JULY 2003

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

IN THIS ISSUE

CAUSING A STIR

The University of Calgary stirred up some strong feelings when it announced that its new course on computer viruses and malware was to teach its students how to create a virus. Jimmy Kuo ponders whether there can ever be a legitimate reason for creating a virus and Fridrik Skulason responds to the University's statement.

page 2 and page 17

HAVE YOU HEARD?

Have you heard the one about the computer user who used their pet's name as their password? Just like jokes, it seems the old ones and the obvious ones are considered the best when it comes to users selecting their passwords. Martin Overton looks at some of the ways in which malware takes advantage of this propensity for choosing weak passwords.

page 12

OLD REGULAR

Alwil's *avast!* is a product line that has been submitted for testing at *Virus Bulletin* on a regular basis for nearly ten years. This time Matt Ham takes an in-depth look at *avast! 4*.

page 19

ISSN 0956-9979

'It makes me wonder: are they trying to serve their students or their egos?'

Jimmy Kuo
Network Associates Inc.

ALBERTA STRIKES AGAIN

Until a month ago, Alberta's contribution to the computer anti-virus field was a program called Killmonk, created by Tim Martin in 1993. Its purpose was to eradicate the Monkey virus that was created at the University of Edmonton (Alberta). A decade later, the University of Calgary (Alberta) is proposing to prepare its students for entry into the anti-virus industry by teaching them how to write viruses!

The anti-virus industry has responded unanimously that graduates of such a course will find themselves unable to land a job in the industry. What the advocates of this university course have not understood is why the AV industry simply cannot hire anyone labelled as a virus writer. The AV industry has forever been plagued by the comment, 'They write the viruses, don't they?' And as our business is based on trust, we cannot afford to give any credibility to that thought.

The University is proposing to teach its students about viruses by allowing them to create new variants in a 'secure environment' – so-called by virtue of its separation from the real world network, the promise that none of the creations could possibly escape into the real world, and the assurances that none of the course participants would ever do anything bad with the knowledge they gain from this experience.

Let me propose a different 'secure environment'. Start with the definition that a virus is the combination of some binary bits with an environment that results in the recursive replication of those bits (possibly modified). So, rather than use present day viruses in a present day operating system, and then endeavour to 'secure' it, teach the students how to write a new operating system first. A one-of-a-kind operating system. No matter what direction follows, that is a 'secure environment'.

In the early 1990s, working in the kernel of AIX, we rewrote it to create a version capable of load balancing. Networked machines would continually report its load to each other. Any other machine could request, 'I'm your tty. Run this code.' Executable code would replicate onto this other machine. And the result was a load-balancing operating system. How much more value would this offer students than having them twiddle a few bits of an existing virus? How much more secure would this be?

One of the reasons why ex-virus writers are considered unemployable is that their past creations live forever. Even if they don't live as viable attacks in today's environment, they persist in the virus database files, stealing from everyone's disk space, and time. But none of the students' creations in the fabricated environment would ever need to be a part of any virus database. This fact distinguishes why ex-hackers and ex-car thieves can be hired by security firms, but ex-virus writers can not.

It takes two weeks to a month to teach a new virus researcher about the viruses he needs to know and understand. But what we need are people who can attend a *Microsoft* presentation on *Longhorn* and realize the security weaknesses we need to address. Or to work for *Microsoft* and make sure the weaknesses don't exist in the first place. Apache worms, SQL worms, *Windows* viruses, and each brand of macro virus exist in their own unique environments. Even the top researchers need months to crack each new environment that we need to tackle.

Presented with all these considerations, the University of Calgary representatives have turned defensive and determined to push on with their proposed course. It makes one wonder: are they trying to serve their students or their egos? Not only will they be turning out students with the wrong training, who have been told they won't be able to get a job in their intended field, but consider this: on a resume, it will list the University of Calgary. It will not list the courses that were taken. How many other resumes will also say 'University of Calgary?' Did they take the course? I won't know. But I will know they went to the University of Calgary. Think they'll get the job?

[The matter of teaching students to write viruses is explored further in this issue on pages 17 to 19 - Ed.]

NEWS

THE BIG WAIT

The big news in June was the announcement that *Microsoft* had signed a definitive agreement to acquire the intellectual property and technology assets of Romanian anti-virus manufacturer *GeCAD Software Srl*. As might be expected there followed significant excitement in the anti-virus industry, with large amounts of speculation flying in all directions. At this point, however, no details have been revealed other than the fact that *Microsoft* plans to launch an AV product in the future.

Of course, this is not the first time *Microsoft* has ventured into the anti-virus field. The company's toes were left a little scalded after its first dip into the anti-virus arena when, in 1993, it released a re-badged version of a *Central Point* product with MS-DOS version 6. Despite the eventual flop of Microsoft Anti-Virus ('MSAV'), there were similarly excitable reactions when the product first entered the market. Indeed, in *VB*'s review of MS-DOS 6 (see *VB*, May 1993, p.17), Dr. Keith Jackson predicted, 'Many anti-virus vendors are going to be hit very hard by the inclusion of anti-virus features within MS-DOS … Place your bets as to who will be most affected, but I am in little doubt that a vast shake-up is imminent.'

Well, on that occasion *Microsoft* failed to shake more than a snowstorm, but *VB* wishes *Microsoft* better luck this time, and awaits with eager anticipation to see what falls out over the next six to 12 months – interesting times lie ahead.

CANADIAN RETREAT

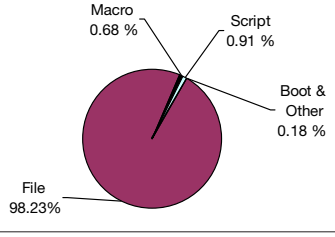
The programme for the *Virus Bulletin* conference can now be found on the *VB* website, complete with abstracts for all papers. This year's programme covers a wide range of subjects, from the detailed analysis of emerging threats and new technologies, to user education, corporate policy and law enforcement. Themed panel discussions at the end of each day will offer the opportunity for some lively debate of pertinent anti-virus issues. This year, there is the opportunity to carry on debating, as the 5th Annual NTBugtraq retreat will be held in the days immediately following VB2003. The Retreat is held at the home of *NTBugtraq* Editor Russ Cooper, approximately 100km north-east of Toronto and promises fishing, bird watching, boating and feasting as well as discussion and debate. The Retreat starts in the evening of Friday 26 September – the closing day of VB2003 – and the nitty gritty begins on 27 September and runs until 29 September. The Retreat is limited to 40 attendees and, judging by the testimonials from past attendees, will be a popular event. More information can be found at http://www.ntbugtraq.com/party.asp and http://www.virusbtn.com/.

Prevalence Table – May 2003

Virus	Type	Incidents	Reports
Win32/Opaserv	File	6313	45.92%
Win32/Klez	File	3221	23.43%
Win32/Dupator	File	1133	8.24%
Win32/Funlove	File	475	3.46%
Win32/Sobig	File	345	2.51%
Win95/Spaces	File	342	2.49%
Win32/Yaha	File	340	2.47%
Win32/Fizzer	File	207	1.51%
Win32/Bugbear	File	192	1.40%
Win32/Magistr	File	166	1.21%
Win32/Gibe	File	152	1.11%
Redlof	Script	103	0.75%
Win32/Lovgate	File	76	0.55%
Win32/Lirva	File	62	0.45%
Win32/Ganda	File	51	0.37%
Win32/Nimda	File	48	0.35%
Win32/SirCam	File	47	0.34%
Win32/Hybris	File	43	0.31%
Win32/BadTrans	File	42	0.31%
Win95/Lorez	File	36	0.26%
Laroux	Macro	30	0.22%
Win32/Kriz	File	26	0.19%
Win32/Elkern	File	23	0.17%
Win32/Braid	File	16	0.12%
Win95/CIH	File	15	0.11%
Others[1]		244	1.77%
Total		13,748	100%

[1]The Prevalence Table includes a total of 244 reports across 80 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

RANDOM NOTES FROM THE UNDERGROUND

Peter Ször, of Symantec Corporation, has compiled his comments on three recent VB articles into a letter to the Editor. Here, the authors' responses are presented after the relevant sections of Peter's letter.

STEMMING THE (OVER)FLOW

I would like to start with Yinrong Huang's 'Stemming the (Over)flow' article in the April 2003 issue of *VB* (see *VB*, April 2003, p.13). The technical feature section 'introduces' the idea of compiler level solutions against buffer overflow attacks, but does not provide obvious references to existing solutions such as StackGuard, ProPolice or Buffer Security Check. I admit that reasonable comparison of existing solutions would go beyond the scope of a single *VB* article and it is the subject of one of my upcoming papers.

The basic idea of the protection is the use of 0xCC opcode insertion in the code flow to raise an exception instead of running the attacker's injected code. The article discusses some of the drawbacks of the solution. However, one of the problems associated with the protection is the raised exception itself. Worms such as CodeRed use corrupted exception handlers that execute using a raised exception. Thus an attacker can use the raised exception of the inserted break point to run the attacker's injected code – which is not desirable.

THE AUTHOR RESPONDS

When utilization of discarded parameter stack space before RET opcode was conceived to protect against a stack overflow such as SQL Slammer, the author did not fully explore other protection mechanisms such as StackGuard, or Buffer Security Check. Therefore, the author regrets that a comparison of these compiling options with utilization of discarded parameter stack space was not included in the original article.

The author agrees with Peter Ször's comment on the potential utilization of exception handling by malicious code to gain the opportunity to execute with the insertion of 0xCC opcode. Therefore, it is probably better to insert 'EB FE' opcodes of an endless loop onto the discarded parameter stack space to prevent the malicious code from going further with 'jmp/call ESP' scheme.

Yinrong Huang, independent reseacher, Canada

MISSION IMPOSSIBLE

Aleksander Czarnowski's 'Mission impossible: WebDAV update' appeared in the May 2003 issue of *VB* (see *VB*, May 2003, p,10). I certainly respect the idea of good practices and therefore I strongly recommend reducing the attack surface by removing non-essential services.

However, the article also discusses the 'interesting educational potential'. Although a good part of the story is covered, including problems with the patch that *Microsoft* provided for one of the vulnerabilities located in Ring 3 (user mode), in NTDLL.DLL (native API), it was not very clear from the article how serious this exploit is.

The exploit was incorrectly known as 'WebDAV' vulnerability, even though WebDAV is only one of the possible ways in which the vulnerabilities may be exploited. In particular, the section of the article entitled 'Why bother' is confusing, since it discusses the StackGuard, ProPolice and Buffer Security Check features to show that this is something that *IIS* would need. I respect this opinion a lot, however this solution would not be applicable to the actual vulnerability in question, since the overflow is external to that of *IIS*'s code base. Thus, recompilation of *IIS* itself would not necessarily resolve all the problems. Therefore, I would also strongly recommend the use of the *Microsoft* patch (after sufficient testing), although issues remain.

THE AUTHOR RESPONDS

I have read Peter Ször's comments carefully, and I believe we have very similar opinions on the subject of IIS. However, some of my thoughts certainly require a little clarification.

First of all, I am not against using patches, hot fixes etc. Currently these play an important role in the process of securing and hardening IT systems – but the situation is very far from being perfect as the process is both costly and time-consuming, while its results still could be doubted. If we use a risk management-based approach then we can limit the number of vulnerabilities that affect our systems by carrying out simple actions such as minimizing the number of services running.

Another important part of such an approach is the deployment of additional protection mechanisms like stack protection. Peter is right that the WebDAV vulnerability would not be stopped even if IIS were protected against buffer overflow attacks. Such a mechanism really only makes sense if deployed on the whole system, not just in one application or library. My point was that such options are already available in the Unix world (on the ProPolice web page you can find information on how to rebuild RedHat and FreeBSD systems, for example), and there are even systems with such protection built in already (Solaris, Immunix, FireBorder OS). It is important to mention that Microsoft is

slowly choosing the same option, which is very good news.

Peter is also right about the misleading vulnerability name and possible attack vectors. As the problem lies inside ntdll.dll library it can be exploited in many ways – even without IIS. The WebDAV name comes from the IIS exploit code, which uses the WebDAV method to trigger buffer overflow. While the name may be misleading it played an important role as many people started to look upon the IIS WebDAV feature as something that could be dangerous, while not necessarily required by their business or technical objectives. It also helps a great deal when explaining to a non-technical audience why IISLockdown Tool offers the option of disabling IIS WebDAV.

To summarize: it is critical to apply hot fixes. The vulnerability discussed in MS03-007 is interesting but also critical – it can be exploited in many different ways, of which IIS is just one example. For an interesting discussion on this vulnerability there are two white papers: 'Analysis of the ntdll.dll WebDAV Exploit', by Eric Hines (see http://www.fatelabs.com/) and 'New Attack Vectors and Vulnerability Dissection of MS03-007', by David Litchfield (which can be found at http://www.ngssoftware.com/). I believe that both these publications explain these comments, and Peter's comments, in more detail.

Aleksander Czarnowski, AVET Information and Network Security

XP, A NEW VIRUS PLAYGROUND?

Mihai Chiriac introduces the reader to 'XP, a new virus playground?' in the June edition of *VB* [see *VB*, June 2003, p.7]. Mihai suggests that 'previously we had seen WinNT.Adonai, 'the world's first virus able to jump to ring 0 … played with the PC speaker'.

Back in 1999, the WinNT.Infis virus had already introduced complete kernel mode, semi on-access file infections in Ring 0, kernel mode which was followed by Win2K.Infis later on (*VB* published information on both [see *VB* November 1999, p.8 and April 2000, p.8]).

It appears that the 'WinXP'.Che virus does not really deserve a new platform category. First, the infection itself does not happen in Kernel mode, but in User mode. Second, it appears the virus works on more than one major Win32 platform. The combination of the two leads us to a simpler platform prefix: 'Win32'. Although I understand that this is somewhat bizarre, it makes things simpler, and it is more likely that a virus name will match between several AV products. (According to VGREP this is hardly the case).

Indeed, the virus author named the virus 'WinXP.Che', probably to make it a little more exciting for the constantly overloaded AV researchers. Did he really manage to create an *XP* virus? Well, first of all, there is nothing in the code that makes the virus *XP*-specific other than some of the hard-coded addresses in its code, as Mihai points out. Second, the virus appears to work on *XP*, but would crash in certain situations, if it ever ran on it. Since the virus manipulates the service descriptor table directly (by patching it), the read-only memory protection on *XP* would lead to a crash on systems with 128 MB physical memory or less (when the protection is active). Thus, in my world, the virus does not really deserve the '*designed for Windows XP*' label just yet, although most systems would have more memory and thus, from the point of view of the virus, this would not really matter.

THE AUTHOR RESPONDS

I absolutely agree with Peter: while VGrepping I've found no fewer than five different names for this virus, ranging from conventional names like 'Che' or 'Cherrat' to funny names like 'Keck' ('Keck' is the name of a very popular cookie in Romania). Even the prefixes differ – some vendors use 'Win2K' or 'WinXP', while others use 'Win32'.

Can we use the 'Win32' prefix in this case? Well, I wouldn't put my money on it. Win32 came in three major implementations: Win32s on Win 3.x, Win32 on the Windows 9x family and the Win32 subsystem on the WinNT family. Since the virus does not run under Win32s or Windows 9x but only on two members of the big NT family, the list of correct prefixes is: 'WinNT' (accurate, but a little misleading to the end user, since the virus does not work on plain NT), 'Win2K' or 'WinXP'.

Another XP-specific part of the virus (and probably the most interesting part), besides the use of hard-coded addresses, is its routine for disabling the WFP, since this routine uses the 'sfc_os.dll' file available only on XP; and, yes, even if the infection is carried out in User mode, the routine for calling syncronous Ring 3 code from Ring 0 sets this virus aside from the normal NT infectors.

Technically speaking, the virus is more advanced than the slightly overrated WinNT.Infis and Win2K.Infis which used hard-coded parameters for int 2E calls (which are not only version-dependent but also service pack-dependent, since they are generated by macros); while the Infis viruses hooked int 2E in the Interrupt Descriptor Table, Che's way of hooking is, again, much more NT-specific (and likely to work with minor modifications on new NT implementations) by modifying the Service Descriptor Table.

Discussions like this can go on for hours; please send other opinions to my inbox: mchiriac@bitdefender.com.

Mihai Chiriac, SOFTWIN, Romania

VIRUS ANALYSIS 1

YOU'VE GOT MORE M(1**)A(D)I(L+K)L

Peter Ferrie
Symantec Security Response, USA

Another day, another exploit is disclosed. A little over two months later, a virus using the exploit is discovered. It seems that some virus writers do read *NTBugtraq*. There is a new member of the W32/Chiton family. The author of the virus calls this one 'W32/JunkHTMaiL', a variation of the name of the virus upon which it is based – W32/Junkmail (see *VB*, November 2002, p.10) – perhaps to draw attention to the self-executing HTML exploit which this virus uses to launch itself from email.

When this virus is started for the first time, it decompresses and drops a standalone executable file that contains only the virus code, using a 'fixed' (taking into account the variable name of the *Windows* directory) filename and directory.

As with the other viruses in the family, this virus is aware of the techniques that are used against viruses that drop files, and will work around all of the countermeasures: if a file exists already, then its read-only attribute (if any) will be removed, and the file will be deleted. If a directory exists instead, then it will be renamed to a random name. The structure of the dropped file is the same as that used by W32/Junkmail. If the standalone copy is not running already, then the virus will run it now. The name of the dropped file is 'Exp*I*orer.exe'. Depending on the font, the uppercase 'i' may resemble a lowercase 'L', making the viral process difficult to identify in the task list.

HOOK, LINE, SINKER

After dropping the standalone copy, the virus alters the Registry in such a way that the virus will be run whenever an application is launched.

The virus alters the 'Shell\Open\Command' keys for the 'com', 'exe', and 'pif' extensions in both the 'LocalMachine' and 'CurrentUser' hives. Both hives are altered because, in *Windows 2000* and *XP*, the Current User values override the Local Machine values. The three extensions are altered because they are all associated with applications. In addition, the change makes removal of the virus more difficult – if the virus is removed before the Registry is restored, then applications will not be able to be launched easily. Fortunately, some improvisation allows for ways around this problem.

If the computer is running *Windows NT/2000/XP*, then the virus will add itself as a service. The virus does not start the service, perhaps because the standalone copy is running already, and *Windows* will perform that action anyway, when the computer is rebooted. If the computer is running *Windows 9x/Me*, the virus will place an undocumented value in an undocumented structure, with the result that the task is not displayed in the task list. This mimics the actions of the recently documented and already very well-known RegisterServiceProcess() API.

IT TAKES TWO TO ARGUE

Whenever the standalone copy is executed, the virus will parse the command line to determine why it is running. The parsing is done in the platform-independent way that is favoured by the virus author – if the computer is running *Windows 9x/Me*, then the virus will use the ANSI APIs to examine characters; if the computer is running *Windows NT/2000/XP*, then the virus will use the Unicode APIs to examine characters. If there are arguments on the command line, then the virus assumes that it was launched via the Registry alteration, and will attempt to execute the application that is named in the first argument.

If there are no arguments on the command-line, then the virus assumes that it has been launched as the standalone copy, and will execute its main code. The main code begins by retrieving the addresses of the APIs that it requires and creating the threads that will allow the virus to perform several actions simultaneously.

The first thread runs once every hour. It will enumerate all drive letters from A: to Z:, looking for fixed and remote drives. If such a drive is found, then the virus will search in all subdirectories for files to infect. Files will be infected if they are *Windows* Portable Executable files for *Intel 386+* CPUs, and are not DLLs.

The method of infection is the same as for some of the other variants in the family – the virus will either append its data to the last section, or insert its data before the relocation table, and alter the entry point to point directly to the virus code. If files do not possess the infection criteria, the suffix of their name is checked against a list of files that might contain email addresses. The virus is interested in files whose suffix is 'asp', 'cfm', 'css', or 'jsp', or contains 'php' or 'htm'. If such a suffix is found, then the file is searched for a 'mailto:' string, and the email address that follows is saved for later.

The second thread runs once every two hours. It will enumerate the network shares and attempt to connect to them. If the connection succeeds, then the virus will search all subdirectories for files to infect.

The third thread also runs once every two hours. It will attempt to connect to random IP addresses. There are two routines for this action, one for ANSI platforms, and one for

Unicode platforms. If the connection succeeds, then the virus will search in all subdirectories for files to infect.

The fourth thread is the one from which the virus gains its name. It runs once every four hours, and will send a single email to the last address that the virus found while searching for files to infect. The virus sends itself using the MIME message format, as described in RFC 1521, and carries an attachment using the MHTML document format, as described in RFC 2557.

While this should present no problems, it appears that a number of developers have overlooked one significant aspect of the formal BNF of, for example, the content of the MIME-Version field. This is that the colon and digits, etc., are separate tokens, and that no white space is required. The formal BNF for the content of the MIME-Version field looks like this:

```
version := "MIME-Version" ":" 1*DIGIT "." 1*DIGIT
```

and a typical MIME-Version declaration looks like this:

```
MIME-Version: 1.0
```

However, when considering the tokens individually, the result is that these are equivalent:

```
MIME-Version          :     1 .            0
MIME-Version:1.0
```

with the obvious problems for those parsers that don't expect white space to appear, or that require at least one space after the colon. The virus attacks the second assumption, by removing the space in all cases.

LAYER UPON LAYER

Microsoft introduced the 'Web Archive' format after the release of *Office 2000*. It is based on the MHTML standard, and resembles a MIME email file, complete with MIME-Version, a Content encoding field, and 'attachments'. An unfortunate consequence of this choice is that such files, when sent as email attachments, can be encoded recursively. Thus, the beginning of such a file might look like this:

```
MIME-Version:1.0
CONTENT-LOCATION:FILE:///.EXE
CONTENT-TRANSFER-ENCODING:BASE64
```

However, after recursive encoding of the type implemented by the virus, it might look like this:

```
 =4DI=6DE=2D=76=65=52s=49=6F=6E:1=28=43H=29.=30
 =63ONT=45=4E=54=2D=4CO=63=61=54=69=4F=4E=3AFIL=65:/
/=2F=2E=45xe
 =63=6Fn=74=45=6E=74-t=72=61=6E=53=46=45=72-
=65NcOd=49=6E=67=3A=62(=4B=7B=
 )a=28=7B+=29s=28=45=29e6(=77Y)4
```

The top level is octet-encoding. It exists to support the sending of characters that are not within the acceptable ASCII range (i.e. foreign and reserved characters), however any character can be encoded with this method. If the octet-encoding is decoded, as will occur when an email program detaches the attachment, it might look like this:

```
MImE-veRsIon:1(CH).0
cONTENT-LOcaTiON:FILe:///.Exe
contEnt-tranSFEr-eNcOdIng:b(K{)a({+)s(E)e6(wY)4
```

Now we see the case inversion and comment insertion that was first demonstrated by W32/Junkmail.

NOT A BUG, BUT A FEATURE

An email sent by the virus will have an attachment called Email.htm. This is a Web Archive file that has a script appended to it. When a file is passed to *Internet Explorer* (*IE*), *IE* will search a large amount of that file for HTML code. This is, according to *Microsoft*, by design.

Thus, an MHTML file with a script appended to it can have that script executed, even though the file does not begin with the '<HTML>' tag. The virus uses a script that requests *IE* to run the file that is located inside the same MHTML file. If the *IE* security settings allow the scripting of ActiveX controls that are not marked as safe, then the file will be launched without prompts, regardless of the zone in which it is executing. *Microsoft* has released a patch (MS03-014) which disables MHTML as a codebase source. The patch is described as applying to *Outlook Express*, however the file that does the work (inetcomm.dll) actually belongs to *Internet Explorer*.

CONCLUSION

The world of security and the world of viruses have become intertwined over the years, and so far we have been fairly lucky that, despite the full disclosure of many exploits, very few have been used in viruses. The successful virus requires knowledge and luck, and while we can't defend against luck, we can see that too much knowledge can be a bad thing.

W32/Chiton variant

Type:	Memory-resident parasitic appender/ inserter, slow mailer.
Infects:	*Windows* Portable Executable files.
Payload:	None.
Removal:	Delete infected files and restore them from backup.

VIRUS ANALYSIS 2

LOV(GATE) IS SWEETER THE SECOND TIME AROUND

Richard T. Fernandez and Paul Vincent M. Sabanal
TrendLabs, Trend Micro Inc., Philippines

February 2003 marked the birth of the first variants of the Lovgate family of Internet worms (see *VB*, April 2003, p.9). In May 2003, three new variants of the worm were released consecutively on the same day – .I, .J and .K. Of these, the .J variant became the most widespread.

The new batch of variants had a greater impact than the earlier set. As new members of the growing family of Lovgate worms, the new variants showcase both a number of new features and some improvements over their predecessors. They are armed with such new features as network folder sharing, anti-virus retaliation and file infection, to name just a few – all of which are reason enough to consider that Lovgate is back … with a vengeance.

WHEN I FALL IN LOV

Once the worm is executed, it ensures that only one copy exists in memory by checking for its mutex named 'I-WORM-Local-Remote-20168 Running!'.

If the event is not found, it drops several copies of itself into the Windows system folder. The dropped files have the following names:

RAVMOND.EXE

WinDriver.EXE

WinGate.EXE

WinEXE.EXE

IEXPLORE.EXE

Kernel66.DLL

Variant .I drops two further copies of itself, named WINHELP.EXE and WINRPC.EXE.

Lovgate contains a backdoor component, which is implemented through the following dropped dynamic link library (DLL) files:

REG678.DLL

Task688.DLL

ILY668.DLL

WIN32VXD.DLL

To achieve file infection, the file DRWTSN16.exe is dropped into the Windows folder.

It is common for malware to make use of the system registry in order to gain execution during startup. Lovgate also uses this method. It creates the following registry entries:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\Run
WinGate initialize =
"C:\WINNT\System32\WinGate.exe -remoteshell"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\Run
Remote Procedure Call Locator =
"RUNDLL32.EXE reg678.dll ondll_reg"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\Run
Program In Windows =
"C:\WINNT\System32\IEXPLORE.EXE"
HKEY_CURRENT_USER\Software\Microsoft\Windows
NT\CurrentVersion\Windows
Run = "RAVMOND.exe"
```

Again, variant .I contains an additional registry entry as a result of its additional dropped file, WinHelp.exe.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\Run
WinHelp = "C:\WINNT\System32\WinHelp.exe"
```

The worm also modifies the registry so that it executes every time a .EXE file is opened. It does this by modifying the value of (Default) to 'winexe.exe %1' in the registry key HKEY_CLASSES_ROOT\exefile\shell\open\command.

On those systems infected with Lovgate.I, the opening of .TXT files are intercepted. The worm changes the value of (Default) to 'winrpc.exe %1' in the registry key HKEY_CLASSES_ROOT\txtfile\shell\ open\command. As a result, this variant of the worm executes every time a .TXT file is opened.

WIN.INI does not escape Lovgate's eyes either. The worm tries to modify the file's run= section as follows:

```
run=%System%\RAVMOND.EXE
```

Note that this technique applies only to *Windows 95*, *98* and *Me* systems, since the run section is not found on *NT*-based systems such as *Windows NT* and *2000*.

Finally, the worm sets itself as a service by creating the following registry keys with each associated value:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
NetMeeting\Remote Desktop (RPC) Sharing
DisplayName = "NetMeeting Remote Desktop (RPC)
Sharing"
ObjectName = "LocalSystem"
ImagePath = "%System%\WinDriver.exe -
start_server"
```

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
Windows\Management Instrumentation Driver
Extension
DisplayName = “Windows Management
Instrumentation Driver Extension”
ObjectName = “LocalSystem”
ImagePath = “Rundll32.exe Task688.dll
ondll_server”
```

MY LETTER OF LOV

Lovgate’s mass mailing routine is its chief spreading mechanism. The first of two mechanisms searches the Windows and My Documents directory for *.ht* files. These two system directories are derived from the following registry entries:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\
CurrentVersion\Explorer\Shell Folders\Winpath
```

```
HKEY_CURRENT_USER\Software\Microsoft\
Windows\CurrentVersion\Explorer\Shell
Folders\Personal
```

For each file that it finds, the worm gathers email addresses by looking for the ‘mailto:’ string in the whole file. Then, it starts sending emails to these addresses, with a copy of itself as an attachment.

To send emails, the worm uses its built-in Simple Mail Transfer Protocol (SMTP) engine by connecting to an SMTP server named smtp.163.com.

The subjects, message bodies and attachments of the emails the worm sends are a random combination of any of the following:

Subject: (any of the following)

```
Reply to this!
Let’s Laugh
Last Update
for you
Great
Help
Attached one Gift for u..
Hi
Hi Dear
See the attachement
```

Message Body: (any of the following)

```
For further assistance, please contact!
Copy of your message, including all the
headers is attached.
This is the last cumulative update.
Tiger Woods had two eagles Friday during his
victory over Stephen Leaney. (AP Photo/Denis
Poroy)
Send reply if you want to be official beta
tester.
This message was created automatically by
mail delivery software (Exim).
It’s the long-awaited film version of the
Broadway hit. Set in the roaring 20’s, this
is the story of Chicago chorus girl Roxie
Hart(Zellweger), who shoots her unfaithful
lover (West).
Adult content!!! Use with parental advisory.
Patrick Ewing will give Knick fans something
to cheer about Friday night.
Send me your comments...
```

Attachment: (any of the following)

```
About_Me.txt.pif
driver.exe
Doom3 Preview!!!.exe
enjoy.exe
YOU_are_FAT!.TXT.pif
Source.exe
Interesting.exe
README.TXT.pif
images.pif
Pics.ZIP.scr
```

LOV GIVES IN RETURN

Lovgate’s rampant spread is largely the result of its ability to trick the user into executing the worm by sending a reply to email messages received in *Microsoft Outlook* or *Microsoft Outlook Express* along with an attached copy of itself.

Since the email reply appears to have come from a trusted source, the original sender of the email may not think that the attachments are malicious, thereby increasing the chances of the malicious file being executed.

The worm traverses the user’s Inbox folder using Messaging Application Programming Interface (MAPI) for incoming email messages as well as other mails located in this folder. The email reply contains the following format:

```
From: <Infected Computer User’s Name>
To: <Original Sender>
Subject: RE: <Original Subject>
Message body:
‘’’<Infected User’s Name>’ wrote:
====
><Original Body>
>
====
<Original Sender’s SMTP account> account auto-
reply:
```

```
If you can keep your head when all about you
Are losing theirs and blaming it on you;
If you can trust yourself when all men doubt
you,
But make allowance for their doubting too;
If you can wait and not be tired by waiting,
Or, being lied about,don't deal in lies,
Or, being hated, don't give way to hating,
And yet don't look too good, nor talk too
wise;
... ... more look to the attachment.
> Get your FREE <Original Sender's SMTP
account> account now! <
```

The email attachment varies and is selected at random from the list shown below:

the hardcore game-.pif

Sex in Office.rm.scr

Deutsch BloodPatch!.exe

s3msong.MP3.pif

Me_nude.AVI.pif

How to Crack all gamez.exe

Macromedia Flash.scr

SETUP.EXE

Shakira.zip.exe

dreamweaver MX (crack).exe

StarWars2 - CloneAttack.rm.scr

Industry Giant II.exe

DSL Modem Uncapper.rar.exe

joke.pif

Britney spears nude.exe.txt.exe

I am For u.doc.exe

SPREAD THE LOV

Like many other widespread worms, Lovgate also propagates through shared network folders. It enumerates network resources and searches for shared network folders with read/write access.

It then drops randomly-named copies of itself with any of the following file names:

Are you looking for Love.doc.exe

autoexec.bat

The world of lovers.txt.exe

How To Hack Websites.exe

Panda Titanium Crack.zip.exe

Mafia Trainer!!!.exe

100 free essays school.pif

AN-YOU-SUCK-IT.txt.pif

Sex_For_You_Life.JPG.pif

CloneCD + crack.exe

Age of empires 2 crack.exe

MoviezChannelsInstaler.exe

Star Wars II Movie Full Downloader.exe

Winrar + crack.exe

SIMS FullDownloader.zip.exe

MSN Password Hacker and Stealer.exe

THE GAME OF LOV

Another social engineering trick that Lovgate uses is the creation of a network shared folder named 'GAME' on the infected computer. The shared folder name may be sufficiently enticing for network users to execute some of the shared files on this folder.

The shared folder is %Windows%\Temp and is shared with Everyone-Full Access. The folder contains several copies of the worm with random file names having file extensions selected from the following:

.dat.exe	.rm.exe
.gif.exe	.txt.exe
.doc.exe	.jpg.exe
.htm.exe	.avi.exe
.mp3.exe	

LEARNING THE WAYS OF LOV

As with the previous variants, these new Lovgate variants also connect to the IPC (Interprocess Communication) share of remote machines.

Each of the new variants also uses semaphores to keep track of the number of threads it has created for remote infection. The difference between these and the earlier variants is that the newer variants use more passwords in their logon attempts.

All three new variants attempt to connect to remote machines as Administrator using the following passwords:

<empty> (no password)	!@#$%^	mypass123
0	!@#$%^&	mypc
000000	!@#$%^&*	mypc123
00000000	123abc	oracle

7	123asd	owner
12	aaa	pass
110	abc	passwd
111	abc123	password
123	abcd	Password
321	abcdef	pc
1234	abcdefg	pw
2002	admin	pw123
2003	Admin	pwd
2600	admin123	root
12345	administrator	secret
54321	alpha	server
111111	asdf	sex
121212	asdfgh	sql
123123	computer	super
123456	database	sybase
654321	enable	temp
666666	god	temp123
888888	godblessyou	test
1234567	guest	test123
11111111	home	win
12345678	Internet	xp
88888888	login	xxx
123456789	Login	yxcv
!@#$	love	zxcv
!@#$%	mypass	

When the logon attempt is successful, Lovgate drops a copy of itself as Net_Services.exe in the remote machine's \Admin$\system32 folder. Then it runs this file as a service named 'Microsoft Network Firewall Services'. It monitors the status of this service constantly, and when it becomes inactive it terminates the remote connection.

TO LOV IS TO LISTEN

Other characteristics inherited from previous variants are the worm's backdoor and password-stealing capabilities. The three identical DLL files, REG678.DLL, Task688.DLL and ILY668.DLL provide the backdoor capability, while the file WIN32VXD.DLL is responsible for the password-stealing functionality.

Lovgate runs the command 'Rundll32.exe Task688.dll ondll_server' to create a service named 'Windows Management Instrumentation Driver Extension'. It also runs the command 'Rundll32.exe ily668.dll ondll_install' to install itself and 'Rundll32.exe ily668.dll ondll_reg' to register itself.

Next it adds an entry named 'Remote Procedure Call Locator = "rundll32.exe reg678.dll ondll_reg"' in the autorun key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.

Once installed, the backdoor listens at TCP ports 1092 and 20168. Port 1092 requires logon authentication while port 20168 does not require any. After the intruder's successful logon, a remote shell will be returned to the intruder's console.

The component 'WIN32VXD.DLL' is used for password-stealing purposes and for notifying the malware author that a system has been infected. This component traverses the windows text of all the processes and child processes to search for possible email addresses, using '@' and '<>' as keywords.

This component also searches for possible usernames and passwords by locating the string 'password' and 'username' from the active processes.

The gathered information is saved temporarily to the files %system%\win32add.sys and %system%\win32pwd.sys before being sent to the email address 'helo_dll@163.com'. The email notification has the subject '333www'.

FATAL LOV ATTRACTION

In an attempt to shut down anti-virus products, Lovgate incorporates an anti-virus retaliation technique. This involves parsing the list of running processes and terminating any process that contains any of a list of specific strings in its process name.

Processes containing any of the following strings are terminated:

rising	RavMon.exe
SkyNet	kill
Symantec	NAV
McAfee	Duba
Gate	KAV
Rfw.exe	KV

LET'S MAKE LOV

The biggest difference between these new Lovgate variants and the previous ones lies in the ability of the new variants to infect other Windows PE executables. The

dropped file named DRWTSN16.exe is responsible for this infection routine.

Upon execution, this component checks first for memory-residency by looking for its mutex. Variants .I and .J search for the presence of the mutex named 'I-WORM-IPC-20168'. Variant .K, on the other hand, looks for the event named 'I-WORM~b-IPC-20168'.

If the worm does not find the event, it searches the local and network drives for PE executables to infect. Files are infected by prepending the virus to the target file and appending a copy of the worm, thereby 'sandwiching' the host program.

This component is present in earlier variants of Lovgate. However, as a result of some bugs in the code, this file infection behaviour did not activate.

CONCLUSION

The new versions of the Lovgate worm possess similar social engineering tricks to those used by the initial variants. Since these types of technique are becoming increasingly popular – and users are gradually becoming more aware of them – such social engineering tricks are starting to become outdated and less effective.

To combat this, the virus author has used some less common techniques in these variants, such as IPC remote infection by significantly increasing the dictionary of common passwords that it uses to crack the administrator account of a remote machine.

Opportunistic as it may seem, these new variants merely exploit what could be considered as the poorest security practice in the world – the use of weak passwords [*for more evidence of this see the following article (pp. 12–16) - Ed*].

W32/Lovgate.I/.J/.K

Type:	Mass-mailer, network worm, backdoor program and file infector.
Infects:	*Windows* Portable Executable (PE) files.
Removal:	Clean all infected files. Delete detected worm copies. Registry entries created by the worm should be deleted. Modified registry values should be returned to original values. Un-share the %Windows%\Temp folder.

FEATURE

YOU ARE THE WEAKEST LINK, GOODBYE! – PASSWORDS, MALWARE AND YOU

Martin Overton
Independent Researcher, UK

With jokes it is often said that the old ones and the obvious ones are the best. How else can you account for the popularity of slapstick and other physical humour and the '*You've Been Framed*' style of TV programme?

According to a number of recent surveys and recent worms, it seems the same is true of computer users' passwords. In other words, the joke is on us, the computer user community!

HAVE YOU HEARD?

Have you heard the one about the user who …

1. Wrote their password on a post it note and stuck it on their screen or 'hid' it under their keyboard?
2. Used their phone number, car number plate, names of family members, pets or their own name?
3. Used 'Password', 'Secret', 'qwerty', '12345' or their user ID as their password?
4. Used the same password repeatedly or re-used a small number of easy-to-remember words/names?
5. Used a repeating character, such as space or z or an x six times?

If you haven't then you have not been involved in computer security for long enough, or worked in a support department – or you are from another planet or universe entirely! Welcome to the 'real' world.

TRUST ME, I'M A SECURITY SPECIALIST

This article will lance the festering boil of computer security; passwords. Just like an embarrassing itch that you don't or won't tell the doctor about, users refuse to seek help or take the medicine that's good for them when it comes to the key to their computer's front door – the humble, but oh-so-important password.

I will also cover some of the recent pieces of malware that have password lists and cracking tools as part of their payload to allow propagation on internal (and external) networks.

The main problem, as demonstrated by Nebiwo (aka Deborm), Mumu (aka SpyBot), Deloder and Lovgate, is that

of weak, easily guessable passwords – or even worse, no password at all – on user accounts and *Windows* shares. Finally, I shall gaze into my crystal ball and try to predict what may be inflicted on us with next from the 'fevered pits' of the malware authors' minds.

DOWN THE WORMHOLE

'You take the red pill and you stay in wonderland and I show you how deep the Wormhole goes …' (borrowed and adapted from *The Matrix*). First, let us investigate in brief just some of the worms that like to carry passwords around with them to use against those who rely on weak or non-existent passwords.

Mumu (aka Bat.SpyBot)

This is a collection of 17 components (including batch files) which spreads via SMB (port 139/tcp) and attempts to gain access to remote systems via the nine passwords held within its code.

The interesting aspect of this piece (or should that be *collection*?) of malware is that, like several other new-ish malware threats, it uses security tools that are more commonly employed by network administrators or other IT support staff. Is this yet another trend? I certainly believe that it is – at the time of writing this article another new variant has been found in the wild.

Nebiwo (aka Deborm)

This piece of malware is not really a password-carrying threat, but it steals credentials from the user logged in on the infected system, and uses them, as well as the following accounts with blank passwords:

- Administrator
- Guest
- Owner

It attempts to use C and C$ shares. This worm has spread quite widely and, like Opaserv, I regularly catch quite a number of infected samples on my SMB-Lure, so it seems to be quite well established in the Internet user community.

Opaserv

On the subject of that large family of worm variants, Frédéric Perriot's analysis of this worm (see *VB*, December 2002, p.6) describes the fact that it carries a distributed DES cracker as part of its body. Could this be a model that other malware authors will follow?

'You take the red pill and you stay in wonderland and I show you how deep the Wormhole goes …'

Borrowed and modified from *The Matrix*

I am still catching thousands of samples of Opaserv variants each month. Over 60,000 samples in total have been captured between October 2002 and the end of May 2003. In the last week alone my SMB-Lure caught four brand new variants of this family, as well as several new malware variants from other families.

Lovgate.K

Lovgate is another well-established family of malware variants (see *VB*, April 2003, p.9 and this issue p.8). Lovgate.K carries a backdoor and mails itself out as well as spreading via SMB, as did other variants of its family.

The .K variant carries a list of 83 passwords in its body for use in a dictionary-style attack on remote hosts found via SMB scanning. However, unlike several other password-carrying malicious programs it, allegedly, uses only the Administrator account.

The problem here is that many default installations of *Windows 2000* and *XP* don't allow you to set/reset the administrator password until after the operating system has been installed.

If, like many companies, you use a static build snapshot, then you may be facing a different problem. Why? Well, unless you have set a 'strong' password on the original system you imaged, you may well have given away the keys to your kingdom!

Furthermore, if you have the same default administrator password on all systems, then you will have a major problem when a brute force attack is successful on just *one* of your systems – effectively, the others are also owned by the malware. Game, set and match to the malware author.

Deloder

Another interesting example of password-carrying malware is W32/Deloder (see *VB*, May 2003, p.5 and http://vil.nai.com/vil/content/v_100127.htm).

Like a growing number of other pieces of malware, Deloder carries other non-malicious programs or components to enable it to spread and/or function. In Deloder's case the components are from VNC (see http://www.uk.research.att.com/vnc/), Cygwin (see http://www.cygwin.com/) and the well-known PSEXEC

tool from SysInternals (see http://www.sysinternals.com). Another variant uses a different remote access tool from that used in the original (VNC). Both of the major variants drop backdoors (Backdoor-ARG) and an IRC bot (IRC-Pitchfork).

Interestingly, Deloder probes not only for C$ and IPC$, but also for ADMIN$, D$, E$ and F$ shares. Basically, the worm looks for the default admin shares that exist normally on the vast majority of *Windows 2000* and *XP* systems – that is, unless your IT department has removed or disabled them.

Deloder carries a list of 86 passwords (the number of passwords varies from one vendor's description to another). As it uses port 445 (Microsoft-ds) to spread, it will only function on *Windows 2000* and *XP*.

As a final and somewhat ironic side note on Deloder, this worm was found happily spreading on the many wireless networks that were set up for the *Infosecurity Europe 2003* show in London in April 2003. It turned out that many of these networks had no security enabled at all, and this was an event *about* security!

Ex-terminate! Ex-terminate!

What's more worrying about this trend is that a number of these worms now carry backdoors, key-loggers and Trojans to disable many AV, personal firewall, IDS and related programs.

What is even more worrying is that some security tools still don't seem to have addressed this problem, and allow themselves to be terminated in this manner.

'84 per cent of computer users consider memorability to be the most important attribute in selecting a password, and 81 per cent of users select a common password where possible.'

Source: 2002 NTA Monitor Password Survey

SURVEYS

The following are some of the results from a number of recent polls and research projects on computer security.

These demonstrate that it is the human element that poses the biggest risk to computer security: no matter how strong your security, it is only as strong as its weakest link – the human behind the keyboard.

NTA 2002

The 2002 NTA Monitor Password Survey (see http://www.nta-monitor.com/fact-sheets/pwd-main.htm) found that 84 per cent of computer users consider memorability to be the most important attribute in selecting a password, and that 81 per cent of users select a common password where possible.

Pentasafe 2002

According to this survey (see http://www.cnn.com/2002/TECH/internet/04/08/passwords.survey/) around 50 per cent of computer users base their passwords on the name of a family member, partner or a pet, while 30 per cent look to a pop idol or sporting hero.

Meanwhile, 25 percent of employees would consider a word as simple as 'Banana' to be a safe and acceptable password – even though it would take a hacker seconds to break into a corporate network using it.

Egg 2003

Below is a list of the most common passwords used as reported in this survey (see http://news.bbc.co.uk/1/hi/sci/tech/2061780.stm):

Child's name	23%
Partner's name	19%
Birthdays	12%
Football team	9%
Celebrities and bands	9%
Favourite places	9%
Own name	8%
Pet's name	8%

Infosecurity 2003 Europe

Here's an interesting quote from another recent survey (source: http://www.securityvoice.co.uk/art.php?art=49): '90% of office workers at Waterloo Station gave away their computer password for a cheap pen, compared with 65% last year.'

The report goes on to say: 'The most common password was "password" (12%) and the most popular category was their own name (16%) followed by their football team (11%) and date of birth (8%).'

Finally, 'Men were slightly more likely to reveal their password with 95% of men and 85% of women giving away their password.'

'… 90% of office workers at Waterloo Station gave away their computer password for a cheap pen, compared with 65% last year.'

From www.securityvoice.co.uk

CRACKING PASSWORDS, GROMIT!

There are a number of methods by which a password for a computer can be obtained or otherwise cracked.

Social engineering

The social engineering approach goes straight to the weakest link in your security: the human behind the keyboard.

Techniques used include:

- Persuading the user to disclose their login credentials (ID and password). We have seen this in the recent PayPal and online banking scam emails, with the perpetrators pretending to be from 'security' or 'the helpdesk' and needing to confirm 'your' password and login ID.
- Key loggers.
- Trojans, including RATs and backdoors.

Guessing

If you know someone quite well – for example a friend or a close work colleague – and they do not follow good password rules, then it is very likely that you would be able to guess their password within a dozen guesses, possibly fewer.

Dictionary attack

This could be as simple as having a list of a few dozen words or many, many thousands of possible passwords and trying each of them.

Brute force

This is the most intensive method; it involves simply trying every possible combination of letters, numbers and in some cases punctuation and other ASCII characters until the correct password is found.

Typically it would start at a, then try aa, then aaa and so on, until either it runs out of combinations to try or finds the right combination, to crack your 'Pa5Sw0rD'.

The main problem with this technique is that it tends to be computationally expensive, and most users would realise that their system was running more slowly than usual.

Sniffing and session hijacking

Both of these methods use tools to perform 'electronic eavesdropping'. Session hijacking tools allow the attacker to steal your credentials as they are sent to the remote system. This can be used to allow the attacker to impersonate you and gain access to the system you were trying to log into or, more often, to modify data in transit between you and the intended recipient.

Packet sniffing tools and protocol decoders are easily used and very effective, especially those that have been written to 'decode' password data. Both sniffing and session hijacking normally require your traffic to pass through a system on the same subnet as the 'sniffer'.

GOOD PASSWORD GUIDE

Below are some basic, but generally sound, guidelines for improving the quality and strength of your passwords.

- Passwords should be a minimum of eight characters.
- Try to include some form of punctuation or one or more digits.
- Use mixed case (include upper case and lower case letters) passwords if possible.
- Choose a phrase or a combination of words, which makes the password easier to remember.
- Do not use a word that can be found in any dictionary (including foreign language dictionaries).
- Do not use a keyboard pattern such as 'qwertyui' or 'oeuidhtn' (look at a Dvorak keyboard).
- Do not repeat any character more than once in a row (e.g. zzzzzzzz).
- Do not create a password consisting entirely of punctuation, digits or letters.
- Do not use things that can be easily determined such as: phone numbers, car registration numbers, names of friends or relatives, your name or employment details, any date. Never use your account name as a password.
- Always use different passwords for different machines.
- Change your password regularly and do not reuse passwords.
- Do not append or prepend a digit or a punctuation mark to a word.

- Do not reverse words.
- Do not replace letters with similar-looking numbers. For instance, the letter i should not blindly be replaced by the digit 1.

Here are a few example passwords that meet many of the above criteria, and none of the pitfalls:

Password: `VB2k3+btORb2`

(VB2k3) = VB2003 + (b) = be (t) = there (OR) = or (b) = be (2) = square.

Password: `TiaS!2Bm1st`

(T) = This (i) = is (a) = a (S)= story (!) = not (2) = to (B) = be (m1st) = missed.

Other useful guides on selecting good passwords can be found at the following:

- http://www.alw.nih.gov/Security/Docs/passwd.html
- http://www.securitystats.com/tools/password.asp
- http://www.securityfocus.com/infocus/1537/.

RISKY BUSINESS

So, how can you attempt to redress the current balance of power that seems to be in the malware authors', and end-users' hands?

Here are a few suggestions – and these are not just for the *Windows* users out there:

- Remove or rename the default Administrator account.
- Disable the Guest account.
- Use the PASSFILT.DLL program on *Windows NT/2K* and *XP*, as this will not allow poor passwords to be used (for instructions see http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/security/passfiltdll.asp).
- On *NIX systems, use a replacement PASSWD binary that will not allow passwords that are weak or that can be guessed easily. Examples include Epasswd (see http://www.nas.nasa.gov/Groups/Security/epasswd/), Npasswd (see http://www.uts.cc.texas.edu/~clyde/npasswd/) or Passwd+ (Passwd+ ftp://ftp.dartmouth.edu/put/security/).

Just like us, passwords need regular breaks in order to be at their most effective – with this in mind, change them regularly, and do not re-use them.

OTHER OPTIONS

Instead of relying on passwords, why not consider the following technologies – some of these effectively replace, or seriously augment password-based systems, thereby making it harder for the malware authors and easier on the end-users without sacrificing the keys to your kingdom:

- biometrics
- smartcards
- tokens.

Yes, there is a cost associated with the use of these, but will not using such technology end up costing you more?

WHAT'S NEXT?

Oh, I hate to crystal ball gaze, both because it can put ideas into the heads of those on the other side and because it often proves to be wide of the mark … but here goes!

I imagine we will see:

- Malware that uses 'brute-force' password cracking methods to defeat 'stronger' passwords, as well as carrying other nasty payloads.
- Malware that uses social engineering to obtain the users' passwords and logon IDs by spoofing a website and email headers, in much the same way as the recent PayPal and bank scams.
- Malware that uses captured SMB packets relating to NT login processes to gain valid account credentials and password hashes. Known as 'Passing the Hash' (see *Hacking Exposed*, second edition pp. 156–159 for more details).

And there are many others …

'No matter how strong your security, it is only as strong as its weakest link – the human behind the keyboard.'

CONCLUSIONS

In the future we will see increasing numbers of new malicious programs that will take advantage of the 'human element'. Social engineering and weak passwords will be the key areas here.

As stated previously, it is the human element that presents the biggest risk to computer security. You can now tell your staff it's official (see http://nl1.vnunet.com/News/1136127 and http://news.com.com/); 'You are the weakest link … in security.' This is something many of us knew already, but were too polite to mention – especially to <*insert your favourite weakest link here*>!

OPINION

TO TEACH, OR NOT TO TEACH?

Continuing the theme started by Jimmy Kuo (see p.2), there has arisen significant debate in recent weeks over the question of writing viruses – is there ever a legitimate reason for creating a virus? The academics at the University of Calgary believe so, and plan to teach their students to write viruses. The anti-virus industry has been unanimous in denouncing the Univeristy's proposals. Here, we present the statement made by the Head of the Computer Science Department at the University of Calgary, and a well articulated response from a prominent member of the AV community.

TO TEACH

As a part of a set of courses on Computer Security the University of Calgary is offering fourth year students – and fourth-year students only – a course on computer viruses and malware. The course will prepare the newest computer professionals with the expertise needed to work in a computing environment which includes more than 80,000 computer viruses and other forms of malware. A critical element of a complete education for the graduating professional computer scientists ***must*** include knowledge about viruses, their nature, and their destruction.

It is time for critics to take their heads out of the sand and work with us to start developing the next generation of computer professional who will be proactive in ***stopping*** computer viruses. The current approach of reacting to the viruses is simply not working. The University of Calgary continues to take a leadership role in this area and this course is another example of the cutting edge research and education undertaken in the Department of Computer Science at the University of Calgary.

Let's be honest: any reasonably intelligent individual can get this information from the internet without having to spend four years at University. There are easier and cheaper ways for them to wreak havoc. It is naïve and dangerous to think that virus writers can be stopped without a better understanding of how they operate.

This course sets viruses within a professional ethical framework, discusses legal factors, and fully considers the environment within which malware exists in the modern computing systems.

The course addresses three primary areas:

Knowledge is critical. Some detractors claim that teaching students about viruses is 'wrong' or 'dangerous' because this kind of software is bad. The simple fact is that viruses and malware exist. It is an undeniable fact of the modern computing environment. We are interested in producing computer professionals who have the expertise necessary to stop computer viruses. Further, a critical element of being able to stop these viruses is to have sufficient knowledge about them to be able to write them. That will come as no surprise to IT professionals who understand that to solve a computer problem it helps to understand what caused the problem.

It is clear that anyone who claims they understand computer viruses well enough to stop them also understands them well enough to write them. Anyone who claims otherwise is simply wrong. This course is not about creating new viruses but about understanding how they function with the ultimate goal of stopping them. A necessary step in ***stopping*** viruses is that the computer professional could also write one so we are using the 'writing' of computer viruses as a teaching method.

Is there another way to teach about stopping viruses without providing adequate knowledge so that the students could write a virus? The answer is simple: No. Anyone who claims they can fight a virus but could not write one is either uninformed or trying to mislead for other reasons. We have to wonder why the anti-virus software companies are so opposed to development of software that could prevent viruses from proliferating.

Protecting the Learning Environment. A valid and critical concern is constraining any viruses studied in the laboratory. The University has taken several steps to ensure any viruses developed or studied are constrained within the laboratory:

Students must be in the fourth year of our program and are only permitted into the program with the consent of the Department of Computer Science.

The laboratory will be housed in a secure laboratory that is locked 24 hours per day 7 days per week. Student access will be monitored and limited to only students taking the course.

No removable media will be taken out of the laboratory once it is brought in so there is no risk of viruses leaving on a floppy or removable hard disk.

No 'wireless access' point will be used within the laboratory so nothing can 'leak' out through the air.

No 'wired' access to the computers in the laboratory will exist. Although the computers in the laboratory will be networked together, it will be impossible for a virus to leave the laboratory as no wired connection will exist to outside computers.

When the course ends – the computers used will be completely cleaned by having all removable media destroyed and all hard disks completely scrubbed down to the BIOS.

We are willing to work with the wider community to ensure the best possible education for our students. At least three groups of people have been contacted, and are willing to work with us, to develop this course:

Anti-virus community: We have been in contact with members of the anti-virus community and they have offered to help us in delivering the course and in developing its curriculum. Most of this community accepts the argument that stopping viruses requires sufficient knowledge to also write a virus so they are willing to work with us.

Ethics training: Philosophers, lawyers, and business professionals will be included in the curriculum so students will have a full professional training in all aspects of computer viruses and malware.

The bottom line: We can pretend that the problem will be solved with old methods, or we can take on the problem to a new level of understanding and action to stop virus. It may not make for a good media story, but it should make sense to anyone who owns a computer.

Ken Barker, Head and Professor, Department of Computer Science, University of Calgary.

NOT TO TEACH

My name is Fridrik Skulason. I have been developing anti-virus technology for 14 years. I was for many years the technical editor of the *Virus Bulletin*, and I am one of the founding members of CARO (Computer Antivirus Research Organization), which includes the leading virus experts from most anti-virus companies.

I read http://www.cpsc.ucalgary.ca/News/virus_course.html with considerable interest and I have a few comments on the points raised there.

'The course will prepare the newest computer professionals with the expertise needed to work in a computing environment which includes more than 80,000 computer viruses and other forms of malware.'

I just wanted to make sure that you are aware of the effects that participation in the course may have on the students' future careers. Most anti-virus companies (including ours) have a policy against hiring former virus writers for anti-virus work. What this means is that in the event that the students actually learn something useful in the course, they will most likely not be able to obtain employment in the anti-virus industry due to their participation in the course, and thus will not actually be able to contribute to solving the virus problem.

'The current approach of reacting to the viruses is simply not working.'

While this is true, it has more to do with flaws in human nature – as long as 97.3% (according to the research by Dr. Vesselin Bontchev) of people do not react in an optimal way to a virus infection, viruses will continue to spread. I fail to see how development of more viruses will help in that regard.

'Some detractors claim that teaching students about viruses is 'wrong'.'

Nobody has made that claim. If you had decided to hold a course on 'Detection and analysis of malicious software', nobody would have objected. You would have received the support of the anti-virus industry and other academics instead of the condemnation you are receiving now. With over 80,000 viruses in circulation, there is plenty to learn from dissecting and analysing those that exist – writing more viruses will simply not produce any new benefits.

'Further, a critical element of being able to stop these viruses is to have sufficient knowledge about them to be able to write them.'

This is not so. First of all, over two thirds of existing viruses are created by modifying existing variants. It does not take much skill to be able to modify virus source code in that way – a reasonably intelligent ten-year-old kid can do that. Is that all the skill you are going to require your students to demonstrate?

There are a few virus writers who have been able to write code of a quality high enough to indicate that they could have been writing 'serious' code (including anti-virus programs), had they decided to – the virus writer who went by the name 'Vecna' is one example that comes to mind – but the bottom line is that the skills required to write anti-virus programs are far, far above those required to write viruses – an important point that you utterly fail to address. Most virus writers are simply not of that caliber … forgetting the 'script kiddies' and those that only modify existing viruses, the remainder write such bad code that (assuming the code shows their true abilities) they would have a hard time getting a real programming job.

'Is there another way to teach about stopping viruses without providing adequate knowledge so that the students could write a virus? The answer is simple: No.'

The knowledge required to write a virus is a small subset of the knowledge required to detect viruses. Real computer virus experts also agree that writing viruses is ethically unacceptable – a position which, sadly, you do not seem to agree with.

'We have to wonder why the anti-virus software companies are so opposed to development of software that could prevent viruses from proliferating.'

Anti-virus companies are not opposed to such development. Anti-virus companies are opposed to anything that appears

to legitimize virus-writing in any way, shape or form. Your university course will produce no real benefits for anti-virus companies or for users. Its only long-term effect will be a black mark on the reputation of the University of Calgary, at least as far as computer security professionals worldwide are concerned. In other words, you will not be trusted in the future.

'*Protecting the Learning Environment.*'

I have a few comments regarding this section. It states that 'No removable media will be taken out of the laboratory.' I hope that this implies an armed guard at the door, doing a full body search of the students as they depart, because anything else would be insufficient. But what about things like printouts of the virus source code? Assuming that the students are really able to create a working virus, I sincerely hope that they will not be able to take home a printout of it, only to type it back in on their home machine. I would very much like to see some assurances in this area.

'*Anti-virus community: We have been in contact with members of the anti-virus community and they have offered to help us in delivering the course and in developing its curriculum.*'

There is also the question: what if a student manages to smuggle a virus out of the lab, and releases it? Does the University's liability insurance cover any potential damage the virus might cause. Members of the anti-virus community, including myself, would have been more than willing to help you develop a course on malware analysis and detection. However, should you persist including the creation of viruses, I expect that all such offers will be withdrawn. No self-respecting anti-virus researcher would want to damage his reputation by being associated with a virus-writing course.

'*Most of this community accepts the argument that stopping viruses requires sufficient knowledge to also write a virus so they are willing to work with us.*'

The vast majority of the anti-virus community condemns the part that involves writing viruses, considering it ethically unacceptable, pointless, and outright stupid. On all mailing lists in the anti-virus community, all real virus researchers have agreed that what you are doing is unacceptable, and simply stupid.

You may be secure in your academic ivory tower, not caring that your course is going to help legitimize virus writing, and will only lead to more viruses being written in the future – more problems in the real world which YOU will be responsible for.

You create a mess, and then we have to clean up after you. Shame on you!

Fridrik Skulason, Frisk Software International

PRODUCT REVIEW

ALWIL avast! 4.0

Matt Ham

Alwil is a well established company from the Czech Republic, whose anti-virus products were introduced as early as 1988, although the company in its present form was not founded until 1991. Despite having distributor deals for various hardware lines, these seem to have taken a back seat in *Alwil*'s development over the years, references to these services being all but non-existent in company literature.

Alwil's *avast!* is a product line that has been submitted for testing at *Virus Bulletin* on a regular basis for as long as I can remember. During this time it has undergone one major redesign and a whole host of minor revamps. Although the minor changes have made little individual impact, the overall changes effected have been large. This has led to the existence of numerous different interfaces for *avast!*, with selections being available for basic and advanced control of the application, as well as the ability to revert to the look and feel of previous versions.

Over time, this incremental design change became unwieldy, and a major version number change seemed due. With the introduction of *avast! 4* this change has been made – though it remains to be seen whether this is merely a cosmetic change or whether the underlying structure of interfaces has been tidied up.

One thing is certain, however – version 4 of the product contains new functionality. This is most clear in the presence of the 'BART' application on the product CD. BART is an acronym for 'Bootable Antivirus Recovery Toolkit' and, in concept, is similar to several other products that have allowed recovery of infected systems from bootable media. The most obvious difference in the *Alwil* implementation is that its developers have worked closely with *Microsoft*, and as a result it should be better able to operate with the *NT*-based *Windows* operating systems which often prove stumbling blocks. A major issue with these bootable media is the matter of updating, so that aspect of BART will be of particular note.

avast! has been reviewed recently as part of a Comparative Review – for detailed results of the product's performance on demand and on access for the *Windows XP* platform, please refer to the June issue of *VB* (see *VB*, June 2003, p.15). The bulk of the testing was performed on *Windows XP Professional* and exceptions to this are noted.

INSTALLATION AND UPDATE

When installing using the CD media *avast!* autoruns, giving the option of Czech, German or English language versions.

Cosmetic differences in the new product are apparent even at this stage, since the old brown colour scheme with embellishments has been replaced with a less cluttered blue version.

The next stage of the installation procedure was the readme file, which includes useful information such as contact details and system requirements. All Windows versions mentioned required a sizeable 50 MB of disk space.

The licence agreement is the next page. This includes the useful permission for any user to install avast! on a workstation, portable and home computer, all in total to be counted as one licence, provided they are not used simultaneously.

Moving on, the target directory is selected next, followed by the installation type and other options. The usual choice of Typical, Minimal and Custom installation options is presented here. The Typical installation provides GUI skins, English language extension, English help and installation of the Professional version. The Custom installation allows any of these options to be removed, and Czech language support may be added. The Minimal installation includes only the English language options. For review purposes the Typical installation was selected.

From here onwards there are a few file transfers and the installation process is more or less complete. The option is presented to have a bootup scheduled scan of the machine, but this was declined.

Just when all seemed complete, however, a new installation process began, for the Mail Protection Wizard. For the purposes of the initial tests this was allowed to install in default mode, since the machine was not email-aware. Automatic account protection is the default option, with it being possible to extend this protection to accounts created in the future. All such protection may also be removed and it is possible to apply protection to accounts on an individual basis. As might be expected, SMTP and POP3 server details are requested at this point, after which the mail protection functionality is installed.

Having completed this two-stage process, a reboot is required in order to finish the installation. An immediate reboot seemed to be the wise option here, since the 'Restart Later' option came with the caveat that it 'may cause system failure'.

After a reboot *avast!* popped up an information box detailing the meaning of the various taskbar icons which are installed. This was a first for me, in that this feature of *XP* was applied usefully. In this case the meaning of the *avast!* icon and its colours are explained. The Virus Recovery Database Generator icon also lurks in the taskbar, this being presumed to be a checksum generator.

Updating of the program is carried out via the iAVS button, which is available at several locations in the GUI. This has slightly different default settings for program and virus database files, in that the program files require the user's agreement in order to be installed. This can be fully automated if required.

There is also the rather less useful option for all updates to be instigated manually, which one hopes will not see much real-world usage. The setup for iAVS was simple on the test machine used – with the option to extract necessary data from *Internet Explorer* making the initial setup a matter of one button click. For those who wish to apply settings manually, the options exist to do so, with authentication also being supported.

Also simple was the update procedure – which took less than a minute for the download of a month's worth of virus definition files. The process produces a few flashing boxes, which could prove irritating to some – and thus are able to be suppressed. The update server was never slow to respond when updates were triggered, making this overall a pleasant system to work with.

One final feature, which was not tested, is push iAVS. In common with most update systems iAVS operates on a pull system – downloading updates when the user application sees fit. With push iAVS enabled the facility is also available for *Alwil* to trigger updates if a particularly vital update is available. The push vs. pull debate has various proponents on both sides, and it is good to see a system which allows both options.

WEB RESOURCES AND DOCUMENTATION

The *Alwil* web presence can certainly be regarded as something of a tangled web, it being very unclear as to which URL should be used. Documentation points the user to http://www.alwil.com/, but http://www.alwil.cz/ – the older URL – contains different content. Various other links direct the bemused surfer to pages on the http://www.asw.cz/ site, which contain different information again. Finally, the link displayed prominently on the GUI is to http://www.avast.com/. All links give English versions of the website(s) by default.

Despite this rather disarrayed web presence, information is not hard to track down. Product details, downloads of documentation, virus information and the like are all available here.

The documentation was examined in electronic form (the manuals being available in PDF format). The file names of the manuals are less than informative – with manuals for Exchange, DOS and Firewalls being opened up before the correct manual was tracked down. However, the contents

were more likely to throw users into a state of dismay, since those on the CD were for the old user interface and the documentation on the website was also found to be for the older product versions. The developers confirmed that this documentation is still in production –though it was also stressed that the program help files were, by and large, identical with the bulk of the manual.

These help files can therefore be considered with this in mind. All in all, the help available is good, although as might be expected, feels more like a hyperlinked manual than a dedicated help resource. There are several areas where popups or embedded information are available outside the help functions, these being very useful in their context. This sort of information becomes scarce, however, where the more complex parts of the functionality are concerned. The net result is that although the help and associated resources are good in content, they seem somewhat disassociated from the program itself.

THE INTERFACE

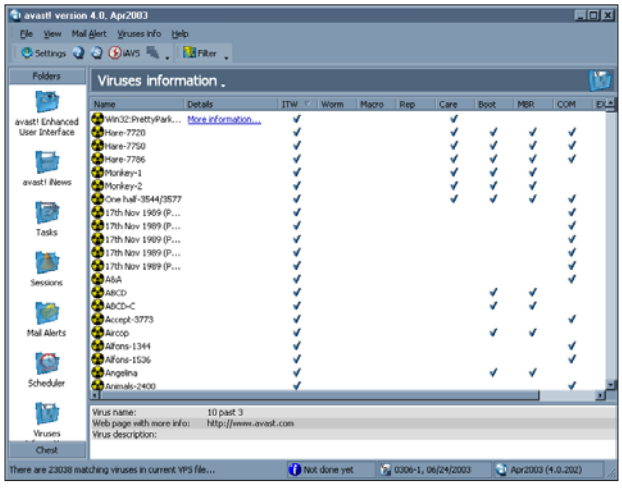
Upon running the *avast!* main program for the first time, the user is prompted for serial number information. Following this, a large popup appears, giving a five-point plan for use of the default, Simple user interface. This provides basic information with hyperlinks to more comprehensive information, and is definitely a good thing where non-technical users might be concerned. One slight irritation, however, is that the information window obscures the interface, and is always on top with relation to the interface.

The Simple version of the interface offers context-sensitive help popups for all functions but two – these being, oddly enough, the scan and pause buttons. All features have fairly self-explanatory and reasonably-sized icons which, together with the help popups and initial information screen, provide better than average assistance to novice users. *Virus Bulletin* readers are, however, a more educated clientele and thus the Enhanced user interface was selected from the menu.

As with the Simple interface, the Enhanced interface opens with what is best described as an introductory page – on which functionality is described, together with links. This is where the similarities end though, since this feature is an integrated part of the GUI and the links lead not to further help but onto the appropriate portion of the interface.

The interface itself is of the familiar large right-hand pane containing the active view, with a left-hand pane for selecting what that view should be – *avast!* terms these different views as folders. There is a separate overall view concerned with the Chest, more of which later.

The folders present in the main view are the general Enhanced Interface information folder described above,



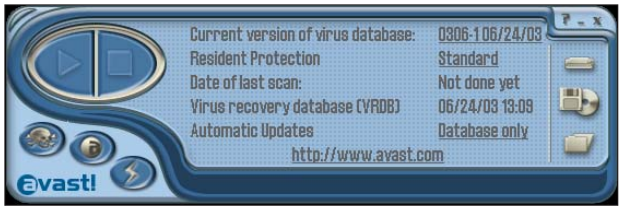
avast! iNews, Tasks, Sessions Mail alerts, Scheduler and Viruses Information. Most of these are fairly standard implementations of their type – Tasks, iNews and the Viruses Information being of sufficient interest to warrant more detailed examination. Sessions holds a list of information output by the various tasks, the details of which are controlled through the Tasks folder.

The iNews folder is home to a selection of informational messages, each of which has an associated longer text. The texts are mostly of the product information variety – describing bug fixes and new features in releases of the software. This standard readme file style of information is augmented with information on the launch of the *avast! 4* product range and *Alwil*'s scheduled presence at tradefairs. Presumably virus alerts can be featured here were *Alwil* to add these, though this does not as yet seem to be the case.

Virus Information is, in fact, the area where such information is to be found. As expected, this consists of a searchable list of the viruses detected by *avast!*. The list may also be sorted on such attributes as whether the virus is worm, macro, etc. and whether it is repairable.

Two less obvious flags upon which sorting can be carried out are one which relates to the 'In The Wild' status of a virus, and another which denotes a virus as requiring particular care when disinfection is attempted. These flags are not entirely accurate or full in their scope. The 'care' flag, for example, is restricted to a collection of boot sector viruses and W32/PrettyPark. While those on this limited list may indeed need special care, the user may experience a false sense of security when dealing with viruses that are not included on this list – for example W32/Navidad which, if simply removed, will render a machine unusable through the registry changes it implements.

Likewise, the 'In The Wild' flag seems to contain a large selection of viruses which were, perhaps, in the wild once,



but most certainly not during this millennium. This flag might be better termed '*has been* in the wild'. However, no glaring omissions were noted during a quick visual scan of those viruses without the flag, which is a good sign.

The main interest in the program lies in the Tasks folder, where configuration and creation of scans is performed. In its usual form this consists of a list of tasks which may be added to, edited or deleted. When editing a task, extra configurations are available if Advanced configuration is selected in a tick box – this was selected for review purposes.

Options available with this configuration are Task Area, Task Type, Results, Sensitivity, Exclusions, Virus, Packers, Report File, Mail Alert and Scheduling. Within these headings there are the usual sets of configurations, although the level of control is very high and can be fine-tuned to an impressive degree.

An example of this is the Task menus, where task types can be not only on demand but on access if required, and targets can be as simple as memory only or as complex as a list of specified directories, files and drives. The default option here is for an on-demand scan using file type recognition – which, oddly, notes itself as being slow. Hopefully this comment will not drive users to use the alternative extension-based scanning.

Related to this option are the Sensitivity options, which offer a choice as to whether files are fully or selectively scanned, according to the likelihood of particular infection in the file. In the same vein, the Packers option allows the definition of which archive types will be examined for infection internally. By default, this covers only self-extracting executables.

The level of information stored in the results database is also highly configurable. Infected files, corrupt files, errors in access, untested files, files where an exclusion list has been applied and files with no problems can all be included individually as entries in this database.

These results are available as entries in the Sessions folder. The Report file configuration option supports the additional production of a standalone report in either plain text or XML format. It is also possible to pass details to a user by means of the Mail Alert options, which covers WinPopup, MAPI and SMTP as methods of transport.

Perhaps the most control can be exercised when viruses have been detected. The options available in this situation are Repair, Move/Rename, Move to Chest, Delete, Stop or a user Interactive action. Admittedly this list is not very different from the majority of programs in the market. The feature that shines out, however, is the inclusion of two logical operators: 'and' and 'if failed'. This makes it simple to create contingency operations.

The comments on task editing so far are concerned with on-demand tasks, which will be the most commonly created and altered in day-to-day product usage. If on-access tasks are edited a rather larger number of parameters may be changed, though these each fall into one of several categories termed Providers.

The Providers are equivalent to the areas selected for scanning when on-demand scans are created. The four major categories are Outlook/Exchange, Internet Mail, Script Blocking and Standard Shield. The last of these is the on-access scanner for the file system and includes behaviour-blocking functionality. The list of individual features in this configuration area is very large, since it covers several different types of Provider under each heading – thus exact details of all options will be skipped.

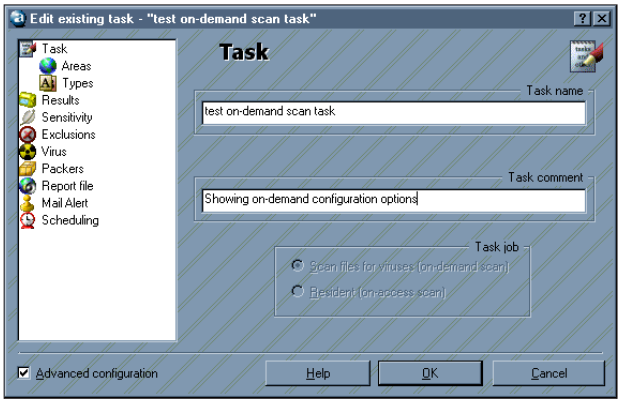
The Chest view is considered sufficiently important to warrant a separate place in the GUI and, when activated, replaces those folders discussed previously. The Chest is similar to a quarantine area, though it has several pieces of functionality which are not standard for the genre.

Primarily the Chest is an isolation area, there being a number of reasons for isolation. First, infected files are stored here where they cannot be executed. Secondly, the user may place files here if they have reason to be suspicious of them. Finally, this is an area in which important system files may be stored as backups – by default, backups were created of kernel32.dll, winsock.dll and wsock32.dll.

Files within the Chest may be permanently deleted, moved, scanned or restored to their original location. They may not, however, be executed or affected by other running applications. It has been noted that the Chest is not checked to assess how large the storage area is becoming. In the case of extensive infections of a machine the Chest can become significant in size and cause system slowdown as it consumes disk space. Although this is an issue for the reviewer, it is hoped that no real-world user will find themselves quite so infested with viral files.

BART

As mentioned previously, the BART CD is, potentially, a very useful tool. Full testing of such a tool could be an epic



task in itself, so it was decided to perform a fairly minimal overview. First, a clean and operational copy of *Windows XP* was booted with the BART CD. This demonstrated its first plus point instantly, in that the CD did boot, but gave the option to bypass it and proceeded with a normal boot – past experiments with this variety of tool have become mildly frustrating with the constant need to extract and replace CDs between boots.

The boot-up process for this situation bore a remarkable resemblance to a standard boot of *Windows XP*. The result is a GUI with five available functions.

The first is a cut-down version of the *avast!* scanner, though as this supports archive decompression and can be targeted, this is perfectly ample for scanning a machine from what amounts to a clean boot of *XP*. It is also possible to create a report file for later consumption, with the same options as the full product, as far as reported objects are concerned.

A disk-checking application, offers surface scans and integrity checks on both static and removable media. Potentially more useful is a registry editor, which searches the machine for *NT*-based registries. This performed exactly as expected. The last two tools are also useful for recovery, being a command line prompt and a text editor. These are, to all intents and purposes, equivalent to Notepad and the command prompt on a *Windows* machine.

Such an overview is interesting, but does not test whether an otherwise inaccessible machine can be booted and disinfected. As a first test a *Windows 98* machine was infected with W32/Navidad.B and the machine 'disinfected' by the removal of the executable. This renders the machine fairly impotent. Using BART it was a simple matter to change the registry so as to restore functionality.

A somewhat harsher test was performed next. A *Windows NT 4* machine was infected with Dodgy, which renders the machine unbootable. Using BART it was a simple matter to determine the identity of the infection, however automatic disinfection was not possible. BART thus succeeds on some counts, but not others.

One notable feature was that all requirements for passwords on the machines were circumvented by the use of BART, making this a useful tool in areas outside that for which it is primarily designed.

It should also be noted that BART is still considered to be in beta, and thus features may be subject to change or improvement. In the case of such upgrades, or even virus definition updates, the question follows as to how a bootable media solution can be upgraded. *Alwil* have clearly considered this from two angles. First, it is possible to use external virus definition files when using BART. This would be useful in the case of a particular update which must be included in scans. A more long-term solution is provided too – *Alwil* provides ISO images of updated versions for download.

CONCLUSION

The common theme throughout this review has been one of pleasant surprise at the degree to which *avast!* has additional and novel features which add functionality. While none of these are truly new (a challenge indeed in such a mature market), the small twists on a standard theme are most welcome.

The BART functionality is, to my knowledge, the first in which a cut-down version of the *Windows* operating system has been licensed directly from *Microsoft* for use in a recovery solution. This does lead to musings as to how long *Alwil*'s contract is secure with *Microsoft*, given that *Microsoft* can no longer be considered a neutral bystander in the anti-virus market.

Overall, *avast! 4* has succeeded in removing a good deal of the obscurity of some options developed during *avast! 3*'s lifetime, making it worthwhile for that reason alone. It is to be hoped that the developers can continue to add interesting new features while retaining the clarity of the current easy-to-use interface.

Technical Details

Test environment: 1.6 GHz Intel Pentium machine with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive running *Windows XP Professional*. 1600+ Athlon XP workstation with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and USB ADSL Internet connection, running *Windows 98 SE*.

Prices: For full pricing details please refer to http://www.avast.com/i_kat_232.html.

Developer: Alwil Software, Prubezna 76, 100 00, Praha 10, Czech Republic; tel +420 274 005 666; fax +420 274 005 888; email sales@avast.com; website http://www.avast.com/.

END NOTES & NEWS

The Black Hat Training and Briefings USA 2003 take place 28–31 July 2003 at the Caesar's Palace hotel, Las Vegas, USA. For full details and registration see http://www.blackhat.com/.

DEFCON 11 will take place 1–3 August 2003 at Alexis Park in Las Vegas, USA. Paid delegates of the Black Hat Briefings USA will receive free admission. See http://www.defcon.org/.

COMDEX Canada 2003 will be held 16–18 September 2003 in Toronto, Canada. See http://www.comdex.com/.

The 13th *Virus Bulletin* International Conference and Exhibition (VB2003) takes place 25–26 September 2003 at the Fairmont Royal York hotel in Toronto, Canada. For information on exhibiting at the event email vb2003@virusbtn.com. Full details, including programme information, abstracts and online registration can be found at http://www.virusbtn.com/conference/.

The 5th NTBugtraq Retreat takes place in the days immediately following the *Virus Bulletin* conference in Ontario, Canada. A welcome event on the evening of 26 September will be followed by the Retreat from 27–29 September 2003. Full details can be found at http://www.ntbugtraq.com/party.asp.

Black Hat Federal 2003 takes place 29 September to 2 October 2003 in Washington D.C. For more information and online registration see http://www.blackhat.com/.

InfowarCon 2003 takes place 30 September to 1 October 2003 in Washington D.C. Military leaders, political forces, academics, and industry members will discuss the concepts of the latest on-going initiatives in the Homeland Security and Critical Infrastructure Protection communities. For details see http://www.infowarcon.com/.

The Fifth International Conference on Information and Communications Security (ICICS2003), is to be held 10–13 October 2003 in Huhehaote City, Inner-Mongolia, China. For full details see http://www.cstnet.net.cn/icics2003/.

The Workshop on Rapid Malcode (WORM) will be held 27 October 2003 in Washington D.C. The workshop aims to bring together ideas, understanding and experience relating to the worm problem from academia, industry and government. See http://pisa.ucsd.edu/worm03/.

COMPSEC 2003 will be held 30–31 October at the Queen Elizabeth II Conference Centre in Westminster, London, UK. This year's conference will include the Compsec 2003 Poster Session, featuring a review of the latest scientific advances in computer security research and development. For full details see http://www.compsec2003.com/.

The European RSA Conference will be held 3–6 November at the Amsterdam RAI International Exhibition and Congress Center, The Netherlands. Further details will be announced in due course at http://www.rsaconference.com/.

The Adaptive and Resilient Computing Security (ARCS) workshop will take place 5–6 November 2003 at the Santa Fe Institute, NM, USA. The workshop will focus on the theme of adaptive defence of information and computing networks. The aim is to stimulate novel approaches to securing the information infrastructure. In particular the workshop will consider long-term developments and research issues relating to the defence of information networks. The deadline for paper submissions is 1 August 2003. For full details see http://discuss.santafe.edu/bnadaptive/.

AVAR 2003 will be held on 6 and 7 November 2003. This year's AVAR (Association of anti Virus Asia Researchers) conference will be held in Sydney, Australia. More details will be announced in due course at http://www.aavar.org/.

COMDEX Fall 2003 takes place 15–20 November 2003 in Las Vegas, USA. See http://www.comdex.com/.

***Eset Software* has announced the release of *NOD32 version 2.0*,** *GFI* has launched *GFI MailSecurity for Exchange/SMTP 8*, while *Norman ASA* has released *Norman Virus Control v.5.6*. For full details see http://www.nod32.com/, http://www.gfi.com/ and http://www.norman.com/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery: £195 (US$310)

Editorial enquiries, subscription enquiries, orders and payments:

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

Tel: +44 (0)1235 555139 Fax: +44 (0)1235 531889
Email: editorial@virusbtn.com www.virusbtn.com