JULY 2007

virus BULLETIN

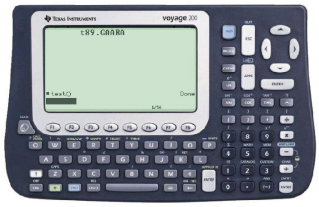
Fighting malware and spam

CONTENTS

ISSN 1749-7027

IN THIS ISSUE

CALCULATE THIS!



Peter Ferrie describes TIOS/Tigraa, a virus that runs on *Texas Instruments* calculators.
page 4

STEMMING THE VARIANT FLOOD

Michael Venable and colleagues explain how program-matching techniques can help in triage, in-depth malware analysis and signature generation.
page 5

LEARNING TO LOATHE

Martin Overton explains why he thinks HTML email is inherently bad and should be considered HaTeMaiL.
page 10

vbSpam supplement

This month: anti-spam news & events, a round-up of this year's EU Spam Symposium, and John Graham-Cumming describes France's new national anti-spam service.

virus BULLETIN COMMENT

'The WildList is more pertinent than ever – particularly given today's threat landscape.'

Mary Landesman
About.com

THE WILD WILDLIST

When the WildList was formed in 1993, it was with the noble intention of protecting users by slicing through marketing hype and identifying the actual threats that anti-virus scanners should detect. In the 14 years hence, the WildList – or more precisely, the WildCore – has become the *de facto* standard by which all reputable anti-virus scanners are measured. But despite its wide adoption, the WildList has struggled to gain respect and has seldom been without controversy. And some say, deservedly so.

A common complaint surrounding the WildList concerns the type of malware represented: only self-replicating viruses and worms make it onto the list – trojans, PUPs, backdoors, bots, adware, rootkits, exploits and nearly half a dozen others need not apply. With such a short list of threats eligible for participation, and such a long list of grievous offenders denied entrance, some question the relevance of the WildList.

Locale-specific malware may impact thousands or even tens of thousands of users. However, the dual reporting requirements of the WildList could prevent a geographically confined outbreak from being properly represented.

Whatever demands the most attention, gets the most attention. Malware that is detected using generics, or is otherwise easily handled by the scanner, will likely be under-reported. Conversely, threats missed by competitors might be over-reported.

Having aired the dirty laundry of the WildList, is it time to pack it in and go home? Are the critics right – does the WildList lack relevance with today's threats and is there a nepotistic management style reminiscent of an old boys' club? Worse, are tests based on the WildList too easy to pass?

As much fun as it is to take cheap potshots and sling similes, the fact is the WildList is more pertinent than ever – particularly given today's threat landscape. By setting a standard, definable bar, the WildList has consistently improved detection across the board. Reputable anti-virus vendors must work (hard) to gain credibility, participating fully in order to engage in the sample sharing necessary to build the library of threats required to score well on the tests. But what WildList testing really offers today is a measure of trust.

The pertinence and sustainability of the WildList is due in no small part to its extensibility. The chief certification bodies – *Virus Bulletin*, *ICSA Labs*, and *Checkmark* – each use the WildList in some fashion as part of their overall certification procedure. This extensibility and widespread adoption has led to considerable credibility for the WildList. That credibility has, in turn, fostered trust. It is this trust that has led to the continued success of the WildList today.

Today's malware isn't a prank. It's not for fun, or for challenge, or to overcome boredom. The imagined idle pastimes of yesteryear's discontented youth are far behind us. Today's malware is about money. And social engineering – the art of tricking the user into infecting themselves – has never been stronger than it is today. One of the favourite tricks for doing this is convincing the user that their system is infected and that 'Scanner X' is the saviour they need. In violation of this trust, Scanner X drops other malware or entices the user fraudulently into paying to remove malware that doesn't actually exist.

Now take away the WildList. Absent any credible, definable, easy-to-understand and widely accepted test criteria. Who are users to believe? Try explaining to your parents – or better yet, your grandparents – why Scanner X is bad and Scanner Y is good. The WildList, and the credibility it brings to the table, is the single best measure we have to draw these distinctions.

Do away with the WildList and we do away with unbiased certification agencies. Do away with the WildList and we do away with the very trust that protects the user. The shortcomings of the WildList can be solved through technology, money, and better management. But trust has to be earned. And the WildList has earned the trust of millions. Let's not consider doing away with that, just when our users need us most.

NEWS

ADVANCE DIARY DATES: VB2008

VB is pleased to announce that VB2008 will take place 1–3 October 2008 in Ottawa, Canada. Reserve the dates and start making your travel plans now!

If you are interested in becoming a sponsor, or require any more information about VB2008, please contact us by emailing vb2008@virusbtn.com.

PITY POOR MS SECURITY WORKERS

Spare a thought this month for the staff of the *Microsoft Security Response Center* who, according to *Popular Science* magazine, have the sixth worst job in science. The science-for-the-masses publication compiles an annual list of those working in 'science' that it feels should be recognised for doing the jobs that nobody else would want to do.

In the 2007 list, working in *Microsoft*'s *Security Response Center* was ranked better only than hazmat divers, oceanographers, elephant vasectomists, garbologists and carcass preparers. In justifying the *MS* employees' rather surprising ranking, the magazine explained that the staff receive approximately 100,000 messages per year, each reporting a possible security failure in one of *MS*'s products. *Popular Science* rates working to fix the bugs in *MS*'s software as 'tedious', and also claims that working in the *MS Security Response Center* 'is like wearing a big sign that reads "Hack Me".'

All together now, one two three: ahh...

CHALLENGE BLUE PILL

Joanna Rutkowska, the security researcher who last year claimed that she can create 100% undetectable malware, has been challenged by fellow researchers to prove it. Rutkowska made the claims about her *Blue Pill* rootkit technology at last year's Black Hat conference. However, Thomas Ptacek, Nate Lawson and Peter Ferrie – who will be presenting a paper at this year's Black Hat entitled 'Don't tell Joanna: the virtualized rootkit is dead' – argue that it is impossible to create a 100% undetectable rootkit, and have invited Rutkowska to prove them wrong.

Rutkowska has accepted the challenge on a number of conditions, one of which is that she and her *Invisible Things* team be compensated for the work they put in to bringing their creation to the required level. She estimates she and her team have already put four person-months into working on *Blue Pill* and that it would take another 12 person-months to get it to a stage at which it was undetectable. Ptacek *et al.* argue that, since they have only spent around one person-month working on their detector, they already stand at a 16:1 advantage. Both 'teams' will present their research at Black Hat USA at the start of next month.

Prevalence Table – May 2007

Virus	Type	Incidents	Reports
W32/Bagle	Worm	2,759,124	27.11%
W32/Netsky	Worm	2,303,735	22.63%
W32/Mytob	Worm	1,917,862	18.84%
W32/MyWife	Worm	759,774	7.46%
W32/Virut	File	402,786	3.96%
W32/Lovgate	Worm	400,973	3.94%
W32/Zafi	Worm	326,987	3.21%
W32/Mydoom	Worm	147,631	1.45%
W32/Sober	Worm	141,590	1.39%
W32/Bagz	Worm	130,778	1.28%
W32/Stration	Worm	107,232	1.05%
W32/Rontokbro	Worm	95,264	0.94%
W32/VB	Worm	79,171	0.78%
W32/Perlovga	Worm	62,857	0.62%
W32/Parite	File	59,342	0.58%
W32/Jeefo	File	54,184	0.53%
W32/Rjump	Worm	49,040	0.48%
VBS/Small	Worm	45,743	0.45%
W32/Funlove	File	42,846	0.42%
VBS/Butsur	Script	24,928	0.24%
W32/Klez	Worm	24,916	0.24%
W32/Looked	File	23,932	0.24%
W32/Nuwar	Worm	23,217	0.23%
W32/Fujacks	File	22,380	0.22%
W32/Sality	File	15,649	0.15%
W32/Mabutu	Worm	14,285	0.14%
W32/Rbot	Worm	12,251	0.12%
W32/Tenga	Worm	11,122	0.11%
W32/Allaple	Worm	10,747	0.11%
W32/Sohanad	Worm	10,654	0.10%
W32/Sdbot	Worm	10,525	0.10%
W32/IRCbot	Worm	7,549	0.07%
Others[1]		79,368	0.78%
Total		10,178,442	100%

[1]The Prevalence Table includes a total of 79,368 reports across 125 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

LIONS AND TIGRAAS

Peter Ferrie
Symantec Security Response, USA

Texas Instruments makes very advanced graphing calculators. What is particularly interesting is that these calculators are programmable, and the resulting applications can be stored in the memory of the calculator. The applications can also be altered by other applications, and that means they can be infected by viruses. This brings us to TIOS/Tigraa, which is the second known calculator virus (the first having been TIOS/Divo), though Tigraa is the more interesting of the two.

MEMORY RESIDENT

When the virus is first executed, it checks whether the ROM call table is in ROM (at 8Mb or greater) or RAM (below 8Mb). This serves as the memory infection marker. If the call table is in ROM, then the virus allocates some memory for it and copies the call table to the allocated memory. The virus uses a function to allocate memory at the top of the heap, then another function to de-reference the returned handle to get the memory address. This is despite there being a single function that performs both of these actions.

In the same way, the virus also allocates some memory to hold a copy of the virus body itself, allowing it to stay resident in memory after the host application terminates. The virus hooks the SymFindNext() function in the new call table, then replaces the original call table pointer with a pointer to the new call table.

PAYLOAD

The virus only checks whether the payload should run when an infected application passes control to the virus code. The payload itself activates only if one of the system timers contains the value 119. However, since that timer is updated at a rate of about 1,500 times per second by default, actually seeing the payload is likely to be a rare event.

The payload is very simple: it clears the screen and prints the text 't89.GAARA ' at column 55 of the first row, then returns control to the host. Despite the message, which suggests the virus may be specific to the *TI89* calculator, the virus also runs on the *TI92+* and *Voyage 200* calculators. Older calculators, such as the *TI84*, use a different CPU (*ZiLOG Z80*), and the virus does not work on them.

SYMFINDNEXT & INFECTION

The SymFindNext() function is a symbol enumeration function. SymFindFirst() begins the search, and the two functions are fairly similar to the FindFirstFile() and FindNextFile() functions on *Windows*, for example.

When an application executes SymFindNext(), the virus gains control. It executes the original SymFindNext() in order to get the information, and the contents of the returned entry will be examined for the possibility of infection. A symbol is a candidate for infection if it is a file (as opposed to a folder) that is not a twin symbol, archived (because the Flash memory is protected), locked, or deleted.

The virus checks that the file is not infected already by searching forwards through the entire file for the string 'GAA'. It is unclear why the virus author did it that way, given that the file size is known, and it would be trivial to search backwards instead – especially since the string appears near the end of infected files. Only after searching for the infection marker does the virus check whether the file is an assembly file, which is the only infectable file type.

Once a suitable file has been found to infect, the virus moves the relocation table further down in the file to make room for the virus body. The reason for this is that the relocation table must appear after the image. It seems that the virus author did not realise that the relocation table is always properly aligned, so copying it can be done in just two instructions instead of 11.

This is the point at which this virus really differs from TIOS/Divo. Whereas Divo replaced the first instruction in the file with a branch to the virus body, Tigraa searches the file for the first instance of a specific instruction sequence ('unlk a6/rts'). If that sequence is found, then Tigraa will replace it with a branch to the virus body. Once again, the virus author appears not to realise that since all *Motorola 68000* instructions are 16 bits long, the instruction will always be properly aligned, so the store can be shortened by one instruction. If the sequence is not found, the file will still contain the virus body, but the virus will never gain control.

CONCLUSION

This virus appears to have been written by a beginner to *Motorola 68000* assembler programming, based on the use of 'lea/mov -(sp)' instead of 'pea', a test for zero after an 'and', the apparent inability to decide between 'clr' and 'sub', and so on. Such things would be forgivable in a true beginner, but the virus author is a well-known security researcher. He released the virus source code under his own name, Piotr Bania, rather than one of his aliases ('Lord Yup' and 'dis69'). It's funny, in a way, that the first virus that we know he wrote came only after he left the 29A virus-writing group. In any case, it's still a virus, and that is to be condemned. Just because you *can* write one doesn't mean that you should.

FEATURE 1

VILO: A SHIELD IN THE MALWARE VARIATION BATTLE

Michael Venable, Andrew Walenstein, Matthew Hayes, Christopher Thompson, Arun Lakhotia
University of Louisiana at Lafayette, USA

The number of variants in malware families appears to be on the rise and is turning into a veritable flood. New defences must be found to detect these variants and curtail the flood.

We propose that program comparison techniques can be an effective shield by assisting in triage, in-depth malware analysis and signature generation. The *Vilo* program search portal is used as an example to illustrate the usefulness of approximated program-matching and the extraction of commonalities and differences from malware variants.

INTRODUCTION

A recent trend in malware production is to generate large numbers of variants at increasingly rapid rates. It is now not uncommon to see thousands of versions of a malicious program released in a short space of time, with each version differing in only minor ways. For instance, consider the recent case of the 'Storm Worm'. *CommTouch Software* reported that over 54,000 variants had been released in under two weeks.

The seemingly endless stream of variants places increased strain on anti-virus researchers, who seek to ensure their products are able to recognize each of the variants.

Something needs to be done to counteract the flood of malware variants. But what?

We argue that program comparison tools are a useful long-term defence against variants – in particular, tools that can determine program similarity, search for matches in a database, and describe commonalities and differences. These tools can be used to organize triage processes and leverage organizational knowledge.

Further, tools that analyse differences and commonalities lie at the heart of assisting in-depth variant analysis and family-aware signature generation. The argument is illustrated using *Vilo*, a set of tools for searching and comparing program variants.

Vilo has been shown to be effective at partitioning malware repositories and can perform searches quickly enough for interactive querying.

This paper introduces *Vilo* and its capabilities. For a more thorough introduction, we invite the reader to visit [1].

THE VARIANT BATTLE

Variant flood attacks are illustrative of how malware authors are increasingly shifting their focus from targeting vulnerabilities in everyday products to targeting vulnerabilities in anti-virus systems. In the case of the variant battle, the weakness is in the defence infrastructure that relies heavily on signatures.

Signatures are frequently reactive – they tend to be effective in defending only after the initial specific attack has been made. In particular, they often fail to detect new versions of malicious programs that have been altered just enough so that the existing signatures no longer match. Moreover, signature generation is a time-consuming, but necessary task that requires expertise.

This combination of properties leaves the infrastructure vulnerable to attacks in the form of a rapid influx of variants.

In one form of the attack, all that is required is to produce signature-defeating variants faster than the signatures can be constructed and distributed. So long as it is relatively easy to crank out a new modification that evades the signatures, it will remain an effective attack. In another form of the attack, a rapid flooding of a large number of variants increases the difficulty of matching all variants whilst simultaneously creating a denial-of-service attack on the limited resources of anti-virus analysts.

Malware authors have recognized these opportunities and are creating variations on a massive scale. The headlines that the 'Storm Worm' trojan have made are unsurprising when one can read the trends forewarned by anti-virus industry reports. For example, according to *Microsoft*'s 'Security Intelligence Report', *Microsoft* found 97,924 variants of malware within the first half of 2006. According to *Symantec* and *Microsoft*, typically only a few hundred families appear in any half-year period. This places the number of variants in an average family in the thousands per half-year period.

The *Microsoft* data shows that the top seven families account for more than 50 per cent of all variants found. The top 25 families account for over 75 per cent. Thus it is a solid bet that any new malicious program found in the wild is a variation of some previous program. The lion's share of the work in handling the flood of new programs would be done if one could recognize even only these topmost 25 families automatically.

Numerous methods can be employed to construct such variants, including packing, manually altering and rebuilding malware, using automated malware generation tools, and automated code modification, such as those found in metamorphic malware. In short, the effort level needed to

create variants different enough to cause havoc is low relative to the number of problems they create. Some means must be found to counteract the variation attacks, but what can be done?

COUNTERING THE VARIANT FLOOD

Behaviour-based heuristics have become an important means of detecting previously unseen variations. Typically, detection in this fashion involves running the potentially malicious program in some type of virtual environment, such as in a sandbox. While executing, the behaviour of the program is monitored until its observed or inferred behaviour sufficiently matches a known proscribed behaviour. Thanks to increasingly powerful machines, this approach is becoming feasible in an increasing number of circumstances. Still, behaviour-based detection methods can be expensive, and there are many ways in which malware authors can defeat the sandboxing or emulation. Other techniques are still desired.

When the variations are constructed using automated methods of mutating the code, the properties of the mutating engine itself may form an entry point for a counter-attack. It may be feasible, for example, to normalize programs before trying to match them, i.e. to remove the variations caused by the mutation engines. Several research groups have worked on this approach, including ourselves. We have shown that it is possible in some circumstances to produce a 'perfect' normalizer for metamorphic engines [2]. It may also be feasible to detect the use of the mutation engine itself by observing properties of the generated code [3]. This approach works much like matching a piece of literary work to its author by observing writing style.

Apart from such mutation engine counter-attacks, the prime counter-attack is most likely to be found in the nature of the variants themselves: similarity.

One possible approach is to capitalize on the high overlap of code between program variants and draw out the similarities and differences in the actual programs themselves. By creating a 'similarity score' between two programs, one could quickly deduce the behaviour of a new sample. Searches can be performed on new samples and anything matching sufficiently closely to a known malicious program can be labelled as malicious.

In addition, knowing the similarities and differences between two files can help steer manual analysis in the right direction. For example, differences can pinpoint new functionality that may need to be analysed further, while similarities identify areas that may previously have been analysed, promoting reuse of organizational knowledge.

To explore this possibility, we have created a demonstration portal called *Vilo* that performs searches on whole binaries and provides tools to assist analysts in extracting the

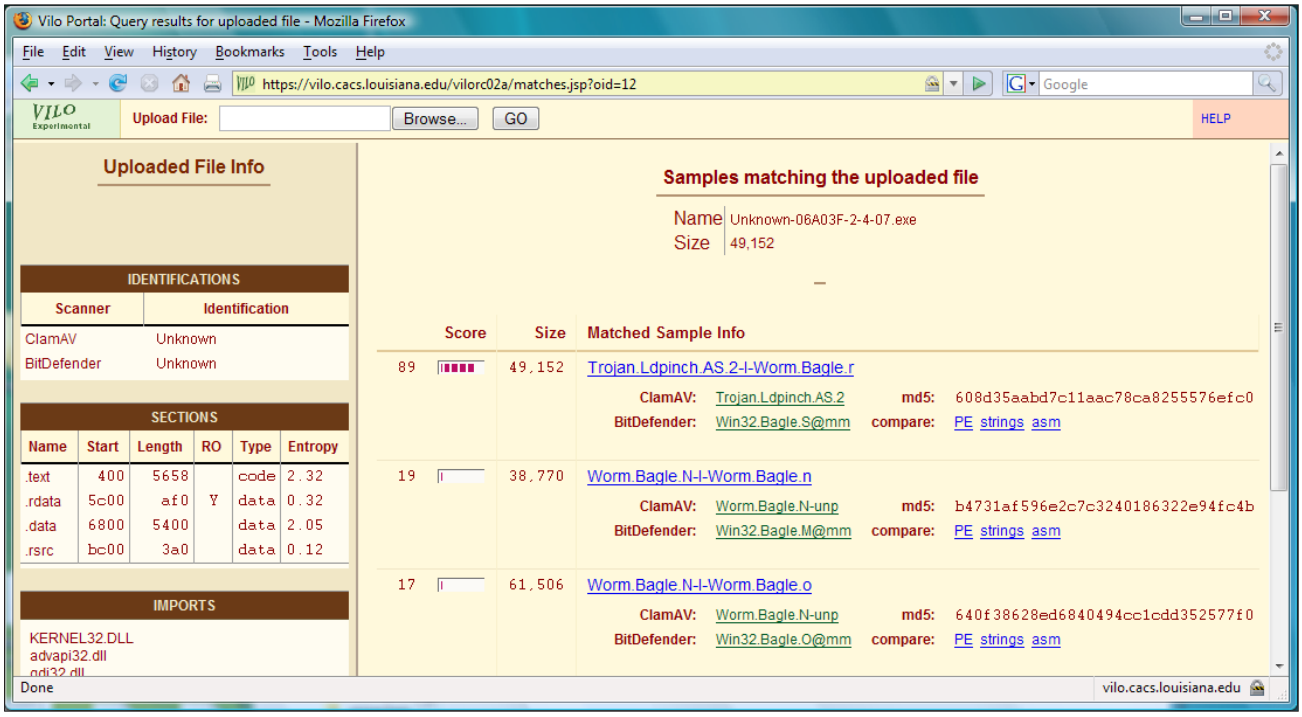


Figure 1: Search results for a variant of the Bagle family.

similarities and differences between files. We argue that program comparison techniques such as these will be important shields in the defence against the variant flood attacks. The general approach is described below and applications to anti-virus analysis are outlined.

VILO

The core part of *Vilo* is a search component. It receives search requests in the form of binary programs. In response, it delivers a list of programs found to be similar to the query, paired with computed similarity scores. A web-based search portal exists to serve as a human-friendly interface to *Vilo*.

Via the portal, users can upload whole binary programs and receive a listing of related programs in order of similarity (as illustrated in Figure 1). For each matched file, users can 'drill down' to view additional information, such as the embedded strings and assembly listing, and compare it against the uploaded file. With *Vilo*, analysts can map malware relationships, find commonalities and dissimilarities between programs, and view 'hot spots' in the assembly listings that the two programs have in common.

The search method used is an adaptation of text retrieval matching using the so-called *tf* x *idf* term-vector query matching methods. These have been used for matching text documents to queries and for the related task of detecting duplicate documents. The search method has been designed so that it is insensitive to changes such as instruction reordering; does not allow common code sequences such as function prologues or code libraries to affect the results adversely; and avoids complicated analysis in preference to simple analysis, such as disassembly, to reduce the likelihood of an unsuccessful analysis.

Figure 2 illustrates *Vilo*'s likely place in the analysis pipeline. *Vilo* has access to the collection of known malicious files and is able to integrate into the existing queue management infrastructure. There, it is available to service requests to support triage, analysis and the generation of new malware signatures.

Next, we will look in more detail at how *Vilo* benefits each of these three areas.

Triage

Anti-virus companies receive new malware samples through a wide variety of sources. It is common to have more sample submissions than people available to analyse them, resulting in a queue into which incoming samples are placed while awaiting analysis.

For efficiency, it is necessary to remove known malicious and benign samples from this queue. This is commonly done by feeding the samples to various anti-virus scanners and removing any files that are identified as malicious. The rest of the samples are submitted to analysts for further analysis.

Unfortunately, many variants are not identified as malicious by the scanners. The unidentified variants must be submitted for further analysis, even though near-identical samples may previously have been analysed. This redundant work can be eliminated by catching the variants before they go to the analysts.

Vilo can assist in this area by filtering files that match closely any known malicious files. Using *Vilo*'s similarity score, it is a simple task to find and remove variants from the queue of incoming files. The web interface in Figure 1 illustrates how the search can help in triage. Here, the results suggest that this sample is likely to be a variant of Bagle.R.

Continuing this example, all organizational documentation on Bagle.R could be delivered along with the new sample to the analyst, thus promoting knowledge reuse and reducing the amount of rediscovery needed on the part of the analyst.

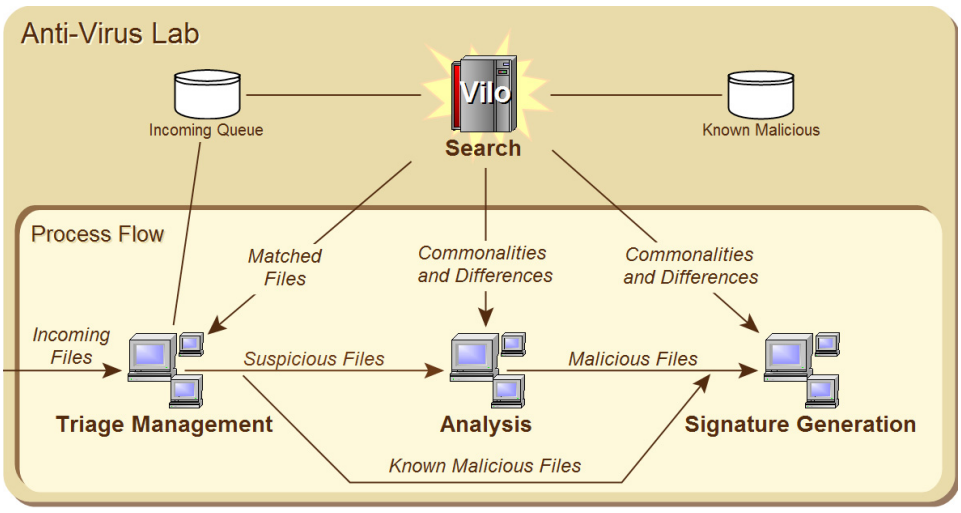


Figure 2: Vilo architecture and process flow diagram.

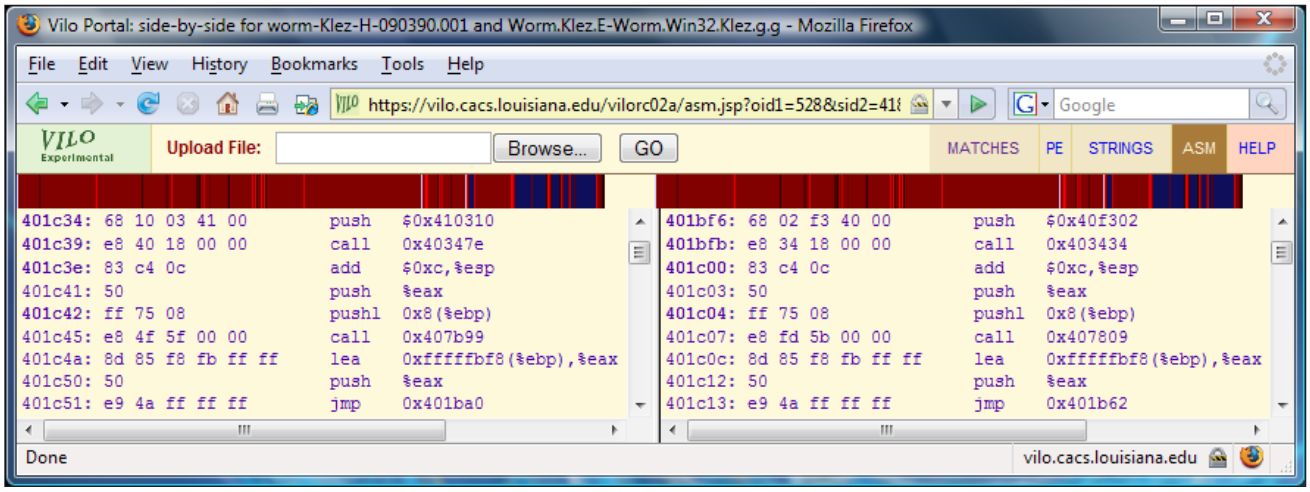


Figure 3: A match found between two variants of the Klez worm.

Analysis

During analysis of a new sample, it can be helpful to have similar files at hand (particularly if past analysis results can be retrieved as well) and to know what makes the files alike as well as how they differ. This information can be used to guide the analyst and to decrease workload. Knowing how two files differ helps the analyst quickly identify new functionality that has been introduced in newly released variants. In addition to running the malware in a virtual machine in the hopes of learning its behaviour, the analyst can find the exact location of new code and can then use that to determine what step to take next in the analysis process.

Similarities identified by *Vilo* can provide valuable insight into the behaviour of a new sample before beginning any detailed analysis. Not only is knowing that a sample is 90 per cent identical to some other sample a good indication that they share a lot of functionality, but instructions in the sample can be matched against a database of code segments further to reveal specific functionality. For example, if a code segment within the sample under inspection matches a segment in the database that is known to be a backdoor, then it can be concluded that the sample also features a backdoor, without the need to launch a virtual machine. *Vilo*'s code comparison tools make this possible.

Vilo allows the user to view a side-by-side comparison of the assembly listing for the uploaded and matched files. Included in this view is a colour-coded overview bar making it easy to spot commonalities quickly among the two assembly listings. A section of the bar with bright red colouring indicates that the corresponding part of the file contains a high number of matches, whereas dark blue indicates very few or no matches. The user can click the overview bar to go to the corresponding position in the file, making it a snap to zoom in and find code similarities between two files.

Figure 3 shows a comparison of samples of two variants of the Klez worm. The figure shows the degree of commonality between the two files. This can be seen by glancing at the overview bar near the top of the window.

The lines of code are also colour-coded and can be clicked to have the program find the corresponding matching code in the other file. In the figure, we've done exactly this to find a piece of code shared by both files. The selected portion is shown as blue text. Notice the matched lines are not identical (jump targets are different). *Vilo*'s approximate search is not affected by such simple differences.

Though not shown, users can also view a similar comparison of embedded strings as well as PE (Portable Executable) file information.

Signature generation

Current static signature generation typically involves extracting a byte sequence from the sample that is common among variants while distinct enough to limit false positives.

When done manually, this is a very time-consuming activity requiring a good understanding of the malware on the part of the analyst. *Vilo*'s search makes it possible to find all common variants easily, and its binary comparison algorithm provides the functionality needed to isolate similarities and differences, making it possible to create signatures that are relevant to all or most of the members within a family.

CONCLUSION

For years, anti-virus products have relied on the presence of static signatures within malicious software as a means of malware identification. In many cases, it is possible to identify several variants of a known malicious program using only a single signature. However, as the number of variants increases, the number of signatures required grows, as well as the time required by analysts to inspect the variants.

Malware authors have realized this and have begun creating variants on a grand scale, reaching into the tens of thousands and easily overwhelming the current infrastructure.

Vilo offers a unique search algorithm suitable for finding variants of known malicious programs, making it applicable in the areas of triage, manual analysis, and signature generation. *Vilo* can operate in the anti-virus back-end as a filtering tool of incoming malware samples. Already analysed malware samples could be culled from incoming malware queues and related programs could be grouped together to improve the efficiency of the analysts.

Vilo includes a web-based user interface that, when given a program, presents the user with a ranked ordering of related programs, making it possible to map out malware relationships. *Vilo* also provides tools to isolate the differences between two files. This information guides the analyst by highlighting new functionality to be analysed and reduces the amount of time needed to analyse a file. In addition, *Vilo* can assist in signature generation by identifying pieces of code that are similar among a group of files.

Malware authors have attacked a weak spot in the anti-virus industry, but the high degree of similarity between variants can prove to be a weakness in its own right. *Vilo*'s patent-pending search algorithm is well-suited for detecting the types of variations typically found in malware, making it a good defence against the incoming flood – a shield in the variation battle.

REFERENCES

[1] Vilo website: http://vilo.cacs.louisiana.edu/.

[2] Walenstein *et al.* Normalizing metamorphic malware using term rewriting. Proceedings of the 6th IEEE Workshop on Source Code Analysis and Manipulation, 2006.

[3] Chouchane *et al.* Using engine signatures to detect metamorphic malware. Proceedings of the 2006 ACM Workshop on Rapid Malcode.

FEATURE 2

HaTeMaiL EMAIL!

Martin Overton
Independent researcher, UK

George Santayana is credited with the following statement:

> 'Those who cannot learn from history are doomed to repeat it.'

And David C. McCullough with the following statement:

> 'History is a guide to navigation in perilous times. History is who we are and why we are the way we are.'

How can we possibly understand risks and threats if we fail to look at their history, and more importantly, learn from it?

With this in mind, let us look back, before looking forward once more.

IN THE BEGINNING ...

> 'I remember the good old days when all email was plain ASCII and to indicate that something was in bold you put a * either side of the word you wanted to emphasize. There were no different-sized fonts, no colours, and no inline pictures (unless they were made up of ASCII characters) ...
>
> 'And then some bright spark decides it would be a good idea to use this new-fangled format called HTML [HyperText Markup Language] ...
>
> 'Pah! Using HTML for email instead of good old plain ASCII, it's just asking for trouble ... it'll all end in tears, mark my words ...'

No, that wasn't me, but it's a common view among those of us who have been in computing a few decades and grew up using the fledgling internet; we who cut our teeth on ASCII email, FTP and NNTP, as well as the more advanced tools available at the time, such as Gopher and WAIS.

I don't use HTML-based email unless I have to, as I still prefer to use plain ASCII. Call me old-fashioned if you wish, but at least I know that there are no nasty HTML exploits in my email, or embedded scripting languages that will be executed when I read the email. No web-bugs, no remote graphics are loaded, unless I want them to be.

LEARNING TO LOATHE

HTML email does more than deliver pretty stationery, clickable links and pictures to our inboxes. It can be the way in which your system becomes infected or how an advertiser or spammer/scammer knows you have opened/viewed the email.

Not convinced that HTML in email is inherently 'bad' and should be considered HaTeMaiL? Well, let me try and show you a number of malicious examples to see if I can convince you.

First I will cover a couple of historical incidents, and then we will move on to more recent times.

BURSTING THE BUBBLE

VBS/BubbleBoy [1] was the first worm that was able to spread via email without requiring the recipient to open (launch) an attachment.

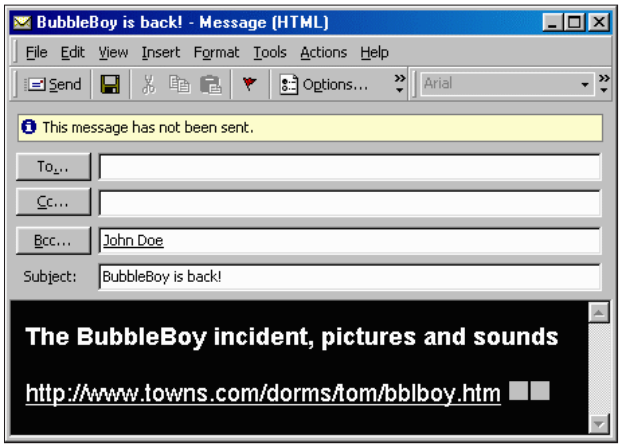


Figure 1: BubbleBoy email. Image courtesy of F-Secure.

Simply rendering/viewing the email in vulnerable versions of *Outlook* will infect a vulnerable system. Scripting can be embedded in HTML-based email so that the script is run automatically when the mail is rendered, and unless you look at the raw email source you won't even know it is there!

BubbleBoy then goes on to modify the owner registration details for the copy of *Windows* that it has just infected to show it registered to 'BubbleBoy, Vandelay Industries'.

Finally, it mass-mails a copy of itself in a similar fashion to Melissa [2].

IT'S ALL KAK

Kak [3] is a worm that, like BubbleBoy, embeds itself into every email sent from the infected system, without any attachment.

Just like BubbleBoy, it infects a vulnerable system on previewing or opening the email, no clicking or double-clicking required as there is no attachment.

Kak is written in JavaScript and it works on both English and French versions of *Windows 95/98* if *Outlook Express*

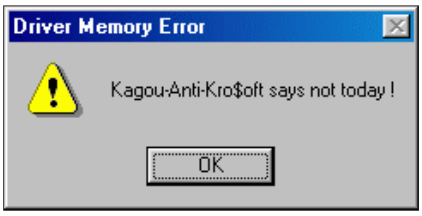


Figure 2: Kak payload message, image courtesy of F-Secure.

5.0 is installed. It does not work in a typical *Windows NT* installation.

The worm triggers on the first day of each month, but only if the system time is later than 18:00. When it triggers it shows the message seen in Figure 2 and then proceeds to shut down *Windows*.

Now let's move on to the more recent uses of HTML in email.

GREETINGS!

One of the current uses of HTML email by malware authors is sending out fake e-cards (electronic greetings cards) to attempt to get people to infect themselves via a social-engineering trick.

The following are a few examples of the fake e-card HTML emails I've seen recently:

Figure 3 shows a professional-looking HTML email, which may very easily be mistaken for a real e-card by an intended victim.

Figure 4 shows an HTML email which uses one of the most successful social-engineering techniques employed by malware authors.

Figure 5 shows another well thought out fake e-card notification, which even mentions that the e-card is a *Flash* executable, thus increasing the chances that the intended target will run the file without another thought.

Of course, each of these lead not to a card but to a malicious file, usually a trojan.

Probably the cleverest example I have seen so far this year was a fake Valentine's Day card which prompted the recipient to install a fake plug-in (malicious software disguised as a *Flash* plug-in) when they visited the website to retrieve their e-card. The clever part was that the prompt to install the 'plug-in' was only displayed the first time the user visited the site – on any return visits the user would simply see a real e-card. More details can be found in [4].

The next section deals with another interesting email that not only uses social engineering but also exploit code. Like

Figure 3: Fake Hallmark e-card email.

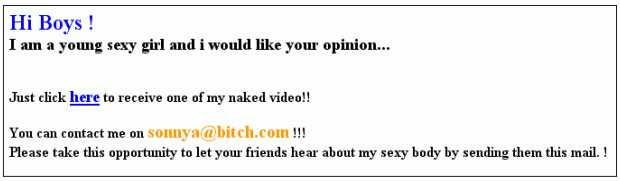


Figure 4: Fake Greetings.com e-card email.

Figure 5: Fake 2000Greetings.com e-card email.

BubbleBoy and KAK, the exploit code will trigger on unpatched systems when the email is opened or previewed on most email clients, not just *Outlook*.

Swen [5] arrives as a very professional-looking HTML email claiming to be from *Microsoft* and warning of a new virus on the loose. The warning email just happens to include the 'patch' to stop the virus; how kind of them to send it out to all their customers! Of course, the patch is, in fact, the virus and the email didn't come from *Microsoft* at all.

Maybe you now see why it is important to look back, as many tricks/techniques are rediscovered, dusted off and reused.

PRETTY, BUT DANGEROUS

Don't look ... I told you not to look!

Figure 6: Paris Hilton WMF email screenshot.

Too late, if the screenshot shown in Figure 6 was an email you had previewed or opened on your system, and you hadn't patched or had other mitigating technologies or methodologies in place (such as a good, up-to-date, and enabled, anti-malware solution and/or fully patched system or one not using *Windows*), then your computer would now be infected. Yes, you would be '0wn3d'.

I have doctored the screenshot, since the real one is a little too risqué to display here. The first picture, the one of 'Paris Hilton' barely wearing anything, is not 'bad'. What I mean is that the picture itself is not the problem in this email, it is simply the bait. The one to worry about is the second picture, which won't render (the one with the red diamond in the screenshot), because it isn't a real picture at all. It is a trojanised Windows MetaFile (WMF), which has exploit code embedded in it to try and infect or take over your computer.

So, why am I writing about this now? I mean, the exploit code used is old, and you should all be patched by now.

The reason I'm flagging this is that I believe that there will be a new phase of 'image' exploitation (in both senses of the word) such as this one using the 'WMF exploit'. I suspect we will see the same social-engineering techniques used with other exploit code and droppers. In fact, I know we will!

ANI EXPLOIT WILL DO

It is not the first time that Paris Hilton has been used as an incentive to exploit *Windows* vulnerabilities. At the beginning of April, pictures of the Hilton heiress as well as pictures of Jenna Jameson and Britney Spears were used as bait for potential victims of the .ANI vulnerability (see *VB*, May 2007, p.4).

CONCLUSIONS

If you haven't already, I would strongly suggest that you set your email program not to render HTML automatically or to download remote graphics. Most modern email clients now have this as a default setting.

If you must use HTML-based email, then please be careful when opening or even previewing HTML emails, as you may start a chain reaction which ends up with your system being turned into a zombie, or worse, and it's all downhill from then on.

There are numerous reports [6] that people are abandoning email as a communication medium. It is claimed that this is mainly due to spam and malware. Certainly statistics from my personal mail server show that in May 2007 over 91% of the mail I received was unsolicited. This is the highest percentage I have seen since I started collating this data at the start of 2004.

So, let me now play devil's advocate, how many of you reading this article agree with the following?

- HTML email was an accident waiting to happen.
- HTML has no real place in emails at all.
- HTML should stay on web pages where it was always meant to be.

And a final question for you:

- Is the reputation of email now so badly damaged that it can never recover the relative trust it once had?

Please send answers to me on a postcard (e-card), a real one that is. Let the flame-fest begin!

REFERENCES

[1] A full description can be found at http://www.f-secure.com/v-descs/bubb-boy.shtml.

[2] A full description can be found at http://www.f-secure.com/v-descs/melissa.shtml.

[3] A full description can be found at http://www.f-secure.com/v-descs/kak.shtml.

[4] http://momusings.com/momusings/2007/02/stupid-cupid-stop-picking-on-me.html.

[5] A full description can be found at http://www.f-secure.com/v-descs/swen.shtml.

[6] For example http://www.castlecops.com/a3794-Spam_pushes_many_to_stop_using_e_mail.html and http://www.symantec.com/press/2003/n031202a.html and http://www.crn.com/security/18842210.

PRODUCT REVIEW

AVIRA PREMIUM SECURITY SUITE

John Hawes

Avira, the company formerly known as *H+BEDV Datentechnik*, has been in business for over 20 years, and the first product based on its *AntiVir* engine was released in 1988. Based in Tettnang near Lake Constance, where the borders of Germany, Austria and Switzerland meet, the company has offices and labs in several other regions and partner links in still more, employs around 250 people and boasts over 15 million customers protected by its software.

The *AntiVir* product range now includes a variety of corporate and home-user offerings, including *Linux* and Unix products – an area which is something of a speciality for the company; the open-source *Dazuko* tool, used by many other security products, was designed, and continues to be maintained, by *Avira* developers.

The company has, for some time, made a basic version of its anti-virus software, *AntiVir Personal Edition Classic*, available free of charge to home users. A premium version, featuring extra detection capabilities including adware and spyware, is released under licence. Corporate desktop, server and gateway products, firewalls, management tools, and protection for mobile devices complete the set of solutions offered by the company.

The *Premium Security Suite* is a home-user internet security setup, and includes the full range of *AntiVir* malware protection, covering multiple vectors, alongside a firewall, spam and phishing filters, and the latest addition, a rootkit scanner.

WEB PRESENCE

Avira's main website, www.avira.com, has an attractively simple home page, with plenty of white space to keep the product information clear and clean, and a smattering of warm red, mostly in the form of the umbrellas that are the company's motif.

Prominent placing is given to the latest and most significant threats, and information on the latest product updates is also ready to hand. A news section at the bottom of the pages carries stories on the latest significant events in the malware world, and the 'Company news' area boasts proudly of *Avira*'s recent excellent ratings from both AV-Comparatives and AV-Test.org, whose testing placed *Avira* at the top of the league for detection, heuristics and scanning speed. *Avira*'s products have a similarly strong performance history in VB100 testing.

The 'Virus Info' section is particularly thorough, with a comprehensive and well laid-out encyclopaedia of malware descriptions. As detailed in a presentation at last year's *VB* conference, *Avira* uses a sophisticated, automated description-creation system, which builds the malware description pages in multiple languages as part of the process of analysing and adding detection for new items. The system is improved continually as the amount of data available is expanded. The website also provides a selection of interesting statistical information, presented in graphs and tables, some security news items, a malware glossary and links to the WildList.

The 'Support' section carries some standard information on updates and product life cycles (support for *Windows 95*, *98* and *ME* platforms is scheduled finally to have come to an end just prior to the publication of this review). A fairly thorough FAQ is provided, covering a broad selection of common issues ranging from where to buy software to the meaning of specific error codes, along with a system for contacting support staff for help. An online form is provided for this purpose, as well as telephone contacts in Germany, Austria and Switzerland. The main focus of the support section for users, however, is an online discussion forum, which seems, if my rather rusty German is any guide, quite a busy place with questions posted regularly and answered pretty rapidly, by both company experts and community-spirited volunteers.

Elsewhere on the site is a wide range of the usual company and product information, with most products and associated updates and patches available to download, protected by a system of licence keys.

The company is the proud owner of the www.freeav.com and www.free-av.com domains, where its freely available home-user product is promoted. The company's founders also run a charitable organization, the Auerbach Foundation, which receives a percentage of all *Avira*'s product sales and puts the funds to use in a variety of community projects – more details of their good works are available at www.auerbach-foundation.com.

INSTALLATION

The product provided for review came in the form of a download, available from the *Avira* site as either a zip or a self-extractor, measuring around 18 MB. Updates were also provided, some 14 MB of them, in a 'bundle' format which the product can be made to absorb into itself, for use in the *VB* test lab and other offline situations.

The installation process was pretty straightforward, with the only options along the way being a selection of modules to include. After spending some time watching files being

extracted, in a window adorned with one of those red umbrellas and the slogan 'More than security', there was a EULA, the application of a licence key in the form of a file with a '.KEY' extension, and then the options, which allowed components such as the firewall, rootkit scanner and mail 'guard' to be dropped from the install. Not wanting to miss out on any of these treats, I let the installation finish, and was offered a readme, which of course I did not read too thoroughly before allowing the reboot requested in a small blue window behind the text page.

When the machine rebooted, a splash screen showed a picture of a chap clambering around what looked like a complex climbing frame, carrying another one of those red umbrellas. A pop-up told me that my last update took place more than three days ago, and the machine was up and running without undue delay.

Updates did not seem to be initiated automatically, and of course attempting to run the update process in the isolated test lab resulted in disappointed error messages about failing to find the required site.

Moving to a machine with a web connection, I was puzzled to see similar results. Checking the connection, web browsing seemed fine through several browsers, and pinging the address of the update servers was successful too. After some tinkering, I resorted to browsing the forums on the *Avira* support site (my limited German told me that most queries on this sort of subject were solved with the question 'Are you running a second firewall?'). I also sent off a support query via the site, but eventually discovered, thanks to more careful perusing of the readme on another install, that running *Cygwin* in conjunction with the product was known to upset the update process, and also that *PGP Desktop 9* may cause users some difficulties.

The manual update process used in the test lab proved much simpler and entirely error free, and the incremental 'ivdf' update files are readily available from the website, with a link provided from the home page.

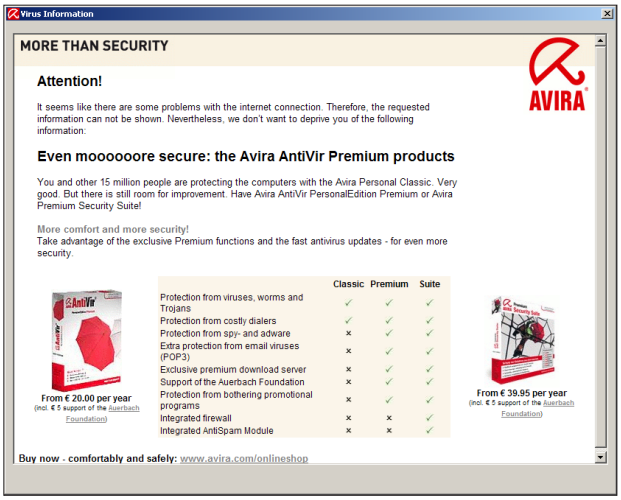
INTERFACES

Once the product was running and safely up to date, I was able to take a calm look through the delights offered by the main interface. Once again, this was decorated with the image of the same chap clambering about high in the air with his umbrella, but the control area below was much more straightforward, angular and serious with a lot of white space, small unfrivolous icons and lots of text.

The opening 'Status' page showed that all my guard systems, including the firewall, were up and running, that my licence was valid, and also told me whether my updates had been successful. Each section had a 'deactivate' link which seemed very responsive, allowing the main modules, on-access protection from malware, mail filtering and firewall, to be switched on and off with ease. There were also links marked 'Configuration' and 'Help' at the top of the page, which led respectively to a separate configuration interface and a detailed-looking help system.

Sticking with the main GUI, a row of tabs led to various aspects of the product. The 'Scanner' tab came first, and included a selection of common areas and groups of areas to scan, as well as a nice simple 'browse' option to pick out specific files or folders, and a system to design custom 'profiles' for repeat scans.

Next came the 'Guard' section, with details of what the on-access scanner had been up to, how many files had been scanned, what nasties had been spotted and what had been done with them. A link to 'virus information' led to details on particular threats on the company website, but on unconnected systems popped up a jovial window informing



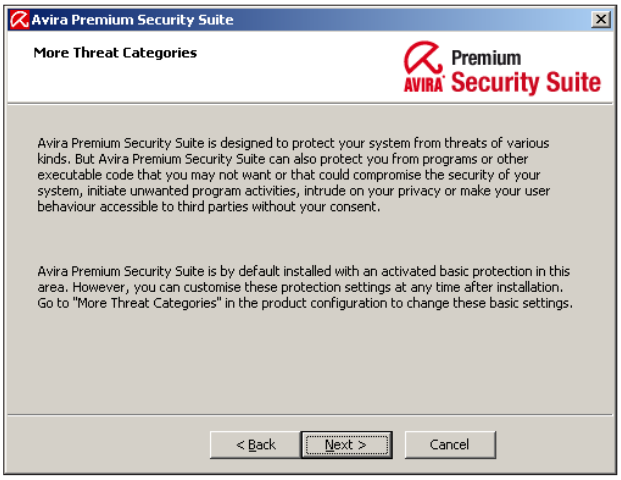
me of my lack of internet connectivity, and suggesting I may want to upgrade from the free version to one of the premium products (making the assumption that anyone without an internet connection is likely to be using the free, home-user version of the software).

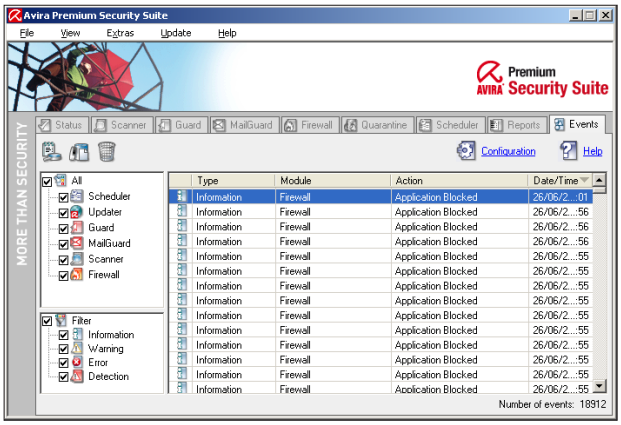
Moving away from the malware-scanning basics, the 'MailGuard' tab showed me the latest emails processed along with some more nice statistics, and offered various options for white- or blacklisting senders, and even switching off spam and malware scanning for certain addresses. The firewall page had a simple slider control, and more data on what was travelling in and out of my machine, along with a list of processes using the network.

The remaining tabs all featured simple, mostly clickable lines of text entries for different subjects. The quarantine page showed all items moved aside out of danger, which could be manipulated in all the expected ways. The 'Scheduler' tab included some default jobs, a full system scan, set up for once a day at noon but not active by default, and an update job, which seemed to set itself up for randomly allocated timeslots.

'Reports' listed the outcomes of various scans and updates, with details opened up on clicking, and 'Events' showed a list of actions and errors, including firewall activity. Oddly, on the system that had failed to apply the offline update successfully, there was no mention of the failure on either of these last two tabs, although failures to run an update from the network were clearly flagged.

Moving on to the configuration interface, at first glance there seems little choice available. For the anti-malware side of things, the file types scanned can be adjusted, the types of access which spark scanning, and the firewall rules and allowed/blocked applications can be changed, but that's about it. Of course, the little 'expert mode' check box makes





all the difference, with a far wider range of options opening up when it is ticked.

Pretty much everything seems to be covered here – the malware scanner has all the appropriate controls for actions and alerts on detection, depth and types of archives scanned, exceptions, heuristics levels and reporting.

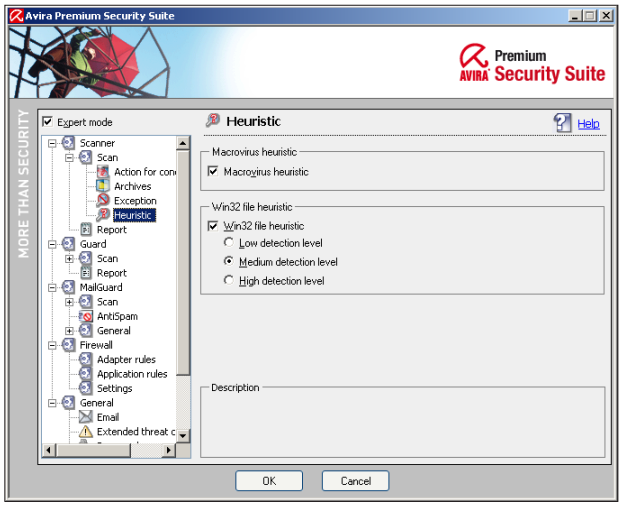
The mail filter has similar settings for malware plus some further tweaking options for the spam filter, although there is no obvious tuning of the paranoia level here, while the firewall has the usual range of default and user-defined rules, sorted into those relating to individual network adapters and connections, and those associated with specific applications, plus a further page allowing one to disable the *Windows Firewall* and to lock down the hosts file.

A final section, marked 'Extras', provides some controls specific to the suite itself, such as some 'extended threat categories' for which detection can be added, and password protection for the controls, an area which offers particularly granular tuning with any part of the interface lockable.

The entire configuration interface features a handy information box in the lower part of each page, which carries a simple description of each option as the mouse hovers over the related controls. In most cases this is pretty informative and phrased in nice, novice-friendly terms, although in a few spots it could offer a little more than the wording on the button in question padded out to fill the space.

DOCUMENTATION

All these pages also link to the appropriate entry in the help system, via the Help button or F1, where more detailed information is provided in a reasonably clear and simple fashion, although most of it is laid out in order of the appearance of controls in the interfaces and does little more than explain their function.



A brief FAQ area includes some instructions on performing particular tasks, with most of the questions seeming rather simple, and a section entitled 'Viruses and more' contains a glossary of common types of malware, with a useful listing of the 'extended threat categories' for which detection can also be enabled.

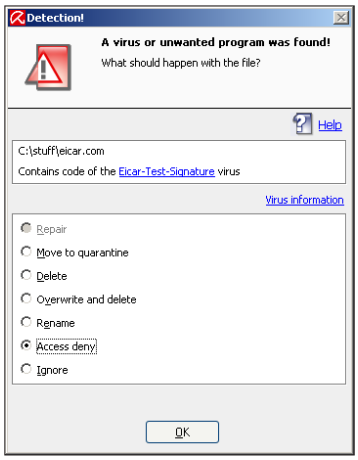
'Extended threat categories' include packed files, files with multiple extensions, jokes, suspicious applications, diallers (apparently a big issue in the German-speaking countries, with much detail provided on their use in Germany, Austria and Switzerland), and also games, with some jovial information added on the economic impact of gaming in the office.

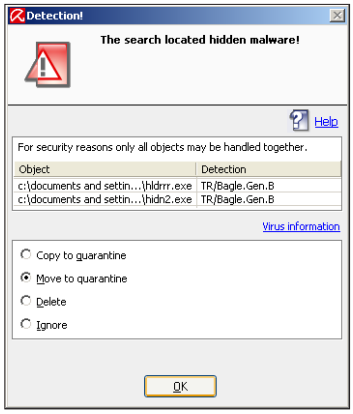
Throughout the Help area, the language shows clear signs of having been translated into English from a rather more formal tongue. Many sentences are long enough to upset the grammar monitors in *Microsoft Word* and the like and there is the occasional odd bit of syntax – for example, the term 'action for concerning files', which pops up quite often in the interface where other, simpler products might merely say 'action', is a regular eyebrow-raiser during VB100 testing.

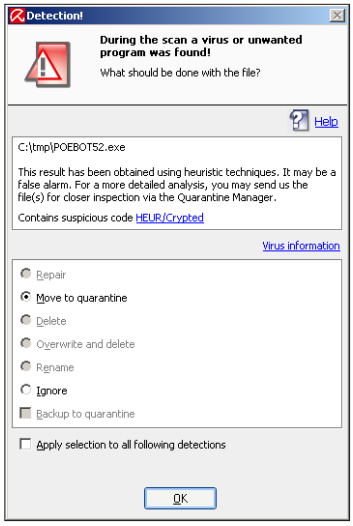
For those requiring more detailed guidance, a full manual is available as a PDF download. The cover is again adorned with photography of those red umbrellas getting out and about, the layout is more task-oriented, and the style and syntax is a lot clearer, although it seems a little flat and unadorned, lacking both illustrations on a larger scale than simple depictions of icons, and internal linking, with a lot of bare, unclickable instructions to 'see chapter…'.

Beyond these stylistic shortcomings, there is ample information in the various resources for most users to find their way around the deepest intricacies of the product.

MALWARE DETECTION AND PROTECTION



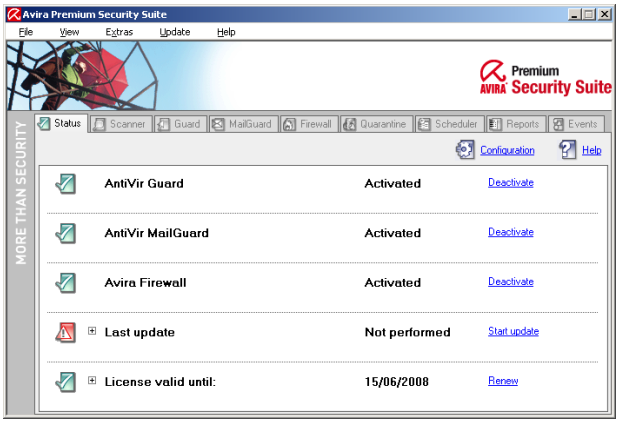




Avira has had some excellent scores in recent VB100 tests, with detection across the *VB* test sets at the very highest level, and its speed results have been similarly impressive. Both on-demand scanning times (over clean files) and the overheads imposed by the on-access component have been among the best in several comparatives – results which, again, have been confirmed by other testing bodies.

Running the product over the latest versions of the *VB* test sets showed no diminishing of the accuracy and efficiency of the product, or of the development and analysis teams behind it. Using a non-updated product, with virus definitions dating from mid-April, the product still managed some excellent detection of more recently acquired samples, including all but two files from the latest WildList (although that does itself also date from April); with heuristics turned up a notch, from the default 'medium' setting to 'high', only a tiny handful of items remained undetected.

Those that were still not spotted included a W32/Rbot variant included on the April WildList, but when this tried to connect outward the firewall component did, of course, pop



up a dialogue requesting permission. Whether the general user would be sufficiently savvy to know that 'Services.exe' is not necessarily a trustworthy item is perhaps a difficult point, but to at least some extent no item of malware tested entirely defied detection in one form or another.

Without the kinds of behavioural and intrusion-based detection gradually coming to the fore these days, the product relies on its advanced heuristics (dubbed 'AheAD'), which are designed to spot suspicious websites as well as files. This, combined with the efficiency and completeness of its signatures, seems to offer a very solid level of detection, and the product has also come top of the league in independent retrospective testing against unknown samples.

On the 'general' tab of the configuration interface a list of extra threat types can be selected. Several of these, such as adware/spyware, diallers, backdoors and phishing attempts, are active by default, but some more, including jokes, games, suspiciously packed files and suspected security risks, can be added for maximum paranoia.

With all these selected, some more items were detected from a stash of hacker tools and other dodgy files, as well as some games. However, I found I could still happily while away my important time with those perennial office favourites, *Minesweeper*, *Freecell* etc. (and also *Spectrum* classic *Manic Miner*). As far as the built-in *Windows* games go, it seems likely that the main situation in which such items would be unwanted is in a business environment, and one would hope that any corporate admin charged with ensuring the workforce keep their noses to the grindstone would be more than capable of removing these from desktops anyway.

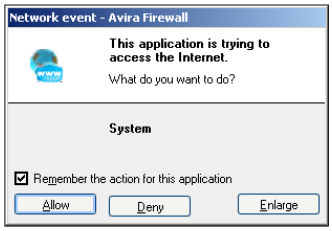
Rootkit scanning is a fairly new addition to the suite and, as malware grows in sophistication and the ability to remain quietly hidden on a system becomes a more important goal for its creators, it is a feature that is increasingly becoming a prerequisite of this type of product. Rootkit scanning is integrated into the scanning interface, with a simple entry in the list of available scans, and any drive on the system can be targeted. The scan is significantly faster than some of the rival products I have looked at, running through an entire system in only a few minutes. It also seems fairly effective, detecting a selection of hidden items and removing them, with a reboot required to complete cleaning.

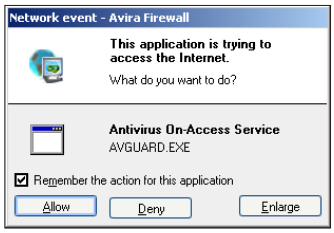
Scanning in general was very fast, and system overheads low, with similar figures to those recorded in the recent VB100 comparative (see *VB*, June 2007, p.10), where *AntiVir* put in an excellent performance. The interface offers options to adjust the priority of on-demand scanning, with the default set to give way to other processes if necessary.

OTHER FUNCTIONALITY

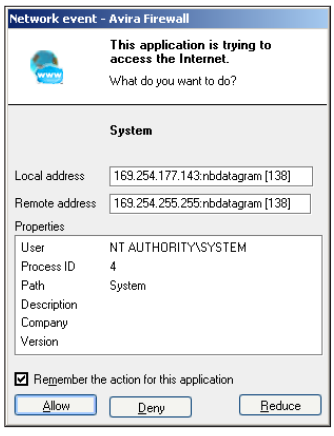
Beyond the basic detection of malware, the product also offers other security features, including some major modules covering access vectors for malware, system attacks and data leakage, as well as some less conspicuous extras.

Top of the list, and becoming something of a ubiquitous inclusion in such offerings, is the firewall, which features the standard set of controls on incoming connections and local applications sending outbound data.

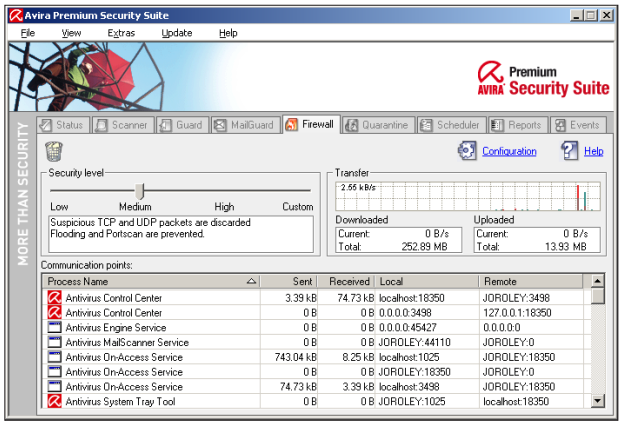




The initial settings seem fairly paranoid, with very little enabled by default. The first few minutes of operating on a normal web-connected machine involved numerous pop-ups requiring permission for a wide range of software wanting to contact outwards, including several involved in firing up some standard browsers and even a couple for other components of the *Avira* suite.



The pop-ups feature an 'enlarge' button, which



presents more detail on the individual item requesting permission, which is useful for more technically minded users, especially when the basic pop-up only says 'System'. For less well-informed users, as mentioned earlier, it seems likely that most such pop-ups will be OK'd without much thought – rather denting the efficacy of such protection – but this is a common problem wherever user understanding is required.

The design of the configuration system is better than many from a home-user perspective, with the often bewildering array of protocols and connection types hidden behind some more novice-friendly sliders and buttons. While this may sacrifice a little of the fine-tuning available in some products, it should cause minimal terror for less proficient users.

The email-filtering side of the software seems similarly simple to operate – a condensed version of the standard malware-handling controls is available, alongside some anti-spam settings. These are minimal in the extreme, with little more than options to switch logging and the use of real-time blacklists on and off, a choice of detail in messages appended to spam subject lines, and a button to purge the 'training' data, which is gathered from the user marking messages as false positives or negatives. Phishing attempts are also flagged.

Other extras include a searchable list of the available detection names, though without further information available offline this seems of little use compared to the excellent online databases provided. As mentioned earlier, password protection of the interface options is available at a very granular level.

CONCLUSIONS

For what it does, *Avira*'s suite is an excellent product. It may not be the prettiest product on the market, but where some rivals have invested in slick and shiny interfaces, *Avira* has focused on thorough coverage of malware and efficiency in the scanning engine, and for most users this is the most important aspect.

A security suite is not something most users expect to be playing with on a regular basis, and this is one that can safely be left to its own devices, shutting out malware without undue effort or system overhead. Defaults are sensible throughout, and a lack of glitter does not make the GUI any less easy to navigate.

While the firewall and spam filters were not given a thorough workout here, they seem perfectly adequate and offer sensible levels of configuration.

If there were any criticisms to be made beyond the mere superficial angle of appearance and presentation, it may be said that there is little here beyond the basics offered by almost all such suites. In the most recent VB100 comparative (see *VB*, June 2007, p.10), a group of new products came to our attention, introducing innovative ideas into the market with the likes of vulnerability watching and patch management already rolled into the products. Other suites include cryptography and data protection, data recovery, system cleaning and optimization, and the need for advanced IDS/IPS and behavioural detection of malware-like activity is also becoming ever greater.

All of this, of course, can easily be added by combining separate specialist products. For pure anti-malware protection with some of the best records of keeping signatures up to speed and some excellent heuristics, combined with highly impressive scanning times and minimal system overheads, *Avira*'s *Premium Security Suite* is pretty hard to beat.

With more functionality being added to its product sets all the time, *Avira* looks set to grow beyond its current localised focus and to become a very strong global player in the near future.

Technical details

Avira Premium Security Suite supports *Windows 2000* and later desktop platforms (including 32-bit *Windows Vista*), and requires a minimum specification of *Pentium*-class 133 MHz, with 40MB free hard drive space and 40MB free RAM. It was tested on:

AMD K6, 400Mhz, with 512MB RAM and dual 10GB hard disks, running *Microsoft Windows 2000 Professional Service Pack 4*.

Intel Pentium 4, 1.6Ghz, 512MB RAM, dual 20GB hard drives, 10/100 LAN connection, running *Windows XP Professional SP2*.

AMD Athlon64, 3800+ dual core, 1GB RAM, 40GB and 200GB hard drives, 10/100 LAN connection, running *Windows XP Professional SP2* (32-bit).

END NOTES & NEWS

The Information Security Asia 2007 Conference & Exhibition takes place on 10 and 11 July 2007 in Bangkok, Thailand. For details see http://www.informationsecurityasia.com/.

The International Conference on Human Aspects of Information Security & Assurance will be held 10–12 July 2007 in Plymouth, UK. The conference will focus on information security issues that relate to people. For more details see http://www.haisa.org/.

The 2nd conference on Advances in Computer Security and Forensics (ACSF) will take place 12–13 July 2007 in Liverpool, UK. For details see http://www.cms.livjm.ac.uk/acsf2/.

Black Hat USA 2007 Briefings & Training takes place 28 July to 2 August 2007 in Las Vegas, NV, USA. Registration is now open. All paying delegates also receive free admission to the DEFCON 15 conference, which takes place 3–5 August, also in Las Vegas. See http://www.blackhat.com/.

The 16th USENIX Security Symposium takes place 6–10 August 2007 in Boston, MA, USA. A training program will be followed by a two-and-a-half day technical program, which will include refereed papers, invited talks, work-in-progress reports, panel discussions, and birds-of-a-feather sessions. For details see http://www.usenix.org/events/sec07/.

HITBSecConf2007 Malaysia will be held 3–6 September 2007 in Kuala Lumpur, Malaysia. For more details see http://conference.hackinthebox.org/.

SecureDüsseldorf takes place 11 September 2007 in Düsseldorf, Germany. The conference will focus on privacy issues. For further information and registration see https://www.isc2.org/.

Infosecurity New York will be held 11–12 September 2007 in New York, NY, USA. For details see http://www.infosecurityevent.com/.

The 17th International VB Conference, VB2007, takes place 19–21 September 2007 in Vienna, Austria. For the full conference programme including abstracts for all papers and online registration see http://www.virusbtn.com/conference/.

COSAC 2007, the 14th International Computer Security Forum, will take place 23–27 September 2007 in Naas, Republic of Ireland. See http://www.cosac.net/.

The SecureLondon business continuity planning 101 workshop will be held 2 October 2007 in London, UK. For further information and registration see https://www.isc2.org/.

The APWG eCrime Researchers Summit takes place 4–5 October 2007 in Pittsburgh, PA, USA. Academic researchers, security practitioners, and law enforcement representatives will meet to discuss all aspects of electronic crime and ways to combat it. For more information see http://www.antiphishing.org/ecrimeresearch/index.html.

RSA Conference Europe 2007 takes place 22–24 October 2007 in London, UK. See http://www.rsaconference.com/2007/europe/.

The CSI 34th Annual Computer Security Conference will be held 5–7 November 2007 in Washington, D.C., USA. The conference program and registration will be available in August. See http://www.csi34th.com/.

E-Security 2007 Expo & Forum will be held 20–22 November 2007 in Kuala Lumpur, Malaysia. For event details and registration see http://www.esecurity2007.com/.

AVAR 2007 will take place 29–30 November 2007 in Seoul, Korea. This year's conference marks the 10th anniversary of the Association of Anti Virus Asia Researchers (AVAR). Inquiries relating to any form of participation should be sent to avar2007@aavar.org.

RSA Conference 2008 takes place 7–11 April 2008 in San Francisco, CA, USA. A call for speakers at the event closes on 27 July 2007. Online registration will be available from 1 September 2007. See http://www.rsaconference.com/2008/US/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues):

- Single user: $175
- Corporate (turnover < $10 million): $500
- Corporate (turnover < $100 million): $1,000
- Corporate (turnover > $100 million): $2,000
- *Bona fide* charities and educational institutions: $175
- Public libraries and government organizations: $500

Corporate rates include a licence for intranet publication.

See http://www.virusbtn.com/virusbulletin/subscriptions/ for subscription terms and conditions.

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

SPAMMER OFFERS NEW SERVICE

Wily email marketer/spammer (depending on your viewpoint) David Linhardt, famed for suing UK spam-fighting organization *Spamhaus*, has attempted to turn the court order he was awarded against the organization into a money-making opportunity, according to spam-watchers.

In September 2006 an Illinois court forbade *Spamhaus* from listing Linhardt's company, *e360 Insight*, and any of its affiliates as spammers. Now, Linhardt is offering a service to companies blacklisted on the *Spamhaus Block List*, in which he will attempt to get them removed from the list by claiming they are affiliates of *e360*.

The first customer appears to have been online marketing firm *Virtumundo.* In a letter to *Spamhaus* last month, *e360*'s lawyer advised that 'effective immediately, *Virtumundo* is a customer of and doing business with *e360*'. The letter went on to demand that *Spamhaus* remove a blocklisting in accordance with the injunction. *Spamhaus* has refused to lift the blocklisting, stating that, as far as it is concerned, 'customers who form nothing more than a contractual relationship with *e360* after the Court's injunction order was entered are not within the scope of that order'.

Linhardt has been using sponsored *Google* links to advertise his services.

EVENTS

CEAS 2007 takes place 2–3 August 2007 in Mountain View, CA, USA. For details see http://www.ceas.cc/.

The 11th general meeting of the Messaging Anti-Abuse Working Group (MAAWG) will take place 8–10 October 2007 in Washington D.C., USA. See http://www.maawg.org/.

TREC 2007 will be held 6–9 November 2007 at NIST, MD, USA. See http://plg.uwaterloo.ca/~gvcormac/spam.

CONFERENCE REPORT

EU SPAM SYMPOSIUM

Sorin Mustaca
Avira, Germany

The second EU Spam Symposium was held in Vienna at the end of May. The conference was held over two days, with 10 presentations on the first day, and an open discussion forum on the second.

THE PAPERS

Kurt Einzinger, general manager of the Austrian Internet Service Provider Association (ISPA), opened the conference with a speech about how the Association works and how it plans to fight on multiple fronts against organized e-crime. The ISPA has 204 members which include: ISPs, companies with an online presence, maintainers of real-time blacklists and others. Kurt explained that spam represents a major problem for ISPs, causing infrastructure overloading and traffic bottlenecks and requiring a lot of manpower to maintain the systems, all of which incur financial losses for the companies.

Jason Steer, Product Manager at *IronPort Systems*, one of the main sponsors of the conference, gave an interesting presentation entitled 'Deconstructing a 20-billion message spam attack'. Jason talked about a series of spam waves that were sent in May 2006 with different variations in order to prevent their detection. The waves consisted of 20 billion messages sent in more than 2,000 unique spam mutations (one every 12 minutes) and through 1,500 unique domains.

Jason described an experiment in which he and his colleagues bought some 'Viagra' from an online meds shop. When they received the product, expert analysis showed that it was fake. He concluded with a view shared by many in the anti-spam industry: that the real cause of the spam problem is not the spammer, but the buyer.

The next talk was a joint presentation by Richard Cox and Carel van Straten of *SpamHaus*, entitled 'How do we balance the needs of privacy with the need to counter spam?'. Richard spoke about the well-known *SpamHaus Project* and described why he feels the internet is worth fighting for. I enjoyed the fact that Richard referred to the

spammers as conventional criminals, and he called for them to be treated as such.

Carel described how spammers use decentralized bot networks and dropped malware that performs RBL lookups in order to make their activities more efficient. The spammers manage to escape law enforcement by distributing their bots, control centres, webservers, proxies etc. in various countries across the globe – preferably in those without anti-spam laws.

The conclusions of the talk were: a small number of ISPs are causing a significant amount of the damage by not having clear usage policies and if we want to start fixing the problem, the ISPs should be the first to take action.

There were two academic papers: one by John Aycock from the University of Calgary, Canada, and the other by Richard Clayton from the University of Cambridge in the UK. John analysed what a spammer or phisher would do with a botnet of a thousand or a million machines. Most people would assume that they would simply send a lot of spam, but John showed us that they can do much more. He described in his paper how the distributed computing power of so many hosts could be used easily to break strong encryption which we take to be unbreakable.

Richard talked about detecting email spam in sampled traffic data while it passes through major internet exchange points (IXP) sited in the UK. These servers are handling more than 100Gbit/s mail traffic. By analysing packet patterns, basic headers and the time at which the messages were sent, an ISP can monitor the emails that enter or leave its network.

The next two presentations were about the laws that are designed to define and control spam in the EU and Mexico. Max Mosing, a lawyer in an Austrian law firm, talked about the 'ups and downs in the history of EU spam regulations'. Despite being rather long, the presentation was very interesting. I don't think that many people realise how hard it is to get a simple (in our eyes) law approved and then applied in 12 different member states. The EU struggled first to define various forms of spam from a legal point of view and then successively, for eight years, issued and refined various regulations to cover all the holes left by the previous ones.

Cristos Velasco, founder of the North American Consumer Project on Electronic Commerce, was the second lawyer to speak, presenting the struggle of various organizations and the government of Mexico against spam and phishing. Even though the number of internet users in Mexico is rising rapidly (there are currently more than 20 million), there are not as many phishing attacks in Mexico as in other countries experiencing a similar growth.

John Graham-Cumming's presentation was called 'So, will filters kill spam?'. He discussed how the spammers keep their techniques up to date in order to bypass the filters. The main idea of John's presentation was that spammers innovate constantly by testing their emails against filters, against webmail services and … by learning from spam conferences.

John also reiterated what we had previously heard in Jason Steer's presentation and will hear again: spam works because people buy the products advertised in it. His conclusion was that spammers will continue to keep pace with improvements in spam filters. As the internet infrastructure improves, so spammers will be able to send even more spam.

The next speaker was Sven Karge from *eco,* a German organization that protects the interests of companies with an internet presence in Germany. Sven talked about a European initiative in which information about spam is collected from the EU member states with the purpose of stopping the senders of these messages. The project name is SpotSpam.net and a detailed description of what it does can be found at http://www.spotspam.net/.

Like last year, the final speaker of the conference was Spammer X, a retired spammer who has also written a book about his 'work'. Spammer X gave an entertaining presentation about current spam trends and shared his thoughts about what the spam of the future might bring: VoIP spam and video spam. He confirmed that the only solution to spam is to stop people buying the advertised products, although he also listed a number of steps that will help to reduce spam including: securing computers, and sending complaints to law enforcement agencies, to anti-spam organizations and to ISPs.

On the second day of the conference an open discussion was held with panel members Richard Cox, Cristos Velasco, John Aycock, Richard Clayton, Carel van Straten and Spammer X. A lot of topics were discussed, ranging from spam and phishing detection to the possibilities and challenges brought by anti-spam laws.

CONCLUSIONS

It was good to see so many experts from so many different fields all brought together because of the same problem: spam. Like last year, though, I was disappointed by the fact that there were no presentations on the subject of phishing. However, the organizers have promised that next year's symposium will include such material.

Webcasts of the presentations are available at: http://www.spamsymposium.eu/archivewebcast.htm.

FEATURE

FRANCE'S ANTI-SPAM DATABASE

John Graham-Cumming
Independent researcher, France

On 10 May 2007 the French national online anti-spam platform, *Signal Spam*, was launched.

The service allows any French resident to send any spam they receive to *Signal Spam* for automatic handling. At the time of writing over 24,000 people have signed up to use the service and over 1 million messages have been received by *Signal Spam*, with an average of 30,000 messages received per day during the first 32 days of operation (happily the infrastructure of *Signal Spam* was built to handle 1 million messages per day).

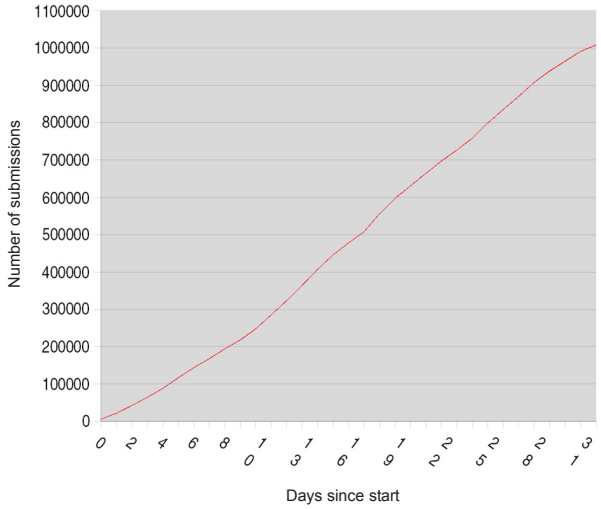


Figure 1: Total number of messages sent to Signal Spam in the first 32 days of operation.

Signal Spam is a non-profit organization (in French law it is 'une association de loi 1901'), created as a partnership between the French government (through the Direction du développement des médias, which falls under the purview of the Prime Minister), a number of other French public bodies (such as the French data protection office: the Commission Nationale de l'Informatique et des Libertés), industry groups (such as the French ISP association: the Association des Fournisseurs d'Accès et de Services Internet, and French direct marketing groups including the Syndicat National de la Communication Directe) and private industry (including founding partner *Microsoft*). It receives funding from the French government as well as from member groups that join the association.

For individuals *Signal Spam* is entirely free of charge: the user simply visits http://www.signal-spam.fr/ and signs up for a free account. Users can opt to provide full contact information if they are willing to be contacted in case of legal proceedings concerning messages they have sent in. However, the minimal user account requires just a username, password and a valid email address (which need not be the one at which the user is receiving spam).

Once signed up there are two ways to send a message to *Signal Spam*: copy and paste via a web form or through a plug-in for the email client. Since full email headers are vital for the analysis of any message there is no message-forwarding option, and the preferred method is the plug-in.

OPEN SOURCE AND AN OPEN API

Plug-ins are currently available for *Mozilla Thunderbird 2.0* and *Microsoft Outlook 2003* and *2007*.

At the insistence of *Signal Spam* the source code for the plug-ins is open source (the *Thunderbird* plug-in is released under MPL, GPL or LGPL; the *Outlook* plug-ins are released under the BSD licence) and the plug-in API's specification is freely available (in French: https://www.signal-spam.fr/index.php/frontend/extensions/api_de_signalement).

The API itself is a simple REST interface running over HTTPS. The plug-in makes an HTTPS connection to http://www.signal-spam.fr/ using the path /api/signaler. The username and password created by the user are sent using basic authentication. The message being sent is base-64 encoded and sent as a simple POST as if it were a standard HTML form element with the name 'message'.

The API replies with the HTTP return code 202 Accepted if the message has successfully been received, 400 Bad Request if there has been a problem with the request itself or another standard HTTP error (signalling a bad username/password for example).

Users are limited to sending a maximum of 500 messages per day. The openness of the API has already spawned a couple of third-party interfaces with a shell script and a mutt script. All the plug-ins and scripts are available at: https://www.signal-spam.fr/index.php/frontend/extensions; anyone creating their own method of signalling messages is encouraged to email support@signal-spam.fr so that it can be included on that page.

Currently, the *Microsoft Outlook* plug-in is the most popular method for sending messages to *Signal Spam* (accounting for almost 48% of the messages), followed by the web form (31.79%).

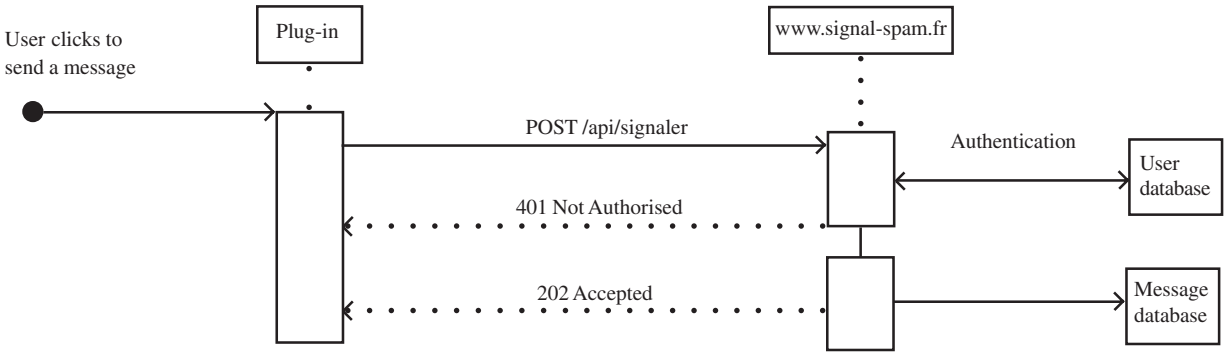


Figure 2: Plug-in interaction with Signal Spam.

MESSAGE ANALYSIS

Once a message is received by *Signal Spam* it is transferred across a secure link to a separate and isolated machine where it is stored for analysis. An automatic analysis process runs constantly, picking up new messages and performing the following sequence of steps:

1. Extraction of the following email headers: From, Subject, User-Agent/X-Mailer, Return-Path and Date. These are stored in the message database for fast searching.
2. Discovery of the injection IP address. This is the most complex part of the process, and involves walking the chain of Received headers and matching them up to look for the injection IP address and evidence of forgery.
3. Mapping of the injection IP address to the network AS number and the name of the service provider responsible for the AS. These details are also stored in the database. The AS information is also used to determine the source country for the message.
4. URL extraction. All URLs present in the message are extracted and stored in the database for searching and reporting purposes.
5. Fingerprint creation. The message is fingerprinted using the Vipul's Razor Ephemeral and Whiplash fingerprints. The actual fingerprint mechanism is extensible and other algorithms can be added as needed.

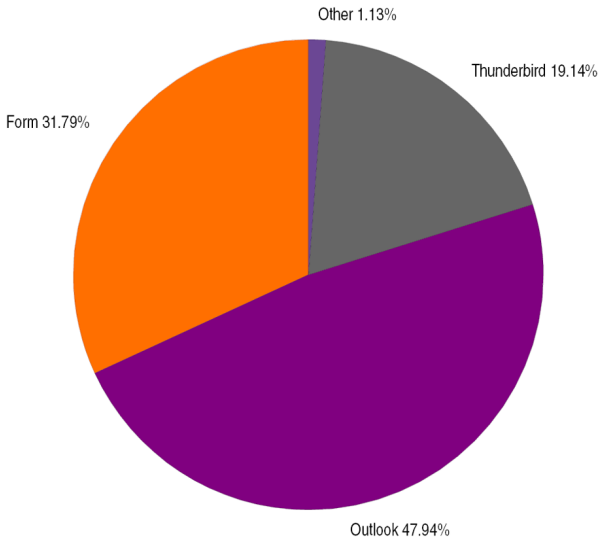


Figure 3: Percentage of messages sent to Signal Spam by method.

Currently the database shows that the top ten message-sending countries (in messages signalled to *Signal Spam*) are: USA, France, China, Germany, South Korea, Poland, Russia, Brazil, UK and Israel.

AUTOMATED ABUSE REPORTS

If the message originated inside France (from a French ISP or other entity that manages a block of IP space) then it's possible for *Signal Spam* to send them automatically an anonymized report of the offending message. Any French entity that wishes to take part must join *Signal Spam* and provide information about the AS or IP address ranges that they control, along with an email address to receive abuse reports.

Abuse reports are generated automatically when the AS or IP address range matches a registered entity in the *Signal Spam* database and are sent using the ARF (Abuse Reporting Format, see http://www.mipassoc.org/arf/) specification. Prior to inclusion in the ARF report the message is anonymized by removing the headers To, Cc, Bcc, Apparently-To, Delivered-To, In-Reply-To, References, Reply-To and by removing email addresses

from any Received header. The following shows an example ARF message as sent by *Signal Spam*:

```
From: <1234-abuse=fai.fr@alerte.signal-spam.fr>
Date: Thu, 8 Mar 2007 17:40:36 EDT
Subject: FW: Earn money
To: <abuse@fai.fr>
MIME-Version:1.0
Content-Type: multipart/report;
   report-type=feedback-report;
boundary="part1_13d.2e68ed54_boundary"

-part1_13d.2e68ed54_boundary
Content-Type: text/plain; charset="US-ASCII"
Content-Transfer-Encoding: 7bit

This is an email abuse report for an email message
received from IP 10.67.41.167 on Thu, 8 Mar 2007
14:00:00 EDT.

-part1_13d.2e68ed54_boundary
Content-Type:message/feedback-report

Feedback-Type:abuse
User-Agent:SignalSpam/0.1
Version: 0.1
Original-Mail-From: <somespammer@example.net>
Received-Date: Thu, 8 Mar 2007 14:00:00 EDT
Source-IP: 10.67.41.167
Reported-Domain: example.net
Reported-Uri: http://example.net/earn_money.html
Reported-Uri:mailto:user@example.com

-part1_13d.2e68ed54_boundary
Content-Type: message/rfc822
Content-Disposition: inline

From: <somespammer@example.net>
Received: from mailserver.example.net
   (mailserver.example.net [10.67.41.167]) by
   example.com with ESMTP id M63d4137594e46; Thu, 08
   Mar 2005 14:00:00 -0400
To: <Undisclosed Recipients>
Subject: Earn money
MIME-Version: 1.0
Content-type: text/plain
Message-ID: 8787KJKJ3K4J3K4J3K4J3.mail@example.net
Date: Thu, 02 Sep 2004 12:31:03 -0500

Spam Spam Spam
Spam Spam Spam
--part1_13d.2e68ed54_boundary--
```

UNSUBSCRIBE ASSISTANCE

Another automatic feature of *Signal Spam* is identifying messages that were from genuine e-marketers that are not spam and informing the user of the correct unsubscribe procedure. Since it's expected that many users will signal messages that are from legitimate e-marketers, *Signal Spam* built a system that identifies these messages and replies automatically to the user.

Any French marketer can join *Signal Spam* and provide details of their newsletters and marketing mails to *Signal Spam* along with a message on how to unsubscribe from each of them. When *Signal Spam* identifies a message from one of these partners it replies to the user who sent in the message with the appropriate information to help them unsubscribe.

Signalement avec id [redacted]

Les données complètes

ID	[redacted]
AS de l'émetteur	15742
Contact possible	oui
De	Harold Taylor <ffraid@ide.com>
Domaine de l'émetteur	ns.telesun.net
Envelope Sender	
Fanions	EXTRAIT_CHAMPS IP_EMETTEUR_VALIDE AS_IDENTIFIE EMPREINTES_VALIDES URLS_VALIDES MESSAGE_TRAITE
Horodatage du message	2007-06-11 16:02:54
Horodatage du signalement	2007-06-11 14:03:20
ID de l'utilisateur	[redacted]
IP de l'émetteur	217.117.78.41
Menace	inconnu
Pays de l'émetteur	UA
Prestataire de l'émetteur	PRIVATONLINE Private Online Autonomous System
Sujet	Achieve the feeling of complete ecstasy!
User-Agent/X-Mailer	The Bat! (v3.5) Home
Agent	formulaire

Figure 4: A message analysed by Signal Spam.

MANAGEMENT BACKEND

Most of the *Signal Spam* website and software is invisible to the public and consists of an administration interface for the creation of reports, database searching and an interface to other parts of the French government (for example, the French Gendarmerie will have access when investigating cybercrimes).

At the most basic level an individual message (known as a 'signalement') can be viewed. Figure 4 shows how a message appears in the interface once it has passed through automatic analysis. In addition to the fields shown in the image the full message can be viewed, as well as the extracted URLs and the message fingerprints.

CONCLUSION

It's very early days for *Signal Spam*, but the system is up and running and getting a lot of publicity in France, and *Signal Spam* has indicated that it is interested in sharing the entire system with other countries so that they can set up their own spam databases.

But the success of the system in helping in France's fight against spam remains to be seen.

John Graham-Cumming will be presenting a paper entitled 'The Spammers' Compendium: five years on', at VB2007 in Vienna (19–21 September). The full programme and details of how to register for the conference can be found at http://www.virusbtn.com/conference/vb2007/.