a full index. Thus, the physical documentation in German passed this review with honours – though this is not always an indication that translations will be of the same quality.

So how did the translations fare? An extra set of documentation in English was also included, though folks at *GDATA* stressed that this was only a draft copy and that the full English version was still in preparation. Even so, the quality of translation was very high, both here and within the product itself, and although a slightly Teutonic style in some sentences hinted at the origins of the translated text there were no real oddities.

In true *Virus Bulletin* tradition, I am duty bound to report on the state of *AVK's* packaging. The box was sadly neither flimsy nor robust enough to provide a major topic of discussion, though the exterior text did include some interesting teasers and, more remarkably, useful details as to capabilities and features within an external fold-out flap.

The registration card also contains the application form for the various support options offered for *AVK*. These include (as part of the enhanced Premium Support) Internet Updates and CD-ROM updates twice a year, a Premium Hotline and an Emergency AntiVirus Service.

An alternative 'off-line' version supplies six update CDs and no Internet Updates. This is already available both inside and outside Germany, though the European Community qualifies for discounts in comparison with the rest of the world. Current prices for this support are DM135 per year within the EC, rising to DM170 for the rest of the world. This covers a single machine, with multiple machine deals available on application.

CD Installation

AVK is installed by the ubiquitous InstallShield after an auto-running front menu offering installation, exploration of the CD or additional multimedia content. The user agreement and location of installation are quickly followed by the almost as ubiquitous choice of Minimal, Standard and User-Defined installation procedures.

The choice of Minimal only installs the 'Integrated user Interface' and help files, which does not include any on-access features and simply provides an on-demand GUI scanner with help. This is indeed 'Minimal', and thankfully not the default setting. Standard, predictably enough the default, offers a virus encyclopaedia, on-access components and right-click scanning to the installed features. It is only with a User-Defined installation that two further options become available.

One, information about *GDATA* and its product ranges, contact information and the like, is relatively unimportant as far as virus detection ability is concerned. The other, *AntiVirusKit Office*, is of rather more importance.

AntiVirusKit Office

AntiVirusKit Office is the component responsible for the lowering of *AVK's* detection rates in our Comparative testing. This may seem strange indeed, though there is more to matters than that statement might suggest. *AVK*, in the version tested in the November Comparative, did not detect macro viruses on-access by default. This situation has since been changed, and the newer version did scan these by default. Since macro viruses require what is, in effect, an extension to the operating system (usually *Microsoft Office*), full pre-execution scanning can be performed within that environment.

This is what *AVK Office* is designed to do, by integrating within *MS Office*, *Outlook* and *Exchange*. This is a trickier method of scanning to test with the traditional tools and not, in any case, a default option in this standalone review. However, the testing of such capabilities is well within the scope of investigation.

After selection of the installation type, the installation process is swift and no problems were encountered. A request to register and a reboot completed the process. This left a desktop shortcut, monitor tray icon and a start menu addition as the means of accessing *AVK's* functions.

Full Installation

If users find themselves more paranoid than to allow for an installation from within *Windows* where, admittedly, there is a good chance of pre-infection, especially in the case of an emergency installation, there is provision made for a slightly more drawn-out but secure method of *AVK* preparation. Details for this are emblazoned on the installation card provided in the box, so although the CD autorun does not mention this method of installation, there is no excuse for ignorance on the part of the user.

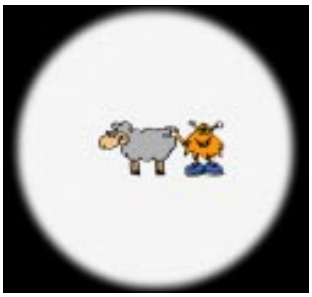
The instructions here are actually remarkably simple, though caveats do exist concerning SCSI drives and the like – simply setting the CMOS so as to make the boot from CD-ROM the default. Once performed, it is then possible to use the CD as a *Linux* boot disk, the only real purpose of which is to run a scan of the machine automatically under *Linux* to ensure that a target machine is clean for installation from CD.

Anti-virus companies have long worked on this pre-install scan problem – most suggesting clean boots from known, uninfected disks. The simple expedient of providing such disks has generally been prevented by *Microsoft* holding copyrights which do not allow such distribution of their proprietary code. In at least one related case, that of *Dr Solomon's Magic Bullet*, this copyright was circumvented by the writing of a mini OS for the purpose of scan hosting. *Linux* offers another alternative, with less overall complexity than is required to write a mini OS capable of dealing with CDs and NTFS.

When run on an *NT* machine the pre-scan had no problems in scanning the usual boot drive, though non-Western keyboard mapping made interrupting the process less than simple. It was also less than obvious choice what should be done when an infection, placed for the purpose, was actually encountered. A command prompt appeared with the presumed default 'Delete' followed by ('OK', 'Report Only', 'Disinfect', 'Cancel', 'Stop'). Pressing return seemed not to prompt any action at all and with keyboard mappings askew and no indication what was to be done to invoke any of the responses, it was a case of a good idea marred by less than perfect implementation.

The scan process itself also displayed some oddities, with some files seeming to be scanned multiple times in a row. With a little more finesse this could be a very welcome addition to the installation procedure. In a final addition to the installation repertoire on *NT*, *AVK* also provides links for extra product downloads in its start menu folder – these being for *Web Speech* and *Palm* management software, which were not within the scope of this review. The option to uninstall was also available here.

This was executed without major problems, though some .DLL files were declared to be in use and undeletable during the process. These did seem to be removed upon reboot, however, as were all files and folders including log files other than the *AntiVirusKit* root folder. The *Windows 98* installation version showed one major addition here – an option to update or create emergency disks was provided. This allows for a customised set of emergency scan disks to be produced in an automated fashion.

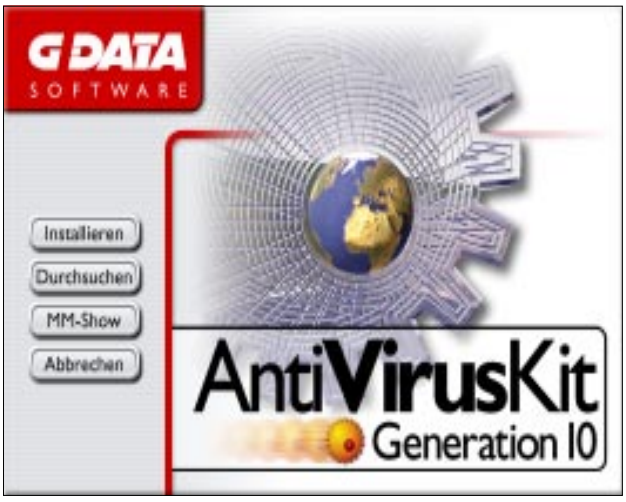


As mentioned earlier, the opening menu multimedia material is available on the CD and accessible from within the program. This provides short presentations on computer viruses, their effects, *AVK* itself and associated topics.

Some of this material will already be familiar to *Kaspersky AntiVirus* (*AVP*) users, though most of it has been specially prepared by *GDATA* to accompany *AVK*. Several of the talks are presented by filmed humans, while others employ animation, like this poignant warning [*taken early in a sequence! Ed.*] of the dangers of Trojans.

The GUI and Monitor

The GUI for *AVK* consists of the popular main iconic mode selector which takes up a strip on the left of the interface.



A mini toolbar stretches across the top of the GUI, below the drop-down menus, while the rest of the screen has contents dictated by selections in these other areas. The days when each option could be reached through only one path are now all but vanished, and the complex interrelations of controls make this something of a difficult area to describe adequately. However, the icon and tool bars are as good a place to start as any.

The bar is used to select the function of the main screen area and consists of the options Analysis, Monitor, Schedule, Internet Update, Records and Information. These fairly standard features are added to on the tool bar, which has control icons for the *AVK* Wizard, Help and Virus Encyclopaedia, with additions dependent upon which main area is selected. For example, this adds drive selections, advanced scanning options, scheduling and scanning if the analysis main screen is active.

The Analysis section, relating to on-demand scans, is the heart of control over the *AVK* GUI. This is divided between a representation of directory tree structure in not quite standard Windows style, and the control area for Settings and Objects.

Settings provides the user with the choice of whether heuristics, check-sums or log are activated, with a further choice of 'Advanced' options. These include settings applicable to both on-demand and scheduled scans and are mainly concerned with the actions to be taken if a virus is either detected by scanning, or suspected through integrity check methods.

The settings for actions follow the usual Report Only, Quarantine, Delete, Disinfect or Stop Scan range in the case of on-demand settings, while the integrity checker allows Record, Stop or 'Accept Deviation'. The last setting appears to be one which is unlikely to enjoy much use. It seems to be a method of employing an integrity checker in action, with the associated overheads, while completely ignoring any information that might be provided by it. Perhaps understandably, this is described in the documentation as a setting that is only of use in cases where viruses are certainly not a problem – though it would still seem wiser to use the record option just for peace of mind.

Objects to be scanned are selected with fewer available options – compressed files, system files and memory are all selectable and activated by default. Types of files to be scanned are set as Automatic Type Recognition by default. Program files, All files and User Defined files form the alternatives. The documentation stresses with emphasis that although All Files is included, it is most certainly not a recommended setting.

The Monitor section, relating to on-access scanning, can also be reached by right-clicking on the tray icon. The options here include Objects and Action on Infection choices similar to those available on-demand. The monitor may also be deactivated here and limited statistics viewed. There is a further level of selection for Excludes and the more mysterious Plug-Ins, the latter appearing to have no options in the copy reviewed.

Next on the list comes the Schedule section, which at first appeared to have no commands associated with it at all. The simple expedient of reading the instructions, however, led to the realisation that the *AVK* Wizard is the method by which jobs are scheduled. This Wizard is of dual use since Internet Updates are considered to be jobs in the same way as are Scans.

The job process involves the selection of areas from a tree, though lack of an internal refresh facility here made CD scanning not quite as straightforward as the scanning of local hard drives. Objects and File types are chosen next, following the standardised settings already discussed, with a further set of choices to be made as to heuristics and the like, again in the same manner as the Analysis section. At this point one new option is given, that of whether to operate the scan in the background or the foreground. Foreground scans have the option of a subsequent quit, depending on whether viruses have been discovered in the associated scan.

Finally, the time that the scan is to be performed must be chosen – this can be immediate, on boot-up, on idle, once, or at regular intervals. The control here is less 'fine-tuneable' than some products on offer with, for example, the days of the week all treated equally. However, this can be seen as having a good side, inasmuch as the common but ridiculous options of scans scheduled every second or century are also unavailable.

Moving onwards, the next section is Quarantine which, unsurprisingly, allows viewing of files which have been put aside and as such is not much of a talking point. Likewise, Information gives the general program version and developer data and Records allows the viewing of log files as discussed in more detail with scanning tests below. The Information area also acts as an alternative method of accessing the multimedia portions of the CD and the virus encyclopaedia.

This leaves as the last section, that devoted to Internet Updates, possible only with a registered product. Registration is prompted for during installation and the reason for registering is explained, although it is also possible from this section. Registering provides access data for the update server. Updates can be performed automatically at certain times, via the scheduler, or, as is recommended, these times can simply be checks to see whether new data is available.

In the latter setting, data will only be downloaded when it is required, but the setting may be overridden temporarily in cases such as corrupted files, where version numbers may remain the same but the requirement is definitely to overwrite existing data.

AVK Scan Tests

AVK was pitted against the *VB* test-sets using the September 2000 WildList. The major problem here turned out to be the format of the results file produced by using the log file viewer. These do offer a good level of information but, unfortunately for *VB* purposes, spread this information over more than one line per detected file. Details of compressed files and their formats are also included, where appropriate. For this reason the results were prepared by directly examining the log files rather than examining the files through the integrated viewer. This problem is not likely to be evident in most real-world situations, however, and otherwise the log files allowed the alteration, after scanning, of information such as day, date and level of information provided, by means of selectable filters.

Since many options exist within the Analysis section of *AVK*, on-demand tests were performed using the default settings as a baseline, while varying both scan type and the types of file to scan. While not covering the whole gamut of variations available this gave a good, if initially surprising, indication of which settings had the most effect upon the time taken to scan and the scan's effectiveness. With time not of the essence the scans were performed directly from CD for detection, while for speed tests the usual *VB* speed test machine was used with files on the hard drive.

The first test involved the removal of heuristics from the scan process, in order to note any changes in detection that this might cause. This produced confusion as soon as results were available, since the report with heuristics disabled declared that more, not fewer, viruses had been detected. Such results inspire paranoia in any reviewer and thus the scans were repeated, looking for signs that this was some artifact of data throughput or due to logging difficulties. The problem was possibly related to the similar oddities seen under the on-access scans, discussed later.

In order that repeat runs could be performed on the same sets, to check for sporadic missed files, the test-sets were scanned on each setting with Delete enabled on a local hard drive. This gave different, but again consistent, results which were more in line with what might be expected and with those results obtained in the recent Comparative – themselves obtained through scanning of data on hard drives. The net impression was that scanning from CD gave repeatable yet different results from those obtained through scanning the same test-set on a disk drive. Of course, this may be a totally false analysis given the data involved, yet the first sets of scans definitely showed odd behaviour for some reason.

The results for local scanning indicated that heuristics made no difference whatsoever – which is probably more of a recommendation of the up-to-date nature of the virus information than the lack of effectiveness of the heuristics.

Only one viral file was missed in the testing in either case – a sample of Avispa – which gave little chance for the heuristics to challenge themselves. It was impossible to run a heuristics-only scan, so this facet cannot really be declared rigorously tested.

Next examined were the settings for the types of file to scan, with Auto Recognition the default compared to Program Files and All Files. User-defined files were not tested as this would reflect upon the user rather than the product. For speed tests the *VB* Clean set was used rather than a scan of viral files – though these times were also noted for the sake of comparison.

On the detection front it soon became clear that changing some of these settings had no great effect upon the results obtained. The Auto Recognition seems to be, currently, identical in its activity to the All Files settings as far as detection goes. This is a sign that the Auto Recognition is doing its job perfectly at the moment, though there is the chance that future developments may change this.

For now, however, the manual's statements that selecting All Files is more likely to cause delays and false positives than have any noticeably good effect, remains true. Altering settings to Program Files did show a noticeable change – due to the omission of .POT, .PPT and extensionless files from the list of extensions scanned, all the *PowerPoint* samples remained undetected.

Since this includes the currently ItW O97/Tristate.B, this setting is definitely not recommended for use. This is one case, however, where *AVK Office* would prove useful, since it does scan *PowerPoint* files constantly and would give an effective second line of defence for the user.

On-access scan testing was performed next and as is the wont of on-access scanners, this showed some variations when compared with the on-demand scans as far as detection was concerned, when the results were analysed.

Also, more disturbingly, this testing gave variations within its own scans of the same type. Of the polymorphic set samples, a selection, differing with each scan, were missed on-access with heuristics enabled. The matter of the missed polymorphics was restricted to scans where heuristics were activated and is possibly the result of time-out issues as are more commonly seen with the larger polymorphics. Since

the big polymorphics in the *VB* test-set are samples of the macro virus W97/Splash.A it was not clear whether this would be the case without a scan of all files on-access. This setting was tried, not only for this reason but also to see whether *AVK* can be forced to scan for macro viruses on-access despite this not being the default choice.

This sort of time-out test would seem likely to be triggered by larger polymorphics, especially the vast and slow-to-scan WM/Splash.A. This was detected fully in 100 samples using a 'no heuristics' on-access scan. The same result with heuristics, however, seemed to rule out an overall time-out fault since, during the tests, the rate of scanning certainly slowed noticeably while processing the larger of the infected files. This problem remained a mystery, though not without a number of leads.

A possible further clue was given in the actions of the on-access monitor when selected during the scan process – this brought up a constantly updated list of infected files, which seemed to freeze or slow initialisation of the main *AVK* application. On occasions where such mid-scan monitoring was initiated, infected files were not detected as viruses and these files were scanned in sequence as a contiguous block. This would suggest that at times of heavy processor usage the *AVK* on-access scanning engine is prone to drop files which are due to be scanned.

Other misses remained consistent and not too frequent – these were the sample of Avispa already missed on-demand, plus two Cruncher samples and the .HTA incarnations of JS/Bubbleboy and JS/Unicle. The default non-compressed files option in the monitor explains the Cruncher misses and the .HTA files may be considered targeted by the *Outlook* component (not tested in this review). When heuristics were not enabled, .HTA samples of JS/Kak were also missed – odd considering that these were not subject to variable detection in the on-demand scans.

AVK Office Detection Tests

AVK Office seeks to protect users from macro viruses. As such, it is disturbing that the *Office* version supported is limited to one particular version – and that is far from the most popular or prone to infection. Tests showed no detection in other *Office* versions, as would be expected. Assuming, however, that *Office 2000* is installed the question still remains – how effective is this protection?

Microsoft Office 2000 was installed and security set to low (noted in the dialogues as *Microsoft's* recommendation when systems are equipped with virus scanning software or no viruses are present). *AVK* was then applied to the machine in question, the *Office 2000* and GUI portions having been selected through a user-defined installation.

As a random choice, WM97/Class.BV was selected for preliminary investigations. The standard *AVK* scans detected this with no problems in infected goat .DOC files and, sure enough, there was an immediate alert when such a file was selected for opening. This is a good result on the face of it, though the sample size was small at that point, but matters were not as they should be.

There was a warning that the document opened contained a virus but nothing was done about this, since the random sample was on CD and could not be disinfected in situ. This does not, however, excuse the fact that the declared infected document could then be saved elsewhere, still infected, or that NORMAL.DOT was infected at the same time.

A further launching of *Word* gave a new warning – that NORMAL.DOT was indeed infected, though this time the offending template was cleaned. An infected file opened after yet another launch of *Word* was cleaned immediately – this one had no infection despite having the same contents as the original but being full access on a local drive.

Curiosity having been piqued, the test was next tried with a read-only file as the infected source, which is an entirely feasible real-world situation. Some companies have, for example, a central database of document templates for standard letters. Once more it was possible to infect NORMAL.DOT and to save an infected file. Since the 'virus removed' and 'virus detected' messages are gone in the twinkling of an eye, this capability to host an infected session of *Word* is not altogether apparent, adding to the potential for disaster.

Given that *Word 97* macro viruses could be detected, it seemed a logical step to determine whether older viruses and those on other applications would also be detected. O97/Tristate.C was chosen for the honour of testing for compatability with other applications, with the *PowerPoint*, *Excel* and *Word* samples being detected upon loading.

Upon testing with other Tristate variants, however, it became clear that detection was far from the perfect score of samples detected in on-access testing. With limited time available for testing, the *Office 2000* component of *AVK* was not tested exactingly, but what was inspected showed definite weaknesses in contrast to its well-implemented on-access scanner.

Internet Update Options

As mentioned, the registration process enables the Internet Update features of *AVK*, which were tested under *Windows 98*. The registration number is required for this process, together with the inputting of a relatively large amount of personal information. Here was one area where the German influence was most pronounced, with address information following a format unfamiliar to most English speaking countries.

This linguistic obstacle proved, however, the most taxing to be found in the update log-in. Anticipating those who have trouble keeping vital information, the process freezes until the user has positively stated that they have noted their log-in password and user name, though these details can be

stored within *AVK* automatically. With registration completed, updating was simplicity itself and with the 10-day hiatus between the pressing of the CD tested and the end of the review writing process, the total update size reached a massive 12 KB in size.

Overheads and Scan Rates

Scanning times for the *AVK* on-demand component were taken for scans with and without heuristic analysis enabled. The results came in at a scan time of 555 seconds for the standard settings and 490 seconds for a scan with the heuristic element deactivated. Given that this represents roughly a 10% additional overhead when performing an on-demand scan with heuristics there seems to be little reason to turn them off.

Varying other aspects of the on-demand scanning process, such as activating All Files mode or deactivating smart decision making as to whether scans are performed, is of little relevance when faced with test-sets where all the files are potentially infected. This was therefore not included in the variations for either on-demand or on-access scanning.

(It is worth mentioning here that *VB* is considering whether to add 'junk' files to the false positive test-sets as the fashion is tending to be to set scanning for All Files – due warning will be given if this is implemented.)

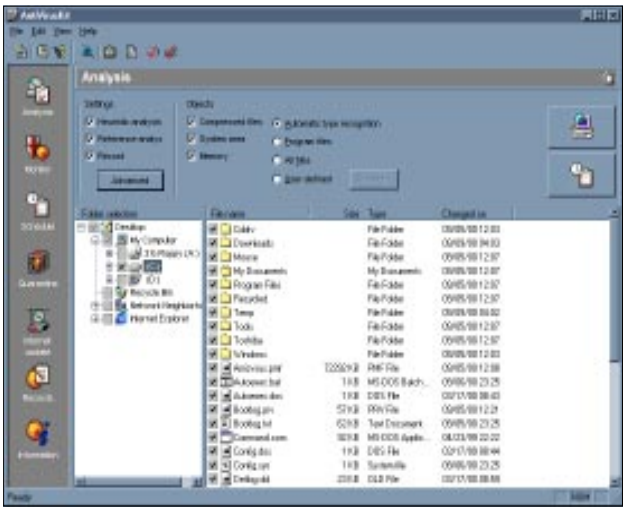
Overhead testing was also performed upon the on-access scanner, using the tried and tested method of a self-timing, automated copying routine. Actual values are omitted as being of no real use, though there were some general trends to be noted outside of the variations inherent in *Windows* file activity. The features included here – standard settings, no heuristics, no scan on write, and use of All Files rather than auto-type recognition – were all compared with a baseline of the scanner deactivated.

The standard setting times were some 60% longer than the baseline with no heuristics, making very little difference to this figure. Removing write checking took the overhead down to around 50%. The biggest reduction was, however, achieved by the change from automatic type detection to All Files – there being only a 40% overhead in this case.

With this in mind, the addition of macros to the scanning list looks all the more appetizing although, as mentioned previously, this does not take into account the scanning of files with no possible viral content.

Conclusion

GDATA's AntiVirusKit is a difficult product to review and even more hard to summarise in a few lines. The presentation is good, the virus detection is good and yet there are still areas where some users will feel unsafe while others can feel secure in their protection. The reasoning behind the decisions which make this the case is entirely logical, so wherein lies the problem?



The reliance upon users having installed *Office 2000* rather than any other *Office* components is an obvious weakness. This is especially so considering that the protection offered by the *Office* plug-in is weaker than the built-in anti-macro virus capabilities which *Office 2000* has to offer.

It is good to see that the on-access scanner is perfectly capable of detecting macro viruses and that this ability to do so is now enabled by default. There are, on the other hand, a number of weird technical glitches associated with *AVK* which allow files to slip through detection on occasion despite detection being perfect in the same files under different circumstances.

Having said all this, the product is capable of very good levels of virus detection and is easy to use and understand. The documentation is clear and thorough and the multimedia snippets add both education and amusement to the package. The boot scan using *Linux* is a fine idea, though not without small but vital flaws, a situation which is reflected in the overall package. If the collection of minor problems and niggles is eliminated by further product developments, it seems likely that *GDATA's AntiVirusKit* will be taking away a VB 100% award some time in the near future.

Technical Details

Product: AntiVirusKit v10.

Developer: *GDATA Software GmbH*, Postfach 10 08 68, 44708 Bochum, Germany.

Price: Standalone version, in German and Polish, soon to be released in English – DM99 (approx US$44). Further prices available through *GDATA*.

Test Environment: Workstations: Three 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disks, CD-ROM and 3.5-inch floppy, all running *Windows NT*. The workstations were rebuilt from image back-ups, and the test-sets were scanned on local hard drives or CD-ROM.
Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/WinNT/200011/test_sets.html. A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.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

***Sophos* is to host a two-day Anti-Virus Workshop on 23 and 24 January 2001** at the organization's training suite in Abingdon, Oxfordshire, UK. For more details about the different courses and training days available, or to reserve your place, contact Daniel Trotman; Tel +44 1235 559933, or email courses@sophos.com.

The UK Security Show 2001, incorporating The IT Security Showcase, is to take place in Hall 2 of the Wembley Arena in London, UK from 14–15 February 2001. The line-up includes interactive product demonstrations and practical installer workshops alongside study-based seminars and debates and more traditional conference-style presentations. For more details about the event visit the Web site http://www.securityshow.com/.

InfoSec 2001, Europe's largest IT security event, is to take place from 24–26 April 2001 in the National Hall, Olympia, London, UK. See the Web site http://www.infosec.co.uk or find out more about the event by emailing infosecurity@reedexpo.co.uk.

iSEC Asia 2001, to be held at the Singapore International Convention and Exhibition Centre from 25–27 April 2001. The conference and exhibition covers IT security topics from anti-virus through encryption to biometrics and digital signatures. For more information and a booking form contact Stella Tan; Tel +65 322 2756 or email stella@aic-asia.com.

InfoSec Paris 2001, the 15th information systems and communications security exhibition and conference, will take place at the CNIT, Paris-La Défense, France from 29–31 May 2001. Companies wishing to participate in the exhibition are encouraged to contact the organisers; Tel +33 0144 537220, or email salons@mci-salons.fr.

Network Associates Inc* (*NAI*) announces the development of *VirusScan Wireless for Mobile Phones, the latest addition to the *Dr Solomon's Wireless* product line. For further information, contact Caroline Kuipers in the UK; Tel +44 1753 217500 or visit the Web site http://www.nai.com/.

***Norman Data Defense Systems* announces that its anti-virus product, *Norman Virus Control* (*NVC*)**, can now be purchased over the Internet in its single user format. For more information, or to place an order, see http://www.norman.com.

***Linux* is coming!** According to *Computer Weekly*, the 1999 Linux Business Expo covered 17,700 square feet. Last month's event alongside Comdex in Las Vegas covered 40,000 square feet – an increase of 226%.

Following an unprecedented reception this year, **LinuxWorld Conference and Expo 2001** is scheduled to take place at the Frankfurt Trade Fair Grounds in Germany from 8–10 November. For more details, see http://www.linuxworldexpo.de/.

San José-based Internet security specialist *Finjan Software* has announced a partnership with Finnish AV company *F-Secure Corporation*. Their joint product, *SurfinShield for F-Secure Policy Manager* is aimed at customers who want to manage multiple security products from a single console. The new product will be available in early December 2000 for *Windows 9x*, *ME*, *NT* and *2000* systems. For further information contact Jukka Kotovirta; Tel +1 358 9 8599 0542 or email Jukka.Kotovirta@F-Secure.com.

***Symantec* has recently implemented a service which offers free on-line security checks for PC users via its Web site.** Features include a guide to computer viruses, a glossary of terms, frequently asked questions and a range of Internet related issues. For more details, see http://www.symantec.com/securitycheck.

'Easing the spectre of your worst nightmare' **London-based *PC MEDICS* claims that, in addition to its Prepay membership service, it will 'alert you to any new virus going round absolutely free.'** The company's Gold and Silver Service membership packages, offering a guaranteed response with support within four and eight hours respectively, start at £249. If you believe the claim that *PC MEDICS'* 'wide-ranging technical knowledge can solve any problem, however big or small', see http://www.pcmedics.co.uk/.

Francesca, Matt and Bernadette would like to wish all VB subscribers a Merry Christmas & Happy New Year