NOVEMBER 2005

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

ISSN 0956-9979

IN THIS ISSUE

WHEN IRISH EYES ARE SMILING

A busy conference schedule, combined with the famous warmth and hospitality of the Irish and a drop or two of the local 'water' were a recipe for success for VB2005 in Dublin last month.

page 14

Virus Bulletin thanks the sponsors of VB2005:

LANGUAGE LOGGING

Using components of *Windows* multilingual support, it is possible to create a file that will capture keystrokes on a target system while using the OS to protect that file from removal or deletion. Masaki Suenaga explains how an IME could be used as a keylogger.

page 6

THE TROUBLE WITH WINRAR

Andreas Marx reports on his extensive false positive testing of anti-virus software.

page 11

vbSpam supplement

This month: anti-spam news and events and John Graham-Cumming looks at measuring and marketing spam filter accuracy.

'It adds insult to injury when the major media outlets misrepresent the facts.'

Gabrielle Dowling
Independent author, USA

IS THE BOOT ON THE OTHER FOOT?

Talk about irony. When I finally managed to log into my email at the conclusion of VB2005, I found folks on AVIEWS were hotly discussing some of the media reports coming out of the conference. In particular, I was struck by a report written for *Silicon.com*, entitled 'Security firms put the boot into the media'. It read: 'At this week's *Virus Bulletin* 2005 conference in Dublin, a panel session featuring representatives from *IBM*, *McAfee* and *Symantec* turned nasty for the assembled press with vendors airing grievances about what they consider to be "a layer of incompetence" in media companies.'

This seemed to allude to the panel I had chaired on the informational problems facing anti-virus administrators. Apparently, I'd hit a sore spot when I threw the following question to the panel (verbatim): 'Do news reports of virus outbreaks typically misrepresent the facts? If so, what are the repercussions of that misinformation?'

It was not an accusatory question, but a pragmatic one, and I was surprised by how it came to take over the panel and audience discourse. It was more surprising still to see how it came to be reported, since I don't think things ever approached the tenor of 'ugly', and in fact the vendors on the panel were generally supportive of media efforts, accurate or not. (Notably, the reporter omitted the voices of those of us on the systems administrators' side, which was equal in its representation on the panel.)

So, putting aside the reporting of the subject, I'd like to touch again on why accurate reporting is an issue for those of us on the administration side.

In the context of the informational problems that face anti-malware administrators, media misreporting is not the greatest challenge we face. And when incidents occur, we certainly don't rely on such reports as primary sources of information. But, in the fog of war, when we are deluged with information – very little of it good, some of it outright wrong, usually with the most critical details missing – and we are trying to process it all as fast as possible, it adds insult to injury when the major media outlets misrepresent the facts. Worse, it adds to the administrative load when we're distracted by queries from end users and the boardroom based on misinformation: it's the last thing we need in the midst of an event (and an 'event' does not necessarily mean that we have an actual or potential problem on our network, but rather that there is an outbreak of something significant against which we need to check the adequacy of our defences and incursion responses).

I think there's another reason this subject stings those in the trenches. As the security field seems increasingly known for folks filled with bravado who like to drop allusions to 'the coming superworm' and dilettantes with little experience writing books that simplistically liken worms to warheads, I am constantly struck by the lack of such swagger in the anti-virus community. They are smart people who have been in the trenches and know better.

Some of the banter that arose at *VB* was finger-pointing by the media, saying they were merely responding to press releases issued by the vendors. That seems wrong, on two fronts. First, fact-checking is part of Reporting 101, and reporters should be aware that press releases are a marketing tool. Second, and more importantly, most if not all vendors have dropped that habit – some actually advise administrators that they are issuing a particular alert simply because of media attention, not because it is being seen broadly in the wild. I love that trend!

Reporters certainly face the same informational challenges as those of us fighting malware, and that's one of the problems that AVIEN/AVIEWS help to address by providing a platform for an experience-rich, marketing-poor exchange of information. But I would say this, and it applies broadly: if your fundamental information is incomplete or not well understood, refrain from extrapolating from it or you'll wind up wildly off the mark.

NEWS

MICROSOFT ASSISTS NIGERIA IN FIGHT AGAINST HIGH-TECH CRIME

The Nigerian government has signed an agreement with *Microsoft* under which the two organizations will collaborate in fighting high-tech crime originating in the country.

Under the agreement, *Microsoft* will provide information, assistance and training to the Nigerian government's Economic and Financial Crimes Commission (EFCC), which was set up in 2002 to address the problem of cybercrime in the country. One of the areas *Microsoft* expects to focus on is botnets – for example teaching officials how to extract useful information from compromised PCs, how to monitor computer networks to detect such attacks and how to identify the perpetrators.

Mallam Nuhu Ribadu, executive chairman of the EFCC said: 'Our economy has lost hundreds of millions of dollars in foreign investment because our credibility and the trust of the international community have been affected.' He added: 'This agreement will be of great benefit to us. It will put us in the proper direction in fighting cybercrime. It will help us to improve our understanding of the technologies involved as well as give us new investigative skills to go after the criminals.'

ERRATA – WINDOWS 2003 SERVER COMPARATIVE REVIEW

Regrettably, there were three errors in the latest Windows 2003 Advanced Server comparative review (see *VB*, October 2005, p.12). In alphabetical order these were:

CAT Quick Heal: Initially this was flagged as having missed a sample of W32/Nimda.A in the .EML format. Subsequent tests revealed that the infected contents of the .EML file were removed, though the file itself remained. This must therefore be considered a detection and a VB 100% award is due to *Quick Heal*.

MWI VirusChaser: Due to an administrative error the tests for this product were omitted from the initial review. The product gained a VB 100% award when tested, with full detection in the wild and no false positives.

Sophos Anti-Virus: The product submitted by *Sophos* was that which was available to the public on the company's website at the time of the review submission. Due to miscommunication, however, the versions downloaded for testing, were an incompatible combination of base scanning engine and virus database updates. Re-testing with the correct combination resulted in full In the Wild detection and a VB 100% award for the product.

Prevalence Table – September 2005

Virus	Type	Incidents	Reports
Win32/Netsky	File	12,228	51.36%
Win32/Bagle	File	4,522	18.99%
Win32/Zafi	File	3,048	12.80%
Win32/Mytob	File	1,023	4.30%
Win32/Mydoom	File	418	1.76%
Win32/Lovgate	File	242	1.02%
Win32/Klez	File	211	0.89%
Win32/Funlove	File	185	0.78%
Win32/Bagz	File	177	0.74%
Win32/Dumaru	File	150	0.63%
Win32/Bugbear	File	140	0.59%
Win32/Pate	File	121	0.51%
Win32/Mabutu	File	101	0.42%
Win32/Valla	File	98	0.41%
Win32/Fizzer	File	94	0.39%
Win32/Mimail	File	87	0.37%
Win32/Reatle	File	86	0.36%
Win32/Swen	File	83	0.35%
Win32/Mylife	File	77	0.32%
Win32/Bobax	File	73	0.31%
Win32/Mota	File	67	0.28%
Redlof	Script	58	0.24%
Win32/Yaha	File	44	0.18%
Win32/Agobot	File	43	0.18%
Win32/Zotob	File	31	0.13%
Win32/Randex	File	26	0.11%
Win32/MyWife	File	25	0.10%
Win95/Spaces	File	23	0.10%
Win32/Gael	File	22	0.09%
Win32/Maslan	File	21	0.09%
Win32/Nimda	File	18	0.08%
Win32/Sobig	File	18	0.08%
Others[1]		250	1.05%
Total		23,810	100%

[1]The Prevalence Table includes a total of 250 reports across 56 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

CRISS-CROSS

Peter Ferrie
Symantec Security Response, USA

Cross-infector viruses demonstrate the flexibility of certain file formats. While some of these viruses have clearly been written to maximise their replication potential (e.g. {W32/Linux}/Peelf, which infected 32-bit *Windows* and *Linux* files, or the member of the W32/Chiton family that infected both 32-bit and 64-bit *Windows* files), most seem to have been written simply to show that it can be done. With the release of issue 6 of the *RRLF* zine in July (published on http://www.rrlf-zine.de.vu/), we received three new cross-infectors, each for a different set of file formats.

MONAD... NO MAD

The first of these viruses is one of a set that captured the media's attention. The virus (which we did not name) is part of a set of viruses for the forthcoming *Microsoft Shell*, or *MSH*, also known as 'Monad'. (Interestingly, one use of the word monad is to signify 'the One' – perhaps the developers at *Microsoft* thought they wouldn't be taken seriously if they called it 'Neo'.)

Originally, *MSH* was expected to ship with the forthcoming *Windows Vista*, but it has since been dropped from the initial feature set – which makes rather a non-event of any viruses written for it.

The virus in question attempted to infect .BAT, .MSH and .CMD files. However, due to a bug which should have been obvious during testing, the .BAT and .CMD forms do not prepend the virus code to the target file, as the virus author would like. Instead, the virus code replaces the target file entirely.

The bug is caused by the virus using 'copy <a>+<b> <b>'. Since <b> is not read before the copy begins, <a> replaces it entirely. To prepend via a copy requires two operations: one to copy the combination to another file, the other to copy that file over the original.

The .MSH code does work as intended and correctly infects all three file types. However, none of the replication types will infect a file whose read-only attribute is set.

Since .BAT and .CMD files differ only in their extension, the virus is really only a two-platform cross-infector. The .BAT/.CMD form of the virus is able to work by relying on the fact that the *Windows* command interpreter will consider lines that are not valid batch commands to be references to external files. Since (presumably) those files do not exist, the *Windows* command interpreter will display error messages instead, but it will then continue to interpret the file.

YOU HAVE NO NEW MESSAGES

The virus attempts to hide its activities by switching off message printing, but some messages (such as those produced by the copy operations) cannot be suppressed, except on DOS, *Windows 9x* and *Windows ME*. It seems that the virus author tried to hide those too, by clearing the screen, but the command to clear the screen appears before the copy operations, so the messages remain on the screen after the replication has completed.

The .MSH part of the virus is able to work by relying on the fact that, as in JScript, an end of line character is not considered to be a delimiter if it appears between the tokens of a statement. Thus a statement can span several lines without causing an error. This is in contrast to VBScript, for example, where each statement must appear entirely on a single line (although multiple statements can appear on the same line, delimited by the ':' character). The several lines in this case form the .BAT/.CMD replication code, after which comes the .MSH replication code.

SEEK AND YE SHALL FIND

The replication from .BAT and .CMD files is achieved by extracting those lines that contain a keyword (the name that the virus author gave it), which appears in every line of the code. These lines are placed into another file, which is then supposed to be prepended to the target files, but as described above, that part simply overwrites the file instead.

In addition, a line of .BAT code that is never executed would have caused some unexpected behaviour if it had been executed. The most obvious effect would be that during subsequent replications, the screen would no longer be cleared at all.

The replication from .MSH files is achieved by searching for files whose extension matches any of the three target extensions, then finding the last of those files which begin with the keyword. Having found such a file, the virus searches again for files whose extension matches any of the three target extensions. The virus then prepends its code to any file that is not already infected, by copying a fixed number of lines of code.

VBJSCRIPT

The second and third new cross-infector viruses are written by a different author. The second, {VBS/JS}/Cada, infects

both VBScript and JScript files by appending the virus code to the target file. It is able to work by relying firstly on the fact that in VBScript the 'rem' command causes the rest of the line to be ignored, while in JScript 'rem' is considered to be an acceptable name for a variable, so the virus assigns a value to it. After that, the entire JScript virus appears on a single line.

Secondly, the virus relies on the fact that in JScript, the '/*' and '*/' symbols constitute a pair that bound a multi-line comment. Thus, at the end of the JScript code, the '/*' symbol appears to begin the comment, followed by the VBScript code on the next line.

Finally, the virus relies on the 'rem' command to cause the rest of the line to be ignored by VBScript, followed immediately by the '*/' symbol to end the JScript comment.

In between, the code searches for all JS and VBS files that can be found in the current directory, and infects any files that are not infected already. The infection marker is the string 'rem=1'. However, since the virus performs no tokenisation of its own, it will consider a file to be infected even if it contains that string as part of a longer string (e.g. 'members_of_harem=1').

OPEN OFFICE

The last of the viruses is a set of four variants that form the {O97M/VBS/JS}/Macar family. They infect the *Microsoft Office* applications *Word*, *Excel*, *PowerPoint*, *Access*, *Project* and *Visio*. The .B and .C variants also infect VBScript files. The .D variant infects JScript files instead of VBScript files. The most interesting thing about the .A and .B variants of this virus is that, unlike typical cross-infectors, which execute different code depending on the file type, this code is exactly the same for all of the *Office* applications and for VBScript.

Infected *Office* documents for *Word*, *Excel*, *Project* and *Visio* execute their macro code automatically, via an auto-macro. Infected *PowerPoint* and *Access* documents require some user interaction to execute: the *PowerPoint* macro runs when the user clicks on a slide during a slideshow (which is made possible by the presence of a transparent AutoShape, which covers the entire slide), and the *Access* macro runs whenever a user opens the first form in the database.

The replication method for .A and .B is unusual – the virus exports its code to a file, reads back that file, removes everything but the virus code (*Office* applications add additional text when exporting macros), then adds what remains to other files. However, this method avoids the blank-line insertion problem that some macro viruses encounter.

TRUST ME

The Visual Basic Object Model was extended in *Office 2000* to prevent macros from accessing themselves, which means that a virus can no longer export its code to a file, then import the file to other documents. The .C and .D variants of Macar work around this limitation by creating a macro that carries the whole virus code and writes it to disk. The dropped code is then executed by the macro. Since the external file performs the replication, no reference to the macro code itself is required. This is also an effective anti-heuristic device, at least from the perspective of the macro platform, since the macro does not replicate, although the external file that runs is highly suspicious.

The script begins by setting the VBA security settings for the chosen application to the lowest level. It knows how to adjust the settings for all *Microsoft Office* versions, from *Office 97* up to the as-yet-unreleased *Office '12'* (the virus author guessed the names of the registry values correctly). The virus works in all the pre-release version of the *Office '12'* applications, with the exception of *PowerPoint*.

GET TO THE POINT

One of the more surprising behaviours, from the user's point of view, is the occasional visible launching of *PowerPoint*. Macar uses *OLE Automation* to infect documents, which is done by running the application, and scripting the actions to take. Thus, whenever Macar decides to infect a *PowerPoint* document, *PowerPoint* is launched (if it was not running already). The reason the launching of the program is visible is because *PowerPoint* does not allow its main window to be hidden, unlike the other target *Office* applications. *Visio* also behaves in an unusual manner – the splash screen is visible, but the main window is not.

Another surprising behaviour is that of *Project* which, once it appears in the Task List, never goes away. This occurs when Macar decides to infect *Project* documents, because *Project*, along with *PowerPoint*, does not allow multiple copies of itself to be running at the same time.

CONCLUSION

Cross-infectors present some interesting technical hurdles for virus writers and, to a degree, for anti-virus writers too (since the target platforms must be identified and replication on those platforms is required for correct naming – the appearance of the sample can also differ there in significant ways, which can affect the detection).

While virus writers' time is best spent doing entirely non-viral things, whatever slows them down is the next best thing.

FEATURE 1

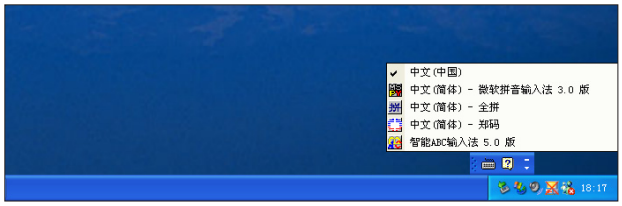
IME AS A POSSIBLE KEYLOGGER

Masaki Suenaga
Symantec Security Response, Japan

The aim of this article is to outline two potential methods for using an IME as a keylogger: by hacking a genuine IME for the Far East versions of *Windows* and by creating a fake IME for other versions of *Windows*. Using components of *Windows* multilingual support, it is possible to create a file that will capture keystrokes on a target system while using the OS to protect that file from removal or deletion.

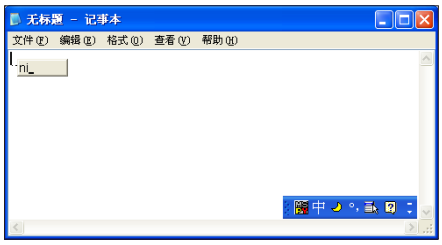
To begin, I'll explain what an IME, or input method editor is. The Chinese, Japanese and Korean writing systems use thousands of characters: Hanzi (Chinese characters) in Chinese; Kanji (Chinese characters), Hiragana and Katakana in Japanese; Hangeul and Hanja (Chinese characters) in Korean. To represent these characters, each of these languages has its own multi-byte character code sets. On ASCII code-based *Windows* such as *Windows 95*, the double byte character set or DBCS is used, where each two-byte sequence represents one character. While DBCS is no longer commonly used, it is still used on *Windows XP* if a program does not call Unicode APIs. Starting with *Windows 2000*, *Microsoft*'s desktop operating systems have primarily used Unicode for cross-compatibility and ease of use.

If a keyboard had thousands of keys, as was once the case with mechanical typewriters, there would be no need to convert multiple keystrokes to a single character. However, most modern keyboards have only around 100 keys. Therefore, we need something to convert keystrokes to characters before being used in an application. This kind of software is called a front-end processor or FEP, and IME is the standard name for FEPs used in *Windows* environments.

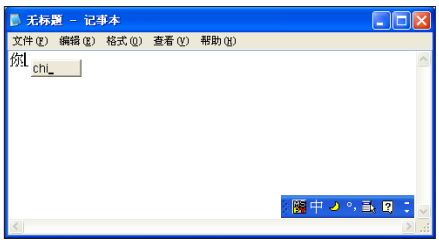


The image above shows some common IME options when the keyboard icon is clicked. The pop-up list shows all the available IMEs or keyboard layouts for a given language.

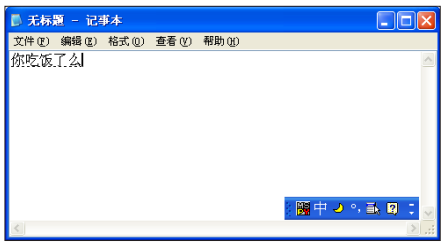
The following pictures illustrate how a user inputs Chinese characters in *Notepad*. The IME status bar is shown in the bottom right-hand corner of the *Notepad* window here, but it can be placed anywhere, and generally is shown either in the bottom right-hand corner of the screen or as part of the Taskbar.



In the first screenshot the user has typed 'ni' while IME is ON. The string 'ni' here is called the 'composition string' in the interface.



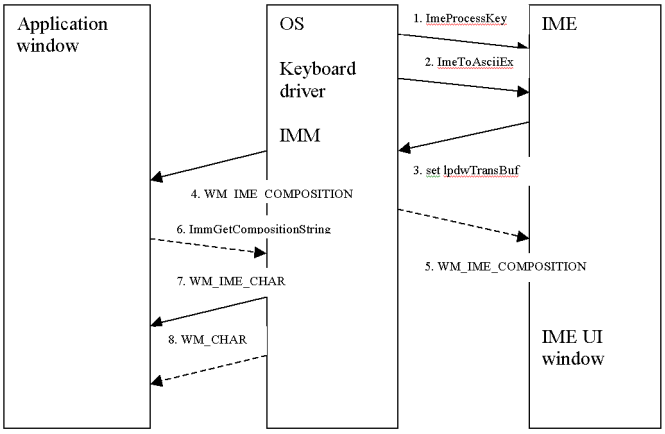
Next, the user has typed 'nichi'. You can see that 'ni' has disappeared from the little grey box, and the character which is most likely to represent 'ni' is underlined with dashes. Both of these strings are called composition strings because they may not match the final text used in the application.



The third image shows the screen when the user has finished typing the phrase 'nichifanleme' and has pressed the Enter key. There is no little grey box, but the five Chinese characters are underlined with dashes. These characters, 10 bytes in DBCS and five words in Unicode, have not been passed to the application yet. At this stage it is still considered a composition string.

Finally, the user has pressed Enter a second time to confirm the characters in the previous string. This string is called the 'result string', since the user has chosen not to alter any of the IME-selected characters for the keystrokes they typed. Each character is sent to the application window with the WM_IME_CHAR window message.

If the window procedure does not process this message, it will receive a WM_CHAR message. So most applications don't need to be aware of IME. (Some applications, such as *Microsoft Word* and *Excel* are fully aware of IME and display composition strings by themselves.) These



complicated tasks are performed by the IME and IMM (input method manager), and an IMM is included in every language version of *Windows* with multilingual support (*2000* and later).

The process diagram above illustrates how the text input process occurs.

1: When IMM receives a key stroke from the keyboard driver, it sends the key stroke to the IME through the ImeProcessKey entry to ask the IME if the key stroke should be processed by the IME. If the IME returns zero, the key stroke will be processed by the OS and passed to the application as WM_KEYDOWN and WM_KEYUP, then further processed to WM_CHAR or WM_COMMAND, and so on. The IME will not receive the key through ImeToAsciiEx.

2: If the IME returns a non-zero value in ImeProcessKey, the IMM sends the character to the IME again.

3: The IME receives the lpdwTransBuf parameter, which will be set by the IME when the process returns from the IME to the IMM. The lpdwTransBuf parameter contains information about window messages to be sent to the application. The IME also receives the hIMC parameter, which contains composition strings, such as the composition string itself, the result string, and any reading information or clause information, depending on the language. The IME modifies the content of hIMC as it processes characters.

4: Any time the IMM receives lpdwTransBuf back from the IME, the IMM checks the buffer to see if it contains a message list. Typically it contains the WM_IME_COMPOSITION message, which should be sent whenever the composition string changes. The IMM sends these messages in the buffer to the application window.

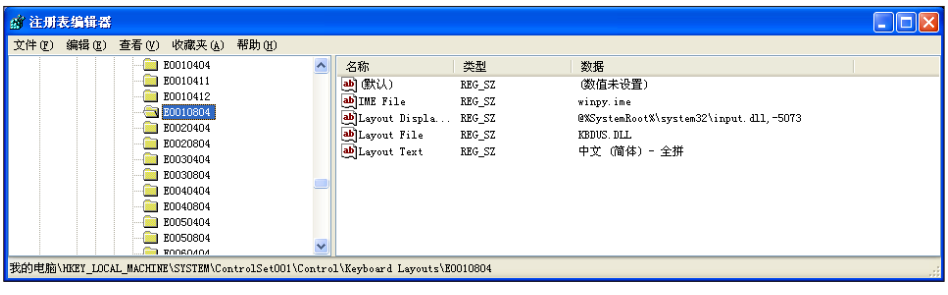
5: If the application is not IME-aware, it will not process the WM_IME_COMPOSITION message and thus the user will not see the text within the application. In this case, the message is relayed to the corresponding IME UI window, which is always created if an IME is activated. An IME UI window will show the composition string as it is typed.

6: If the application is IME-aware, it will process the WM_IME_COMPOSITION message. If there is a need to get the contents of composition strings, it calls the ImmGetCompositionString API in IMM32.DLL. The WM_IME_COMPOSITION message can also notify that the string is determined and the result string has been generated. If the application gets the determined string directly from IMM and pastes the string into its document, it should not call DefWindowProc on the WM_IME_CHAR message, because further processing will generate the same character twice.

7: If the application is not IME-aware, it will receive the WM_IME_CHAR message. If the application uses the GetMessageW API (along with DispatchMessageW), it will get one Unicode character in a WM_IME_CHAR. If the GetMessageA API (along with DispatchMessageA) is used, the application receives one DBCS character in the message. If the application does not call DefWindowProc on WM_IME_CHAR, it will not receive the WM_CHAR message later.

8: If the application is not designed to use IME at all, it will get a WM_CHAR message as the result string is generated. If GetMessageW/DispatchMessageW are used, it receives a Unicode character, which is exactly the same as when getting WM_IME_CHAR. If GetMessageA/ DispatchMessageA are used, it receives two WM_CHAR messages for each DBCS character; the higher byte on the first message, the lower on the second.

An IME is a DLL file, typically with the file extension '.IME', and is usually placed in the *Windows* system directory. IMEs are registered as keyboard layouts in the registry at the following location: HKEY_LOCAL_MACHINE\ SYSTEM\ControlSet001\Keyboard Layouts (see below).



The IME has an 'IME file' value ('winpy.ime'). When these registry values are set, a user can choose to enable this IME through the Control Panel. Once it has been enabled, the IME will be shown in the list of keyboard layouts that we saw earlier.

The registry key HKEY_CURRENT_USER\Keyboard Layout\Preload contains the list of keyboard layouts to be selected by the user. The value '1' is the default layout, which is loaded automatically in every process during the user's session. In Japanese and Korean versions of *Windows*, *Microsoft*'s IME is the default.

AN IME IS LOADED IN EVERY PROCESS

Any keyboard layout, including an IME, is loaded in every process. KBDUS.DLL, the US keyboard layout DLL, contains only data and has no room to place any extra code. But an IME is a program that has several predetermined entries. No application can reject an IME; an IME acts like a system DLL. Before the instruction enters the entry point of an application, the IME file is loaded, thus DllEntry has been called.

Think of Japanese and Korean OSs. If the genuine *Microsoft* IME file is replaced with a hacked version using MoveFileEx or another method, the hacked IME will be loaded, even in the System process. This means the hacked IME can run even in the user sessions where the hacked IME is not the default keyboard layout.

A removal tool cannot repair or replace the IME files because the correct files differ depending on the OS and the Service Pack installed. If the hacked IME hinders the replacement of the files, it becomes even more difficult to fix the problem.

IS NON-FAR EAST WINDOWS SAFE?

For those who don't use an IME, a simple IME that does not convert characters can be installed and become the default keyboard layout. There is a slight difference in UI, especially in the language bar, but most users will not understand why it happened unless they have experience using IMEs.

The average user would have no idea that the IME was running in his/her English OS. They might even search some commonly known registry load points for the culprit, but would likely find nothing.

It is far easier to develop a fake IME, which does not convert characters and does not display a window, than it is to infect or hack a genuine IME, or than to develop a whole new fully functional IME for Chinese, Japanese and Korean users.

AN IME CAN BE ADDED

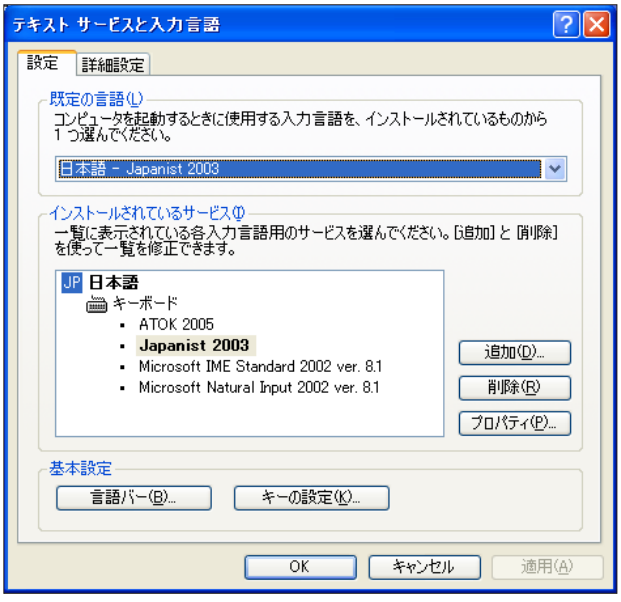
In Korea, there seems to be no need to develop IMEs other than *Microsoft*'s. But in Japan, because many users have become accustomed to their favourite input methods, multiple third-party IMEs are sold.

The image below shows the default display of the IME status bar on Japanese *Windows XP*.

When the icon of keyboard is clicked, the list of available IMEs in the current language is shown. If no IME is added, only one IME is displayed (MS-IME2002 here).

If two IME products are installed on the machine, the list will display three IMEs and their names. The IME currently in use is marked with a tick. (*ATOK 2005* and *Japanist 2003* are third-party Japanese IME products.)

The screenshot shown below is the dialog of the 'Regional and Language Options' Control Panel. The standard language or default language can be selected here. The standard language will be loaded automatically in every



process in the session of the user who has selected the language.

The user also can remove an IME from the list.

IMES CAPTURE EVERY KEY STROKE WITHOUT HOOKING

As already discussed, the OS (IMM) always sends any keystrokes to the currently selected IME through ImeProcessKey and ImeToAsciiEx entry points in the IME file, which is just a DLL with the file extension 'IME'. It generally exports some mandatory entry points that should be called from the IMM.

The simplest way to log keystrokes is to log them in ImeProcessKey and return zero. ImeProcessKey receives virtual keys. If zero is returned, the IMM will no longer interact with IME for that keystroke. IME runs in every process and can act as if it is a part of any program.

If the IME sends out packets in the *Internet Explorer* process to the system or to monitoring tools, it appears as though *Internet Explorer* has sent them out, just like injecting such a routine into *Internet Explorer*.

Unlike the injection technique, however, an IME does not have to hook keyboard-related APIs or messages. Existing security tools would not detect such suspicious behaviour performed by the IME.

If ImeProcessKey returns a non-zero value, a virtual key is eventually passed to TranslateMessage by the application window procedure, then it is passed to the ImeToAsciiEx entry point to be converted to a character.

IMES ARE LOADED IN SAFE MODE

It is not surprising that IME is always loaded, even in Safe Mode. This means that the standard (default) IME cannot be deleted from the system, unless a user with a different default IME logs on.

The MoveFileEx API can be used to rename the IME in use. At the next login, the user will see the second IME in the list become his/her default IME.

MORE HARMFUL ACTIONS

Genuine non-infected IME files can be removed from the computer easily. However, if an IME is designed maliciously, code could be added, making the following possible:

- It may be able to change the default IME of all the users.
- Even if the user changes his/her default IME, IMEs that have already been selected cannot be changed. The user must re-login or reboot the computer. A program that forcefully changes the IME in use can be developed, but on *NT*-based *Windows* the IME module would remain in memory. If the IME runs a thread, it can keep running. And what if the thread checks the standard IME periodically and changes it back again?
- MoveFileEx sets some registry values. If a malicious code deletes the values, it will be difficult to delete the IME file.
- An IME is loaded even in 16-bit applications and the command prompt.
- An IME can load WinSock, enabling it to access the Internet.

DETAILS OF THE IME INTERFACE

The following are some important entry points, or exported functions, that IMEs should have.

DllMain

This is the start address of the IME DLL file. This is called when an IME file is loaded by the IMM during the initialization of an application if the IME is the default, or if it is loaded when the user selected the IME manually.

ImeInquire (LPIMEINFO lpInfo, LPTSTR lpszUIClass, DWORD dwSystemInfoFlags)

The IMM would call this entry at least once to retrieve the IME's properties. On *NT*-based *Windows*, this is called only once and the properties are stored in the global system memory.

An IME can set the member of lpInfo, among which fdwProperty has IME_PROP_UNICODE bit (0x00080000). If this bit is on, all the IME interfaces will become Unicode-based. If it is off, they become ANSI-based. All the interfaces are called from IMM, which would convert between Unicode and ANSI if the application does not match the code system.

An IME should set a string in lpszUIClass, such as 'MY_IME_UI_MAIN'. The IMM would automatically create the window of class 'MY_IME_UI_MAIN' at the time the application creates the first window. IME-related window messages are passed to this IME UI window. This window becomes one of the child windows of the application window. RegisterWindow should be called by the IME.

ImeSelect (HIMC hIMC, BOOL bSelected)

This is called when the IME is selected and unselected. If it is a standard IME, this is called after DllMain and before the entry point of the application is called.

hIMC is a handle to Input Context, in which an IME should store data.

ImeSetActiveContext (HIMC hIMC,BOOL fFlag)

This is called when the application is activated and deactivated.

ImeProcessKey (HIMC hIMC,UINT vKey, LPARAM lKeyData, CONST LPBYTE lpbKeyState)

The IMM calls this first to ask the IME whether this key stroke should be processed by the IME. If the IME returns false (zero), the IMM does not send this key to the ImeToAsciiEx.

If the MSB of lKeyData is on, the key is released, otherwise the key is pressed. lpbKeyState is a 256-byte matrix indicating whether each VKEY is down or up (including CAPS LOCK state, etc.). vKey is just a virtual key and needs the states of lpbKeyState to see what character is input.

ImeToAsciiEx (UINT uVKey, UINT uScanCode, CONST LPBYTE lpbKeyState, LPTRANSMSGLIST lpTransBuf, UINT fuState, HIMC hIMC)

If IME returns true (non-zero) in ImeProcessKey, the virtual key is passed to ImeToAsciiEx. This entry is the most important for the genuine IME; in this routine the IME converts alphabet, Katakana or Hangeul jamo (parts) to Chinese character, Hiragana, Kanji, Hangeul or Hanja. This routine determines the usability and performance of the IME.

IME should set the contents of lpTransBuf in order for a generated character to be input into the application window, otherwise the key stroke will be lost here. A genuine IME would send WM_IME_STARTCOMPOSITION, WM_IME_COMPOSITION and WM_IME_ENDCOMPOSITION sequentially.

A member hCompStr in hIMC has the composition string. hCompStr is a handle to the COMPOSITIONSTRING structure. A genuine IME should set members of dwCompStrLen, dwCompStrOffset, dwResultStrLen and dwResultStrOffset in COMPOSITIONSTRING and (especially dwCompStrLen and dwResultStrLen) may change at every key stroke.

If dwCompStrLen and dwCompStrOffset remain unchanged, the application is highly suspicious as an IME. dwCompStrLen is the length of the characters which are being converted and shown in a special IME UI window. dwResultStrLen is the length of the character string which is determined and input into the application window. Again, if dwCompStrLen is never greater than zero, but dwResultStrLen is greater than zero, the IME is highly suspect.

WEB-AWARE?

An IME that consults the web as to some information related to the input character strings could be developed as a product. But this action should only be initiated when the user performs a specific operation. There would be no legitimate purpose or value for sending all the keystrokes to the web. Similarly, there would be no legitimate reason for sending keystrokes input into a specific window.

If an IME always loads a socket library, there may be conflicts with the application. If the user runs a 16-bit version of an email client, the application would not run properly. Therefore (even though the number of users running a 16-bit Internet application is very low), any quality commercial product should avoid this.

Listening on a port should be avoided too, since it is hard to tell what port will be opened by the application. So, if you come across an IME that does this, beware, it could be a rogue one.

CONCLUSIONS

In the Far East versions of *Windows* used in China, Japan and Korea, genuine IMEs can be hacked and altered to log keystrokes or carry out malicious actions. The installer of the keylogger will replace an IME file and might set some IME-related registry values. Virus analysts must look out for either ImeProcessKey or ImeToAsciiEx entries that log keystrokes in a file or registry, send keystrokes through a socket, or do anything that is unnecessary as an IME functionality. A hacked IME would not behave any differently from the user's perspective.

In the other language versions of *Windows*, the keyboard layout can be changed to a fake IME which does nothing but log keystrokes, send the keystrokes or some other malicious behaviour. In this case users would notice the change in behaviour of the IME, but they are still able to input keys without problems. The installer will drop an IME file, add some IME-related registry values and change the registry of default keyboard layout. In this case, virus analysts should watch out for an ImeToAsciiEx entry that does almost nothing compared to what would be expected in a genuine IME.

FEATURE 2

THE FALSE POSITIVE DISASTER: ANTI-VIRUS VS WINRAR & CO

Andreas Marx
AV-Test.org, Germany

In October 2003 I wrote an article for *Virus Bulletin* about false positives in anti-virus software (see *VB*, October 2003, p.17). To be more exact, the article was about viruses being reported by scanner A in the program or data files of scanner B – and *vice versa*. This problem was caused mainly by unencoded virus scan strings and disinfection routines (e.g. registry keys and files which should be removed) in addition to overzealous heuristics.

COLLECTING FILES

Two years later, we have built up a collection of more than 15,000 GB (15 TB) of clean files in order to enhance our false positive tests. We used two main sources for these files: first, we read in some 10,000 CDs and stored a copy of the ISO images on several storage systems. Secondly, we are mirroring more than 150 different FTP servers and downloading all new files on a daily or weekly basis.

Having such a huge test set creates some problems. For example, a couple of well-known companies have indeed released viruses or other malware together with their software. However, the number of such files is small (about 150) and insignificant compared to the several billion clean files. We left the infected files inside the collection, as all virus scanners should flag them – and if they don't, we know that some scanner tasks might have failed. A couple of these files seem to have been infected by CIH in the past and subsequently cleaned, without removing all parts of the virus, which disqualifies them for both true and false positive tests.

One of the bigger problems is related to the fact that several AV companies release updates at least once a day or even on an hourly basis. This means that test results become outdated rather quickly, since the PCs used for such a test – 15x Pentium IV 2.8 GHz and 15x Athlon 64 3500+ – would require a couple of days to scan the whole collection, for just one scanner. If it took an average of one week per scanner (taking into account common problems like crashes and required restarts) we would need more than half a year just to test the number of scanners that are included in *Virus Bulletin*'s latest VB 100% tests.

TROUBLE WITH WINRAR

To get around this problem, we focused on some key areas only. In the past I have had a couple of discussions with the author of *WinRar*, in particular about enhancing the virus protection in *WinRar* (some malware uses RAR archives instead of just ZIP files) and about a lot of false positives in his software, caused by anti-virus software. The latest *WinRar 3.50* readme file reads:

> '[…] 7. SFX modules: a) SFX modules are not compressed by UPX anymore, so they are larger now. UPX compression caused numerous false alerts by antivirus software. If you wish to use compressed modules, you can get UPX from http://upx.sourceforge.net and compress *.sfx files in WinRAR folder […]'

This was the first interesting test item where all of the scanners could be covered: a scan of the files from ftp://ftp.rarlab.com, which we had been monitoring for a couple of years. So we should have a copy of almost every version of *WinRar* released. We limited our scans to 896 EXE files (877 MB) inside the '/rar' subdirectory of the FTP server where copies of *PocketRar*, *WinRar* and some additional software can be found – and we were a little shocked by the results.

THE FIRST TEST-RUN ...

On 21 August 2005 we tested AV tools from the following vendors:

AntiVir (*H+BEDV*)	*Kaspersky*
Avast (*Alwil*)	*McAfee*
AVG (*Grisoft*)	*NOD32* (*Eset*)
BitDefender (*SOFTWIN*)	*Norman*
ClamAV	*Panda*
Command (*Authentium*)	*Proland*
Dr.Web	*Proventia-VPS* (*ISS*)
eSafe (*Aladdin*)	*QuickHeal*
Fortinet	*Sophos*
F-Prot (*Frisk*)	*Symantec*
F-Secure	*Trend Micro*
Hauri	*VirusBuster*
Ikarus	*eTrust-INO* & *eTrust-VET*(*CA*)

Of the 27 scanners tested, six reported up to 111 infections and two of them reported up to 709 'suspicious' files (see Table 1, left-hand column). Some examples:

- *Avast* reported that a 'sign of "Win32:Trojan-gen. {UPX!}"' was found in the file 'wrar300r.exe\Zip.sfx'.
- *AVG* found 'wr330sc.exe – Trojan horse Agent.M'.
- *ClamAV* reported – 'wr341ro.exe: Oversized.RAR FOUND' and 'wr32b1el.exe: Trojan.Spy.Banker.C*Y FOUND*'.

Program	Infected [suspected] files 1st scan 2005-08-21	Infected [suspected] files 2nd scan 2005-09-11
AntiVir	0	0
Avast	10	0
AVG	1	0
BitDefender	0	0
ClamAV	111	111
Command	0	0
Dr.Web	0	0
eSafe	0 [203]	0 [203]
eTrust-INO	0	0
eTrust-VET	0	0
Fortinet	10 [709]	0 [244]
F-Prot	0	0
F-Secure	0	0
Hauri	0	0
Ikarus	1	1
Kaspersky	0	0
McAfee	0	0
Nod32	0	0
Norman	0	0
Panda	0	0
Proland	0	0
Proventia-VPS	0	0
QuickHeal	9	8
Sophos	0	0
Symantec	0	0
Trend Micro	0	0
VirusBuster	0	0

Table 1: False positives caused by the different AV tools in case of files from ftp://ftp.rarlab.com/rar/.

- *eSafe* complained about the file 'wr341cz.exe – Infected with suspicious Trojan/Worm'.
- *Fortinet*'s detection included '"wr341cz.exe" is infected with the "W32/PoeBot.D-bdr" virus' and '"wr34b1tr.exe" is infected with the "W32/Bancos.GP-tr" virus'.
- *Ikarus* reported 'wr311sc.exe – Signature "Win32.Elkern.C" found'.
- One of *QuickHeal*'s findings was 'pk33b1.exe – Infected : (TrojanSpy.Bancos.B)'.

From an initial look at the 'malware names', it appears that (self-extracting) *WinRar* archives are often used for the packaging of malware, like password-stealing Trojans or Backdoor programs. Furthermore, it looks like some signatures are simply not created properly, which causes false positives when the *WinRar* stub is found. Besides this, some scanners are a little over-paranoid with their heuristics and create too many false positives if files are simply runtime-packed.

The scan time was very interesting too, as some tools were really checking all files inside the *WinRar* archives (which were self-extracting RAR files most of the time), while others only checked the small Win32 stub of the SFX archive, without inspecting the files inside.

For example, *Sophos* proved to be the fastest scanner with a scan time of only 30 seconds, *Trend* took 40 seconds, *Fortinet* about 700 seconds (11.5 minutes), *BitDefender* around 750 seconds (12.5 minutes), *Kaspersky* 1,300 seconds (22 minutes), *Hauri* about 2,400 seconds (40 minutes) and *Proventia-VPS* 3,200 seconds (53 minutes). It should be noted that *Proventia-VPS* is not a virus scanner working with signatures, but a behaviour-based product which requires a longer scan time. From the scan time requirement, one can easily see which of the scanners really inspected all 42,843 files inside the 896 (self-extracting) EXE files. If an AV program doesn't scan the whole self-extracting *WinRar* archive, it is not able to find infected files inside it and thus, it's also less likely that false positives are caused.

... AND THE SECOND TRY

On the day of our initial test, we notified the AV companies, discussed the results with them and provided samples of the files to those who requested them. Then, on 11 September – exactly three weeks after the first test – we repeated the false positive test with the same set of files (no new *WinRar* versions had been released in the meantime).

The number of trouble-makers had decreased significantly, but there were still a lot of files flagged as being 'infected' or 'suspicious' by many of the tested programs (see Table 1, right-hand column).

All of these AV companies were notified again, of course. The high number of false positives generated by *ClamAV* can certainly be considered critical. However, the 203 'suspicious' warnings by *eSafe* and the 244 which were left by *Fortinet* are not really good either.

THE COST OF FALSE POSITIVES

It seems to me that files need to be processed more carefully, especially in the case of installers (like the *WinRar* stub) or runtime engines, as illustrated by the following example.

A well-known computer magazine contacted me on 30 August regarding the games ActionBall 2003, ActionBall 2004 and Jumpy Balla 2003, which can be found at http://www.happy-future-software.de/. Of the 27 AV tools

tested, six found a virus inside the files. *Dr.Web* told us it had found 'Win32.HLLW.Franvir', *F-Secure* and *Kaspersky* both complained about 'P2P-Worm.Win32.Franvir', *Ikarus* found 'P2P-Worm.Win32.Franvir', *NOD32* showed an infection of the 'Win32/Franvir.C worm' and *VirusBuster* reported 'Worm.P2P.Franvir.B'.

After issuing our report, we received a response from *F-Secure*, explaining that the file had been created by GameMaker 4.3, which was also used by the Franvir worm. An email from *Kaspersky Lab* arrived just a few minutes later, explaining that all games created by this tool use the same interpreter stub – it is just the data segment with the game logic that differs. An email from the *NOD32* team arrived three hours later, confirming the false alert and indicating that it will be fixed with the next engine update. However, none of the other companies responded or fixed it.

This false positive was rather significant, as the computer magazine had just produced several hundred thousand cover CDs which included these games. Just a couple of hours later, they would have destroyed all the CDs to make sure they were not about to distribute possibly infected software. With the resulting delay in shipping the magazines (it would have been necessary to remove all 'old' CDs manually and newly created CDs would have had to have been stuck in) and the cost of creating new CDs, the magazine estimated that the damage caused by the false positive could easily have reached a level of several hundred thousand euros.

CONCLUSION

A lot of AV companies have automated the process of creating signatures for static malware. Due to the fact that a lot of malware uses *WinRar* self-extraction archives at some point, the number of false positives had been growing rapidly in this area. False positives could not only prove costly for companies if they find some 'suspicious' tools on their hard disk, but the case of the magazine cover CDs illustrates how else false positives can have a significant impact on businesses. It should be noted that *WinRar* and *GameMaker* were just two examples of what could be many more.

Therefore, a large collection of 'known good' files is essential in order to create high-quality software. Some of the smaller commercial AV companies and the developers of the Open Source project *ClamAV* urgently need to do something in this area.

While well-working processes already exist in order to report new malware and add detection for those files, it is important to attain the same high quality of processes in the case of false positives. This will hopefully reduce the impact of false positives in future and we will be able to remove files causing false alarms faster than ever before.

LETTERS

IN RESPONSE TO REVIEW COMMENTS

In your October 2005 product review (see *VB*, October 2005, p.12), Matt Ham comments that 'The *Kaspersky* entry this month was a great surprise, consisting of a command line scanner rather than the usual GUI' and further that 'the interface required a fully functional SQL database to be installed on the machine in question'.

In fact, the command line scanner is only one element of *KAV 5.0 for Windows File Server*. The product 'interface' is in fact the *Kaspersky Administration Kit* management console. When installed using minimum configuration, with the 'Administration Server' option unchecked, it does not require any database at all.

Of course, *Kaspersky Administration Kit* also provides wider functionality. If a system administrator needs to administer multiple anti-virus installations from one place he can use *AdminKit* (see http://www.kaspersky.com/productupdates?chapter=146274756) on one or more machines in the LAN. This requires MSDE or SQL Server to be available on *any* machine on which *AdminKit* is installed. In this case the Administration Server stores all the data about the corporate anti-virus protection system in an MSDE or SQL Server database.

The review comments further: 'While many servers will have SQL available, those which do not will require a new installation which is free neither in a financial nor in a manpower sense'. While it's true that a *Microsoft SQL Server 2000* licence costs money, MSDE (the *Microsoft SQL Server 2000 Desktop Engine*) is freely available from *Microsoft* at no charge (http://www.microsoft.com/sql/msde/default.mspx). Further, *Kaspersky Lab*, as a *Microsoft* partner, is permitted to re-distribute MSDE with its applications.

David Emm
Kaspersky Lab, UK

AUTHOR'S REPLY

The ability to use MSDE in order to supply a back end for the administrative functions was an unfortunate oversight on my part. *Virus Bulletin*'s policy is, however, that all tests for comparative reviews must be performed in the default state of the product, which in this case certainly seems to be without the administrative database installed. This type of installation, where a GUI is able to be installed somewhere, yet not necessarily on the machine tested, has always been a potential source of debate on server platforms.

Matt Ham
Virus Bulletin, UK

CONFERENCE REPORT

IN DUBLIN'S FAIR CITY

Helen Martin

VB2005 was a double record breaker – *Virus Bulletin*'s longest and largest conference to date. We were delighted to welcome well over 360 delegates to The Burlington hotel in Dublin for the debut of the event's new longer format – and, for the second year in a row, the conference was described by delegates as the best *VB* conference they had attended.

THE IRISH ROVER

In a change from tradition, this year's conference programme kicked off at 2pm on Wednesday afternoon, but delegates also had the option of attending sponsor presentations in the morning. Each of the four conference sponsors (*BitDefender*, *Computer Associates*, *Eset* and *Trend Micro*) was invited to make a presentation on a topic of their choice and the result was four highly engaging and well attended sessions – their popularity largely due to the companies' excellent selection of speakers and topics (and their wise decision to steer clear of too much self-promotion).

By 2pm, as the last of the delegates took their seats for the conference opening address and the opening credits rolled, the larger of The Burlington's two conference halls was filled almost to capacity. Amongst the crowd it was great to see a large number of familiar faces – some of whom we hadn't seen since the conference was last in Europe a couple of years ago – as well as a very respectable number of new faces, who we hope will also become conference regulars.

Four presentations in each stream (technical and corporate) made for a relatively gentle start to the conference on Wednesday afternoon and gave delegates a taste of what was to come over the course of the next two days.

Despite the new start time and the new format, some *VB* traditions are not for changing. One of these is the informal welcome drinks reception held on Wednesday evening. This year drinks were served in the hotel's Buck Mulligan's bar – a traditional Irish-style bar which was soon packed to the rafters with *VB* delegates sampling the local 'water'.

Indeed, the local water became something of a theme at the VB2005 – rarely was a *VB* delegate seen without a glass of the stuff in their hand (after hours of course), and if you don't believe me, just take a look at the photographs!

WHEN IRISH EYES ARE SMILING

If the turnout for the conference was good, the turnout for the gala dinner was exceptional, the numbers boosted by accompanying partners as delegates took the opportunity to show their loved ones that *VB* conferences are not all work and no play. The 420 diners were led Pied Piper-style into dinner by four barefoot Celtic drummers who then proceeded to raise the roof with a spectacular performance on stage, culminating in a frenzied crescendo that was enough to leave ears ringing through the first course of the meal.

Pure genius – the cream of the AV industry relax after hours at VB2005.

Continuing with the traditional Irish theme, the evening's entertainment was rounded off by a Riverdance-style dance troupe who gave a highly energetic performance that was enough to get even the most rhythmically-challenged tapping their toes.

THE SERIOUS STUFF

Of course, between the fun and the Guinness breaks there was a very full programme of presentations which provided excellent fodder for lobby lounge discussions long into the evening.

Continuing where we left off last year, the spam stream was expanded for VB2005, with presentations in both the corporate and technical streams. On the corporate side, Oren Drori looked at commercial and non-commercial ways of fighting spam, Dmitri Alperovitch revealed some of the interesting spam-related statistics drawn from *CipherTrust*'s sender reputation systems, and Jamz Yaneza looked at some best practices for evaluating anti-spam solutions. In the technical stream, Dmitry Samosseiko must be congratulated, not only for managing to keep his audience alert and engaged first thing on Friday morning, but also for drawing a sizeable crowd while Vesselin Bontchev presented in the other stream – neither of which could be described as a mean feat.

Ex *VB* editor Nick FitzGerald explained why he believes user authentication is a bad idea – even going so far as to say that authentication is 'worse than nothing at all'. John Graham-Cumming described his experience of introducing 'pseudo-words' to his Bayesian text classifier, and Matthew Prince reported on the work of Project Honeypot, urging engineers to work together with legislators and law enforcement officials in the fight against spam.

Vesselin Bontchev pulled the crowds in with his presentation on the current status of the CARO Malware Naming Scheme. As well as describing the scheme in full, Vesselin took the opportunity to make a mini-presentation, explaining in his own indomitable style why he believes *MITRE*'s newly-launched Common Malware Enumeration (CME) initiative will end up causing, rather than alleviating, confusion.

In the technical stream Jarno Niemelä revealed 'what makes *Symbian* malware tick' and, with a little help from able assistant Mikko Hyppönen and a video camera, demonstrated *Symbian* Trojans in action live on stage.

Eric Chien outlined some of the ways in which spyware makes its way onto users' machines and described the methods used by spyware to build profiles of its victims. He illustrated the type of detailed data that is relayed by spyware applications. Meanwhile, Joe Telafici and Seth Purdy presented the results of several weeks investigation into 'the Transponder Gang', a convoluted network of interrelated sites, people, companies and unwanted programs, highlighting some of the difficulties that are faced by spyware researchers.

Jason Bruce concentrated on spyware's close relative adware, presenting his ideas on defining 'acceptable' adware so that malicious adware can be blocked while legitimate advertisers can be free to go about their business.

Other highlights included Martin Overton's comprehensive overview of bots and botnets, in which he detailed the full extent of the problem and called for improved security policies and procedures. Charles Renert outlined *Microsoft*'s Data Execution Protection (DEP) and put it to the test against recent exploitation techniques – concluding that, although not a cure-all, DEP is a laudable first step in the fight against vulnerability exploitation. And Kimmo Kasslin demonstrated the stealth techniques used by advanced *Windows* rootkits as well as presenting techniques for detecting hidden objects.

This year's panel discussions were lively as usual. The first of these sessions, led by Gabrielle Dowling, was based around the subject of information provision in a virus outbreak situation. Although the discussion was somewhat hijacked by the topic of media reporting (see p.2), panellists Nick FitzGerald, Eric Chien, Jeannette Jarvis, Dmitry Gryaznov, Andrew Lee and Martin Overton did manage to air some of their opinions. In the second panel discussion, chairman David Perry asked panellists Vesselin Bontchev, John Aycock, Costin Raiu, Andrew Lee, Morton Swimmer and Alex Shipp 'who is hiding the virus writers?' but alas the 50-minute time slot was insufficient for the investigators to truly get to the bottom of the matter.

There is not enough room to mention more than a small selection of the presentations here, but my thanks go to all of the VB2005 speakers for the time and effort they invested – the overall standard of papers this year was exceptional and key to the success of the event.

CANADIAN QUEEN

Although pleased with this year's achievements, it is in the nature of the *VB* team to strive to put on an even better event next year, and planning has already begun for VB2006. Next year *VB* will revisit Canada, this time landing in Montréal – a city that effortlessly combines French flair with North American modernity. The conference will take place 11–13 October 2006 at the Fairmont The Queen Elizabeth. I look forward to seeing you there.

PRODUCT REVIEW

NOD32 FOR WINDOWS NT/2000/XP/2003/X64 WITH CENTRALIZED MANAGEMENT

Matt Ham

Before I begin, I must thank the folks at *ESET* who offered their product for review at very short notice after the software I had originally planned to review was withdrawn. I must also point out that, given such a tight schedule to perform the review, there were some areas where I could not perform quite as many tests as I otherwise would have done.

Secondly, a few words about the product title. The 'x64' part of the product description is inclusive rather than exclusive. That is to say, the executable supplied should run on both 32-bit and 64-bit versions of the operating systems noted, detecting the appropriate software to install automatically. Due to time constraints, the tests were performed on both 32-bit and 64-bit hardware, but the test operating systems were exclusively 32-bit. (As next month's comparative review will cover the 64-bit version of *Windows 2003 Server*, there will not be long to wait for the product to be tested on that platform.) One thing that occurs to me is that the product's name may be a problem in future, since calling a 64-bit program '*NOD32*' seems something of an anachronism, while renaming to '*NOD64*' might cause loss of brand awareness. No doubt *ESET*'s marketing department is pondering this very point as I write.

ESET is a Slovakian company which has enjoyed a long and successful presence in *Virus Bulletin*'s comparative reviews over the years. While the effectiveness of the product has remained more or less constant, the company and product feature set have experienced several changes.

The company now has a much more global presence even than two years ago, with aggressive marketing having worked well from the point of view of sales and general product awareness. While two years ago *NOD32* was little known outside the anti-virus industry, it now features strongly when users are asked to name well-known products.

As for the product's feature set, this has improved markedly from the point of view of a larger organisation. Administration tools are now fully supported, having been non-existent initially. The administrative features were thus investigated in this review as a point of major interest. With comparative reviews in the months before and after this review both featuring *NOD32*, the detection capabilities of the software were not investigated in this test.

The platforms used for testing were *Windows 2000* and *2003 Server* on Intel and *Windows XP Professional* on AMD64 and Intel. Unless otherwise noted, comments refer to the *Windows 2000 Server* platform with *Windows XP Professional* workstations.

WEB PRESENCE AND DOCUMENTATION

ESET's primary web presence can be found at http://www.eset.com/, with various foreign language versions also available. The site contains a collection of press releases, product information and virus data which can be said to be typical of the genre.

While much of the virus data is available in a particularly uninspiring list format, the information on virus occurrences as reported by *NOD32* users is very interesting. Data here is available for particular problem items, with detailed breakdowns of occurrence available over various timescales. 'Problem items' is a term chosen with care, since phishing emails figure largely in the statistics alongside the more easily categorised worms.

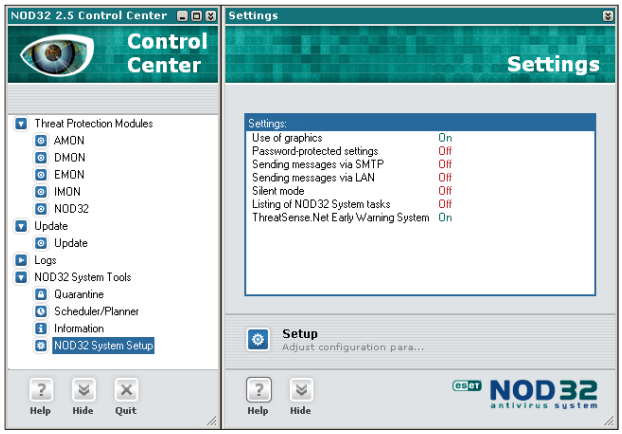
Due to the short timeframe available for the review, the documentation was supplied only in PDF format. In the process of testing the documentation proved thorough, useful and laid out in a logical fashion. The only problems lay in the extensive use of graphics within the file, for decorative rather than illustrative reasons, which made scrolling through the manual somewhat jerky.

However, only light use of the documentation was required, since the help functions within the software and the readme.txt files located in the Start menu gave ample information on how to operate the software. Particularly useful features were the 'How do I add servers?' and 'How do I add clients?' links in the main Remote Administrator Console. These served far better than most quick start guides in providing immediately useful detailed information.

INSTALLATION AND UPDATES

Scanner installation was performed on all platforms from a single 9 MB file. Initially this decompresses to a temporary location before offering the obligatory three options for installation. In this case the options are Typical, Advanced and Expert, with Expert being chosen for the purposes of investigation.

Once this selection has been made the licence agreement appears. Since virus detection statistics can be sent to *ESET* when certain installation options are chosen, this agreement includes a privacy clause. The clause contains the potentially paranoia-inducing statement that this may be information concerning 'you and/or the user of the computer and/or platform' and that *ESET* may then 'share this information with trusted third parties'. On a different note, this licence agreement is one of the few which would



allow the product to be used in nuclear power stations, life support or air traffic control systems.

Next, the choice returns to the mundane with the selection of the location for installation. This is followed with the rather more important setting of update parameters. In order to use the automatic updating feature a username and password are required for the *ESET* update servers. These can be specified at this point. Five server locations are available, though no information is given as to the physical location of these. The facility for automatic selection of server probably makes this information unnecessary, though for particularly paranoid users, it would be helpful to be able to set one download address in stone.

Next in the settings, silent mode may be engaged. This feature is likely to be of little use in a standalone machine, and is clearly designed for administered machines. In silent mode, only those actions requiring direct user intervention are announced on the machine where the software is installed. However, all information is logged for the administrator. Similarly of greater use where an administrator controls the machine, there is the option to password-protect existing settings.

Look and feel options are the next to be addressed in the Expert setup process, with the presence or absence of an *ESET* splash screen being configurable. More control is offered over the whole GUI, with the option to use the non-standard *ESET* style of interface or a GUI that is more instantly recognisable as being *Windows*-based. The *ESET* interface is certainly novel in some of its behaviour, so this option may be useful where users who are particularly easily confused are concerned. The change from one to the other can be performed at any time, though some strange window resizing issues were noted after having done this without an interim reboot.

As mentioned, silent mode supports the gathering of information and the next settings deal with whether additional information is passed by email, instant messenger, both or neither.

The following section concerns ThreatSense.Net, which is *ESET*'s statistics and automated sample submission system. It is this which requires the privacy statement within the EULA. Details given at this point are somewhat more reassuring though. .DOC or .XLS files are never submitted as part of the automated process, which removes a good deal of potentially sensitive data from the equation. The generic data sent for statistical purposes is also far less personalised and less likely to be used for any marketing purposes than the EULA might potentially allow. It is also possible to turn off sample and/or statistical submissions, or to fine tune automated sample submissions by extension. In combination these settings should cover the security requirements of a wide range of potential users.

The next step in this exhaustive setup procedure is to select whether the on-access scanner, AMON, is started automatically when the machine is rebooted. At this point there is a warning that terrible things may occur if another on-access scanner is activated already.

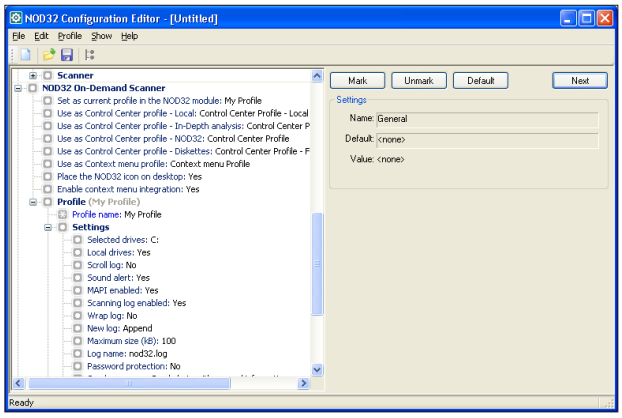
In reality, terrible things did not occur when *NOD32* was installed over a limited selection of other on-access scanners. On the other hand, the installation process on *Windows XP SP2*, did not make use of the Security Center to detect that another product was already present. This would also seem to be a very late stage of the installation decision-making process at which to discover that the machine is not ready for installation due to the presence of other software.

Options continue with those offering access to the on-demand scanner. Context menu activation is a default, though desktop shortcut placement is not, but may be selected. Further selection of options allows the specific scanning of *Microsoft Office* file types using a separate on-access module, in this case named DMON.

Another similar function is available for POP3 and HTTP traffic, this being called IMON. IMON apparently has the potential for conflict with several other products, which makes it a bad choice for server installations, though activated by default on workstations. Admittedly POP3 and HTTP should not be used much, if at all, on most servers and conflicts with, for example, *Microsoft SQL* server can be considered a more important potential source of problems.

Outlook also has its own scanning module, this one named EMON. However, this functionality is not available for any other common mail clients.

With all these selections made, the process of installation takes only a few seconds to finish. However, the full



installation process does not complete until a reboot has been performed – a minor issue on most workstations but a source of grievance on servers.

The administrative server and console were also installed for the purposes of the review. This process was trivial, even when selecting the Expert installation method.

Updates were performed using two different methods, with a third also available. The first method is that used in most *Virus Bulletin* tests in the past – that of a manual update on a local machine. This has been replaced in general usage by the Internet update system, the setup of which was described above. This performed with no hiccups.

The default update timer is one hour, though updates may be set to occur at boot in addition to those scheduled. The initial, largest, update took no more than two minutes on a one-Mbit ADSL line, with updates after this being almost negligible in duration. Clearly, some larger updates will be required on occasion, but network traffic did not seem excessive during normal use. The third method of updates is that achieved through use of the administrative tools. These will be discussed later.

FEATURES

When installed there are three main components to the whole package, which can be considered as separate, though interlinked, entities. These are the main scanning interface of the *NOD32* Control Center, the Remote Administration Console and *NOD32* Configuration Editor.

The Control Center offers the usual functionality of any on-demand and on-access scanning interface and thus warrants only a few comments.

The most noticeable feature visually is the split window interface. Rather than starting as an application with a large blank right-hand pane, *ESET* has opted to open these panes as and when needed. This works well after the initial culture-shock. However, this aesthetic is not fully complete, with tabbed dialogs opening up for some configuration options and the main scanning interface. This is an area which could perhaps be more completely integrated.

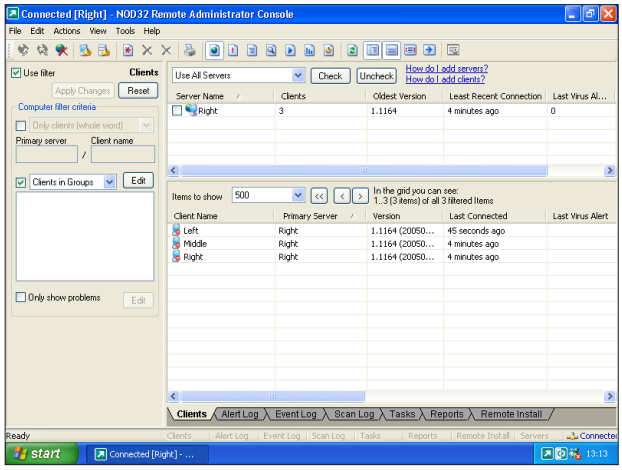
The Configuration Editor is installed as part of the Administration Console package, although it is a standalone application in its own right. The main use of this will likely be the production of custom *NOD32* configurations to be installed onto other machines.

The configuration files, in their native state, consist of XML and as such can be hand-edited by anyone with the barest idea of how XML is formatted. The use of this interface, however, speeds up the process markedly. Allowing trees to be collapsed and temporarily ignored, for example, takes away the pain of regarding a ten-yard-long file while editing otherwise separated portions in the body.

The Remote Administration Console is the part of *NOD32* with which I am the least familiar. As is to be expected in such applications there are two main parts of the administration functionality: the Server and Console. The Server portion must be installed on a machine which has a source of *NOD32* to offer – in most cases this will be a machine connected to an update source. Whether this is connected directly to the *ESET* servers or from updates supplied by a further server is a decision for the individual administrator. Once the Server software is installed it acts as a service and is not visible in day-to-day usage.

Interactions with the Server are made through the Console, which may be installed on any machine which has access to the Server, including the Server itself.

The Console links to the Server, controlling the various possible administration activities. The link between Console and Server proved problematic initially, but the frequent tweaks I was performing turned out to be the problem rather than the solution they were intended to be. Upon leaving the



Console and Server in a stable state for several minutes they connected with no further problems.

As an administration tool, of course, there must be another factor in addition to the Console and Server, this being provided by the clients. Most *Windows*-based platforms are supported, though those that do not hail from the *NT* family – primarily *Windows ME* and *Windows 98* – do lose some administrative functionality.

Existing clients are one matter, but also of interest is the installation from the Server to machines which are not already running *NOD32*. This can be performed in several ways. The most likely is the push method. Using this method a distribution package is produced from an existing installation source of the product. There is capacity for producing a variety of different packages as desired, so as to cater for different users. The clients may be arranged in groups with different packages associated by group.

With the distribution package ready, all that remains is for the package to be pushed to the potential client machines. Many machines may be selected for installation simultaneously, and the network can be scanned for machines which are not already clients. The combination of these two functions allows large-scale distributions to be performed with relative ease. One factor does seem to be a potential irritation, however: machines which have had the package installed require a reboot to initiate scanning, and there did not seem to be an easy way to trigger this automatically. Admittedly, automatic reboots have a habit of enraging end users, but it would be nice to be able to force a reboot upon recalcitrants.

However, this method is not the only one supported. In addition, a smaller application may be sent via email, which when executed will pull the remaining portion of the software across the network and perform the installation. This is the recommended method on *Windows 95*, *98* and *ME* machines. As a user, however, this would leave me with serious doubts, since I have been trained to think of all unexpected email-borne applications as being potentially harmful. A better use of this method – which is indeed suggested in the help files – would be as part of a logon script.

With the installation of machines thus catered for, the updating of machines is a similar, yet less involved task. Since machines to be updated can be assumed to have *NOD32* installed on them already, it is simply a matter of setting the clients to poll the Server for updates, which can be done in their initial configuration. Servers too can be set to receive their updates from local or net-based locations in their initial rollout or subsequently.

Of course the installation and updates process is only half the requirement within the administration component. The other factor is that of centralising the results and alerts from their various clients.

The results are available in two forms. The less useful is the raw data, available under the clients, log, task and remote install tabs of the Console. As they stand these are likely to become overwhelming quickly without some sort of filtering method. Filters are thus provided which can exclude or include servers, clients and varying subgroups of these. Within each category there are more specific filters – for example the 'Show only problems' filter on the clients tab is simple, yet undoubtedly of great use.

Even this degree of filtering has its limits in producing overviews and summaries however, and for this omission the reports tab is supplied. This has a comprehensive listing of different reports available, from those useful for statistical analyses of various virus frequencies to those useful in determining problematic clients. There are sufficient tweakable settings here to produce a good range of statistics in many different formats.

CONCLUSION

The addition of administration features to an already sturdy scanner is always a good thing for the company involved, as long as the administration can be performed usefully. From the (admittedly small-scale) tests done here the basic functionality required in such a product would all seem to be present. Such important matters as the effects of network congestion when installing to many clients were not tested, and thus cannot be commented upon, however.

With its more aggressive marketing methods, and some fairly high-profile head-hunting of new staff, combined with this relatively new administrative clout, *ESET* is clearly hoping for increased market share, especially in the United States. While the future currently looks promising for *ESET*, many other companies with similar ambitions have crossed from Eastern Europe with varying degrees of success. Where *NOD32* will be in the market-share pecking order in two or three years time still remains something of an unknown quantity.

Technical details

Test environment: Identical 1.6 GHz Intel Pentium machines with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive running *Windows XP Professional SP2*, *Windows 2003 Server* and *Windows 2000 Server*. AMD64 3800+ machine with 1 GB RAM, 80 GB hard disk, DVD/CD-ROM and 1 MBit ADSL Internet connection running *Windows XP Professional SP2*.

Developer: Eset Software, 1172 Orange Ave, Coronado, California, 92118, USA. Tel: +1 619 437 7037. Fax +1 619 437 7045. Email sales@eset.com. Web: http://www.eset.com/.

END NOTES & NEWS

The 12th ACM Conference on Computer and Communications Security takes place 7–11 November 2005 in Alexandria, VA, USA. For full details including a list of accepted papers and online registration, see http://www.acm.org/sigs/sigsac/ccs.html.

WORM 2005 (the 3rd Workshop on Rapid Malcode) will take place 11 November 2005 in Fairfax, VA, USA. The workshop will provide a forum to bring together ideas, understanding and experiences bearing on the worm problem from a wide range of communities, including academia, industry and the government. For more details see http://www1.cs.columbia.edu/~angelos/worm05/.

The CSI 32nd Annual Computer Security Conference and Exhibition takes place 14–16 November 2005 in Washington, D.C. Topics covered include: awareness training and education, risk and audit, compliance and governance, critical issues, attacks and countermeasures, forensics, identity and access management, and working with developers. For full details see http://www.gocsi.com/.

The eighth Association of Anti-Virus Asia Researchers International Conference (AVAR 2005), takes place in Tianjin, China 17–18 November 2005. The theme of this year's conference will be 'Wired to Wireless, Hacker to Cybercriminal'. For details email avar2005@antivirus-china.org.cn or see http://aavar.org/.

ACSAC 21 (the Applied Computer Security Associates' Annual Computer Security Conference) takes place 5–9 December 2005 in Tuscon, AZ, USA. The complete programme and online registration are available at http://www.acsac.org/.

Infosecurity USA will be held 6–8 December 2005 in New York, NY, USA. The conference will take place 6–8 December, with the accompanying exhibition running from 7–8 December. The full conference programme will be announced this month. For details see http://www.infosecurityevent.com/.

The inaugural AVIEN/AVIEWS conference will take place from 11am to 4pm Eastern Standard Time on 18 January 2006 by webcast. Details of how to register will be released and circulated on the AVIEN and AVIEWS forums in due course.

The Black Hat Federal Briefings & Training takes place 23–25 January 2006 in Washington, DC, USA. Registration for the event is now open. The deadline for submission of papers is 15 December 2005. See http://www.blackhat.com/.

RSA Conference 2006 will be held 13–17 February 2006 in San Jose, CA, USA. An early bird reduced registration rate is available for those who register before 18 November 2005. For more details see http://2006.rsaconference.com/us/.

The Black Hat Europe 2006 Briefings & Training will be held 28 February to 3 March 2006 in Amsterdam, The Netherlands. For details including online registration see http://www.blackhat.com/.

The 15th EICAR conference will take place from 29 April to 2 May 2006 in Hamburg, Germany. Authors are invited to submit full papers, abstracts and posters for the conference. The deadlines for submissions are as follows: non-academic papers (abstracts) 25 November 2005; academic papers (in full) 13 January 2006; poster presentations 24 February 2006. For more details, including the full call for papers, see http://conference.eicar.org/2006/.

The Seventh National Information Security Conference (NISC 7) will take place from 17–19 May 2006 at St. Andrews Bay Golf Resort & Spa, Scotland. Enquiries may be directed to tina.deighton@sapphire.net or via http://www.nisc.org.uk/ .

The Fourth International Workshop on Security in Information Systems, WOSIS-2006, will be held 23–24 May 2006 in Paphos, Cyprus. For details see http://www.iceis.org/.

CSI NetSec '06 takes place 12–14 June 2006 in Scottsdale, AZ, USA. Topics to be covered at the event include: wireless, remote access, attacks and countermeasures, intrusion prevention, forensics and current trends. For more details see http://www.gocsi.com/.

The 16th Virus Bulletin International Conference, VB2006, will take place 11–13 October 2006 in Montréal, Quebec, Canada. Online registration and further details will be available soon at http://www.virusbtn.com/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery: £195 (US$358)

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 Web: http://www.virusbtn.com/

vb Spam supplement

CONTENTS

NEWS & EVENTS

DMA ADOPTS AUTHENTICATION

The Direct Marketing Association (DMA) announced to its members last month that they will be required to adopt email authentication systems. DMA president and CEO John A. Greco, Jr. explained that this latest addition to the Association's set of ethical guidelines would protect the integrity of responsible marketers' brands and improve the likelihood that legitimate email reaches its intended recipient. He said, 'Consumers can have more confidence they are getting a legitimate, valid offer from a trusted source. Marketers get fewer false positives, increased deliverability and better protection for their brands against illegal use. It's a win-win for everybody.' The DMA has more than 4,800 corporate, affiliate and chapter members from the US and 46 other countries, all of which will be required to adopt some form of email authentication over the coming months.

EVENTS

TREC 2005, the Text Retrieval Conference, takes place 15–18 November 2005 at NIST in Gaithersburg, MD, USA. For more details see http://trec.nist.gov/.

The third Conference on Email and Anti-Spam, CEAS 2006, will be held 27–28 July 2006 in Mountain View, CA, USA. The conference encompasses a broad range of issues relating to email and Internet communication. The conference format includes short and long presentations selected by peer review, as well as invited addresses. Those wishing to present long or short papers are invited to submit their proposals before 23 March 2006. Full details can be found at http://www.ceas.cc/.

FEATURE

MEASURING AND MARKETING SPAM FILTER ACCURACY

John Graham-Cumming
The POPFile Project

'Over 99% accurate!' 'Zero critical false positives!' '10 times more effective than a human!'

Claims about the accuracy of spam filters abound in marketing literature and on company websites. Yet even the term 'accuracy' isn't accurate. The phrase '99% accurate' is almost meaningless; 'critical false positives' are subjective; and claims about being better than humans are hard to interpret when based on an unreliable calculation of accuracy.

Before explaining what's wrong with the figures that are published for spam filter accuracy, and describing some figures that actually do make sense, let's get some terminology clear.

POPULAR TERMINOLOGY

The two critical terms are 'spam' and 'ham'. The first problem with measuring a spam filter is deciding what spam is. There are varying formal definitions of spam, including unsolicited commercial email (UCE) and unsolicited bulk email (UBE). But to be frank, no formal definition captures people's common perception of spam; like pornography, the only definition that does work is 'I know it when I see it'.

That may be unsatisfactory, but all that matters in measuring a spam filter's accuracy is to divide a set of email messages into two groups: messages that are believed to be spam and those that are not (i.e. legitimate messages, commonly referred to as 'ham').

With spam and ham defined, it is possible to define two critical numbers: the false positive rate and the false negative rate. In the spam filtering world these terms have specific meanings: the false positive rate is the percentage of

ham messages that were misidentified (i.e. the filter thought that they were spam messages); the false negative rate is the percentage of spam messages misidentified (i.e. the filter thought that they were legitimate).

To be formal, imagine a filter under test that receives *S* spam messages and *H* ham messages. Of the *S* spam messages, it correctly identifies a subset of them with size *s*; of the ham messages it correctly identifies *h* of them as being ham. The false positive rate of the filter is:

$$(H - h)/H$$

The false negative rate is:

$$(S - s)/S$$

An example filter might receive 7,000 spams and 3,000 hams in the course of a test. If it correctly identifies 6,930 of the spams then it has a false negative rate of 1%; if it misses three of the ham messages then its false positive rate is 0.1%.

How accurate is that filter? The most common definition of accuracy used in marketing anti-spam products is the total number of correctly identified messages divided by the total number of messages. Formally, that is:

$$(s + h) / (S + H)$$

or, in this case 99.27%.

99.27% sounds pretty good when marketing, but this figure is meaningless. A product that identified all 7,000 spams correctly, but missed 73 hams (i.e. has a false positive rate of 2.43%) is also 99.27% accurate.

And therein lies the reason why 'accuracy' is useless. Since spam filters quarantine or delete messages they believe to be spam, a false positive is unseen by the end user. And a false positive is a legitimate (often business-related) email that has been lost. If you had to chose between a filter that loses 1 in 1,000 hams or one that loses nearly 1 in 40, you'd surely chose the former. The difference in importance between missed spam and missed ham reflects a skew in the cost of errors. (For a longer discussion of methods of calculating a spam filter's performance numbers see *VB*, May 2005, p.S1.)

While I'm on the subject of meaningless marketing words, take a look at 'critical false positives' (CFPs). A critical false positive is apparently a false positive that you care about. Anti-spam filter vendors like to divide ham messages into two groups: messages that you really don't want to lose, and those that it would be OK to lose. The handwaving definition of these two groups tends to be 'business messages' and 'personal messages and opt-in mailing lists'. Given that it's impossible to define a critical false positive, spam filter vendors have incredible latitude in defining what is and is not a CFP, and hence CFP percentages are close to useless.

TWO NUMBERS

In my anti-spam tool league table (ASTLT, see http://www.jgc.org/astlt/) – which summarizes published reports of spam filter accuracy – I use two numbers: the spam hit rate (which is the percentage of spam caught: 100% – false negative rate, or *s*/*S*) and the ham strike rate (the percentage of ham missed, i.e. the false positive rate).

A typical entry in the ASTLT looks like this:

Tool	Spam hit rate	Ham strike rate
MegaFilter X	.9956	.0010

This means that *MegaFilter X* caught 99.56% of spam and missed 0.1% of ham. The table is published in three forms: sorted by spam catch rate (best to worst, i.e. descending); sorted by ham strike rate (best to worst, i.e. ascending); and grouped by test. (Entries in the ASTLT are created from published reports of spam filter tests in reputable publications. The full details are provided on the ASTLT website. It is important to note that it's difficult to compare the numbers from different tests because of different test methodologies.)

The top five solutions from the current ASTLT figures (where top is defined by maximal spam hit rate and minimal ham strike rate) are:

Tool	Spam hit rate	Ham strike rate
GateDefender	.9954	.0000
IronMail	.9880	.0000
SpamNet	.9820	.0160
CRM114	.9756	.0039
SpamProbe	.9657	.0014

Here, the 'best' filter is the one with the highest spam hit rate and lowest ham strike rate. In the sample of entries above *GateDefender* is overall best, with *IronMail* close behind.

The use of two numbers also means that charts can easily be drawn where the upper right-hand corner indicates the best performance. All that is necessary is to plot the spam catch rate along the X axis and the ham strike rate along the Y axis (albeit in reverse order). Figure 1 shows the position of the top five solutions in the ASTLT.

However, testing organizations such as *VeriTest* (http://www.veritest.com/) wish to publish a single figure giving the overall performance of a spam filter. The simplest way to do this is to combine the spam hit rate and ham

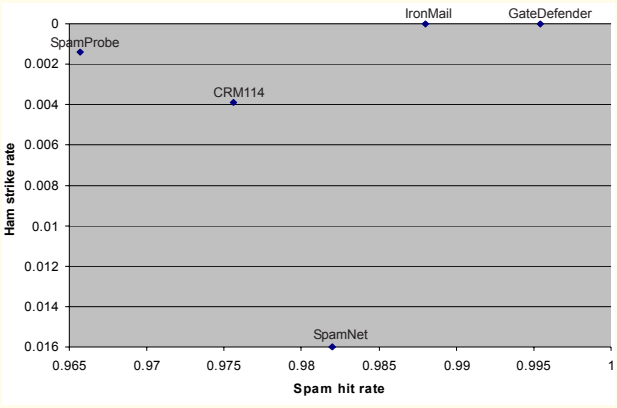


Figure 1: Spam hit rate and ham strike rate for the top five solutions from the current ASTLT. The upper right-hand corner of the chart indicates the best performance.

strike rate by weighting the contribution that those two numbers make to an overall 'performance' score for the filter. Clearly, the way in which the weights are created needs to reflect how much importance an end user gives to missed ham vs. delivered spam.

In *VeriTest*'s case the spam hit rate contributes 40% of the overall score and the ham strike rate contributes 60%. To achieve the final score, the first thing they do is to translate each of the percentages into a score on the scale 2 to 5.

For the spam hit rate the top score, 5, comes at greater than .9500:

Spam hit rate	VeriTest points
At least .9500	5
Between .9000 and .9500	4
Between .8500 and .9000	3
Less than .8500	2

For the ham strike rate the best score comes with less than .0050:

Ham strike rate	VeriTest points
Less than .0050	5
Between .0050 and .0100	4
Between .0100 and .0150	3
Greater than .0150	2

VeriTest then takes the two 'VeriTest points' for a filter and combines them to obtain a final score (between 2 and 5), with 40% contributed by the spam hit rate and 60% by the ham strike rate.

```
Score = (spam hit rate points * 0.4) +
                    (ham strike rate points * 0.6)
```

(For more on *VeriTest*'s methodology see: http://www.veritest.com/downloads/services/antispam/VeriTest_AntiSpam_Benchmark_Service_Program_Description.pdf).

Using that scheme it's possible to score the top five tools in the ASTLT:

Tool	Spam hit rate	Ham strike rate	SHR points	HSR points	Score
GateDefender	.9954	.0000	5	5	**5**
IronMail	.9880	.0000	5	5	**5**
SpamNet	.9820	.0160	5	2	**3.2**
CRM114	.9756	.0039	5	5	**5**
SpamProbe	.9657	.0014	5	5	**5**

The combined scores put four of the tools on the same footing, and only *SpamNet* is scored lower because of its poor ham strike rate.

Part of the problem here is that there is no discrimination between spam filters once they reach a spam hit rate of .9500, or a ham strike rate of .0050. Better discrimination occurs if the scale is extended to 10 points, with the spam hit rate and ham strike rate broken down further.

The top score of 10 is given if the spam filter gives a perfect performance and misses no spam. Between .9500 and perfection each percentage point change (.0100) adds a point:

Spam hit rate	Points
Perfect (i.e. 1)	10
Less than 1	9
Between .9800 and .9900	8
Between .9700 and .9800	7
Between .9600 and .9700	6
Between .9500 and .9600	5
Between .9000 and .9500	4
Between .8500 and .9000	3
Less than .8500	2

Similarly, points for the ham strike rate can be extended to 10, breaking down ham strike rates below .0050 every tenth of a percentage (.0010):

Ham strike rate	Points
Perfect (i.e. 0)	10
Less than .0010	9
Between .0010 and .0020	8
Between .0020 and .0030	7
Between .0030 and .0040	6
Between .0040 and .0050	5
Between .0050 and .0100	4

Ham strike rate	Points
Between .0100 and .0150	3
Greater than .0150	2

Now rescoring the top five tools using the same weighting (40% for spam catching ability and 60% for correct ham identification) a distinction emerges:

Tool	Spam hit rate	Ham strike rate	SHR points	HSR points	Score
GateDefender	.9954	.0000	9	10	**9.6**
IronMail	.9880	.0000	8	10	**9.2**
SpamNet	.9820	.0160	8	2	**4.4**
CRM114	.9756	.0039	7	6	**6.4**
SpamProbe	.9657	.0014	6	8	**7.2**

As spam filters improve, such discrimination between small changes in spam hit rate and ham strike rate are vital in determining which spam filter is the best.

Determining the right weights is difficult and subjective. Is a missed ham twice as bad as a missed spam, 10 times as bad? It's hard to know the answer. What is needed is a way of weighing the cost of an undelivered ham and the cost of a delivered spam.

COST AND SENSITIVITY

To try to model that, imagine that an organization receives *M* messages per year, that *Sp* percent of the messages are spam, and that the organization has determined that a delivered spam costs *Cs* (you choose the currency) and an undelivered ham costs *Ch*.

The annual cost of a spam filter can be determined in terms of its spam hit rate (SHR) and ham strike rate (HSR) as follows:

```
Cost = Sp * M * Cs * (1-SHR) + (1-Sp) * M * Ch * HSR
```

It's possible to simplify that formula when comparing filters by first eliminating *M*, yielding a cost per message (CPM):

```
CPM = Sp * Cs * (1-SHR) + (1-Sp) * Ch * HSR
```

And then, instead of assigning absolute values to the costs of missed messages, replace *Cs* and *Ch* within their relative costs. By assigning the cost of a delivered spam a base value of 1 and an undelivered ham a relative cost of *H* the formula can be used to compare filters:

```
Simplified cost = Sp * (1-SHR) + (1-Sp) * H * HSR
```

And given that the percentage of all messages that are spam is well known (and probably knowable for a given organization), an absolute value for *Sp* can be inserted. Imagine that 65% of all messages are currently spam:

```
Simplified cost = 0.65 * (1-SHR) + 0.35 * H * HSR
```

Now for any spam filter's published or tested spam hit rate and ham strike rate it's possible to plot *H* against the simplified cost. In that way an organization can determine which filter to choose based on the sensitivity to changes in *H*.

Figure 2, for example, is a graph showing the cost of each of the top five spam filters in this article with *H* varying from 1 to 10 (i.e. a false positive is between 1 and 10 times the cost of a delivered spam):

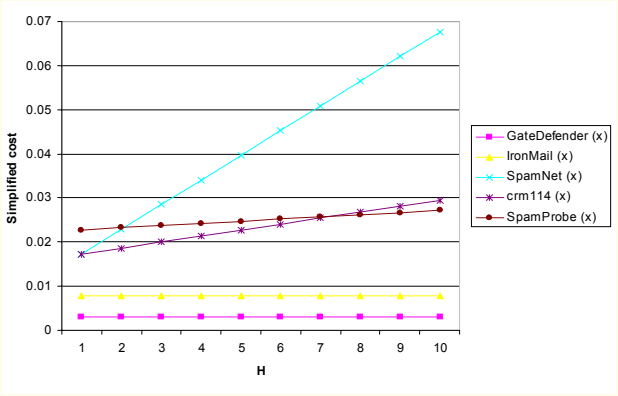


Figure 2: Simplified cost of each of the top five spam filters in this article.

Because *GateDefender* and *IronMail* had a ham strike rate of .0000 the cost is constant and *GateDefender* (with the best spam hit rate) is the cheapest overall. (In a real test it would be better to evaluate the actual spam hit rate and ham strike rate before plugging them into the formulae above; it's unlikely that a ham strike rate of .0000 is currently feasible in the real world).

An interesting cross over happens when *H* is around 7. At that point *SpamProbe* becomes cheaper to use than *CRM114*; this reflects *SpamProbe*'s lower ham strike rate. *SpamNet* quickly becomes the most expensive solution because of its high ham strike rate.

CONCLUSION

Spam filters are becoming more and more accurate; they are catching more spam and missing less ham. But it is still important to weigh two numbers when evaluating a filter: its ability to catch spam and its effectiveness at delivering ham.

(*Author's note: I am always on the look out for new tests to include in the league table; if you know of any please email them to me. The figures in this article are from the published test results that I know about; other tests may show that the products mentioned have better performance than indicated here.*)