SEPTEMBER 2007

virus BULLETIN

Fighting malware and spam

CONTENTS

ISSN 1749-7027

IN THIS ISSUE

RISE OF THE BOTS

Having spent a lot of time thoroughly analysing how bots work – studying the overall bot 'ecosystem' as well as the individual files – Luis Corrons has noted that they have achieved a certain level of autonomy, almost to the extent that they have a life of their own. We're still a little way from Skynet, but it's only a matter of time…

page 4

MEN IN WHITE

Bit9 CSO Ian Poynter explores some of the questions raised by Vesselin Bontchev last month about the validity of whitelisting as an endpoint security technology.

page 6

BITDEFENDER TOTAL SECURITY

John Hawes takes an in-depth look at *BitDefender*'s latest home-user product.

page 17

vbSpam supplement

This month: anti-spam news & events, and Tobias Eggendorfer describes how both SMTP and HTTP tar pits can be used to help get rid of spam.

virus BULLETIN COMMENT

'Both the disease and the cure now differ significantly from their original forms.'

David Emm, Kaspersky Lab, UK

AV IS ALIVE AND WELL

I recently saw an article announcing the slow death of AV technology. It set me thinking about how 'anti-virus' solutions have evolved to deal with the changing nature of malicious code.

Threats are more complex and numerous than ever before. Much of today's malicious code is designed specifically to hijack computers and make money illegally. Today's attacks are rapid and they can be as wide-reaching or selective as cyber criminals desire. Malicious code can be embedded in email, injected into fake software packs, or placed on 'grey-zone' web pages for download by a trojan installed on an infected machine.

In any field of human activity, each generation learns from its predecessors, continues to implement proven methods, and also tries to break new ground. This is also true of virus writers; successive waves of malicious code have redefined the threat landscape. Security solutions have evolved to match new generation threats and both the disease and the cure now differ significantly from their original forms.

Initially, viruses were relatively slow-spreading. Although a significant number of outbreaks were caused by file infectors, boot sector viruses and multipartite viruses were the main threat up to 1995. The use of stealth techniques to hide infection and encrypted code to hinder analysis and detection also evolved during this period.

To start with, anti-virus programs were on-demand only. Due to the slow spread of viruses and the slow increase in the number of new viruses, scanners were used to detect and remove infected code. In many cases, companies wouldn't install anti-virus programs on individual machines (although attitudes tended to change once a company got hit by a virus). In addition to regular scanning, a stand-alone machine was often used to screen incoming floppy disks. It was only once the virus count reached 300 (which seemed a lot at the time) that real-time protection was developed and implemented. Anti-virus programs were updated just quarterly, or monthly by the 'paranoid', with updates delivered on floppy disks.

Anti-virus programs were mainly signature-based. Some employed behavioural analysis; however the nature and scale of the malware threat did not justify mainstream deployment of these technologies.

Increased use of the Internet and of email changed things significantly. First there were macro viruses, which spread more quickly than preceding viruses by 'piggybacking' data files (primarily documents) on email. Then came email worms: they hijacked email to distribute their code proactively, further speeding up the infection process. The problem of spam also emerged.

In an effort to stem infections before they reached employees, the anti-virus function was shifted from desktops to email servers and Internet gateways. New threats spreading at 'Internet speed', a growing number of global epidemics and an increasing number of threats exploiting application vulnerabilities also forced AV vendors to respond more rapidly to new threats. Weekly and then daily (or even hourly) updates became the norm.

Growing concerns about the potential time lapse between the appearance of a new exploit and the means to block it fuelled the development of proactive technologies and their integration into Internet security solutions that exceeded the scope of traditional anti-virus programs. The use of proactive technologies (e.g. heuristic and generic detection) dates from the early to mid-1990s. However, the scope of anti-virus programs has been further extended by integration of personal firewall, intrusion prevention and behavioural analysis technologies. AV today is much more than just AV.

In the early days of viruses, no one anticipated the quantity or variety of malicious programs that exist today. Each wave of malware development brought new challenges that required a change to existing solutions, the development of new solutions or the integration of non-AV technologies. The threat landscape is radically different to that of 20 years ago, and so are today's security solutions. Early AV solutions look one-dimensional compared with the holistic solutions offered by today's security software providers. Signature scanning remains, but in the context of a wider strategy. There's no question that AV is alive and well.

NEWS

UNSAFE COMPUTING IN ABUNDANCE

Last month saw a flurry of reports and statistics on unsafe computing practices. To kick off, almost a quarter of Internet-connected users in the UK believe they have suffered from a virus attack in the past year, according to figures released by the Office of National Statistics. In a survey, 23% of respondents claimed that their computer had been affected by a virus in the past 12 months, and just under half of the respondents admitted to rarely or never backing up the data on their PCs.

Moving on to corporate security, *Panda Security* has revealed that 59% of companies that scanned between 20 and 30,000 PCs using its online malware audit service *Malware Radar* were harbouring active malware on their systems. For companies with smaller networks (10 to 19 machines) the figures were barely any better, with active malware being found in 47% of the companies using the audit service. The number of infected companies with fewer than 10 machines fell to 37% for those with between five and nine machines, and 35% for those with fewer than four.

Staying with corporate security, a study by *Cisco* and the *National Cyber Security Alliance* (*NCSA*) has found that mobile workers are not being diligent about security. The study examined the behaviour of workers using smartphones, PDAs, laptops and other mobile devices in the US, UK, Germany, China, India, South Korea and Singapore. Of those surveyed, 73% confessed that they were not always aware of the security risks involved in mobile working or of the best practices for secure mobile working. Reasons given for the failure to comply with security best practices (when aware of them) included being too busy, not considering it a priority, and considering security to be the responsibility of the IT department.

The survey also showed that many mobile workers hook up to unauthorized wireless connections – hopping onto unsecured wireless networks in their neighbourhood or in public areas. Reasons given by some of the 54% of Chinese mobile workers and 20% of their US counterparts who admitted to using unauthorized connections included their own connections not working or simply not wanting to pay for their own connections. An alarming number of the mobile workers surveyed (44%) said they regularly open emails and attachments from unknown sources, with many saying that the small screen size of PDAs and smartphones makes it difficult to identify suspicious emails.

Of the US respondents a disappointing 39% claimed never to have received any security training, while 14% couldn't recall whether they had received training or not. Following the study NCSA executive director Ron Teixeira has called for mobile security awareness to be made a top priority in businesses worldwide.

Prevalence Table – July 2007

Virus	Type	Incidents	Reports
W32/Netsky	Worm	1,979,252	28.88%
W32/Bagle	Worm	1,166,324	17.02%
W32/Mytob	Worm	1,006,519	14.69%
W32/MyWife	Worm	441,315	6.44%
W32/Virut	File	330,992	4.83%
W32/Sober	Worm	325,116	4.74%
W32/Stration	Worm	274,814	4.01%
W32/Mydoom	Worm	210,386	3.07%
W32/Lovgate	Worm	202,631	2.96%
W32/Sality	File	190,353	2.78%
W32/Zafi	Worm	187,064	2.73%
W32/Bagz	Worm	80,393	1.17%
W32/Rontokbro	Worm	41,001	0.60%
W32/Parite	File	36,885	0.54%
W32/Rjump	Worm	32,899	0.48%
W32/Funlove	File	30,778	0.45%
W32/VB	Worm	28,181	0.41%
W32/Jeefo	File	24,376	0.36%
W32/Looked	File	22,396	0.33%
W32/Klez	File	20,634	0.30%
W32/Grum	Worm	16,809	0.25%
VBS/Small	Worm	16,764	0.24%
W32/SirCam	Worm	16,609	0.24%
W32/Small	File	13,332	0.19%
W32/CTX	File	12,862	0.19%
W32/Fujacks	File	10,300	0.15%
VBS/Butsur	Script	9,937	0.14%
W32/Mabutu	Worm	9,005	0.13%
W32/Tenga	Worm	8,193	0.12%
W32/Womble	Worm	8,127	0.12%
W32/Nahata	File	7,596	0.11%
W32/Yaha	File	6,674	0.10%
Others[1]		85,358	1.25%
Total		6,853,875	100%

[1]The Prevalence Table includes a total of 85,358 reports across 223 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

FEATURE

THE LIFE CYCLE OF BOTS

Luis Corrons
Panda Security, Spain

I have spent a lot of time thoroughly analysing how bots work – studying the overall bot 'ecosystem' as well as the individual files. It is curious to see how they have achieved a certain level of autonomy, in such a way that they almost have a life of their own. We are still a little way from Skynet [1], but it is only a matter of time…

CONCEPTION

As with all life cycles, the starting point is conception. One of the most effective and effortless methods of creating a multi-functioning bot is to use a framework. We must also bear in mind that bot source code is available on the Internet, so anyone with a basic knowledge of computers can make a bot. Of course, there is always the option of programming your own bot from scratch, but this is rarely done these days.

Once finished, the bot creator usually distributes and sells their 'creature' as if it were any other customized software program. We could compare buying a bot with buying a pet, where the customer can buy the pet that best suits their needs. But, of course, there are many more options with bots than with animal species. I won't list them all, but some of the most typical options are the following:

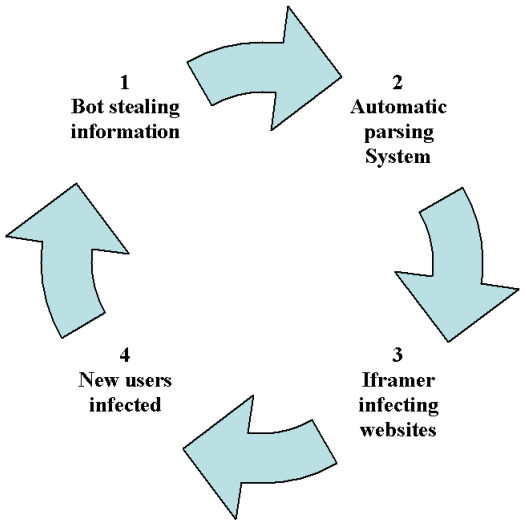
- Sending spam.
- Launching DoS attacks.
- Stealing information (banking information, information from e-shops, email accounts, all type of passwords, etc.).

Of course, the bots are provided with updates and guarantees that when they are sold, no signature-based anti-virus program will be able to detect them.

GROWTH/REPRODUCTION

Now we have the little baby, it's time for it to grow up and reproduce itself. First, the creature needs to infect at least a small number of computers. There are many ways to do this, though the most common methods involve infecting websites with kits similar to MPack [2], or sending spam.

Having infected a few PCs, the bot creator only needs to sit back and wait for the new ecosystem to develop on its own. This is one of the most fascinating phases in the cycle of malware. Some hackers have managed to create complex systems from which they get feedback. Let's explain it step by step:



1. The trojans (bots) start working by stealing the user's data and uploading the logs to the corresponding server.
2. In the server, the logs are parsed in search of ftp accounts, storing the data in a text file.
3. An 'iframer' application accesses the file containing the information from the stolen ftp accounts and starts accessing all of them automatically. The application searches for certain directories and modifies the pages hosted there with iframe tags that point to an infecting server (with MPack, WebAttacker [3] or any other similar system). Tools such as IcePack [4] incorporate the functionality of the infecting server, ftp account checker and iframer in the same package.
4. The infection system only has to wait until users visit the (legitimate) websites whose pages have been modified. The user is infected with a trojan that is small (a few kb) and silent (it displays no messages), and whose sole function is to download and install more malware. The malware downloaded may be the trojan downloader itself. We have seen in the installed servers systems that allow the trojan to change its shape every few seconds or even a different one for each computer, in order to avoid anti-virus signature detection and ensure the longevity of the trojan.
5. The data of all the newly infected users is uploaded to the server, where it is processed to extract new ftp accounts in order to access websites and infect them, thus starting the cycle again.

DEATH

There are a number of different factors that can lead to the death of bots:

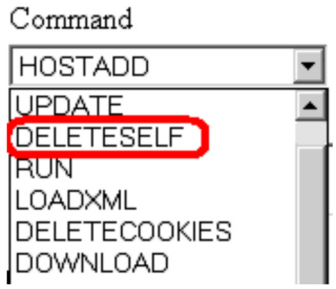
Predators

Firstly, we must mention anti-malware products, especially those which include proactive technologies rather than relying on signatures to detect the bot (detection is thus harder for malware to evade).

Secondly, we must mention other strains of malware. In order to protect themselves from detection by security programs, it has become common over the last several years for certain strains of malware to monitor systems for the existence of certain other malware. If they find any, they will delete them from the system by ending their processes, deleting their files and registry entries, etc. These strains of malware have now evolved further and we have started to see them monitoring systems for the existence of (and removing) other rival worms/trojans, and even older versions of themselves. By doing this the malware can minimize the instability of the system and the resource consumption of the computer, as well as getting rivals out of the way. Only the strongest can survive.

Filicide

A curious option we have seen in many bot control panels is for the hacker to send an instruction of self-deletion to all the bots, which means the end of their existence.



Overpopulation

There is a theory that overpopulation of the planet may lead mankind to his own extinction. But let's not get our hopes up over the same being true for bots: the critical factor in overpopulation is not so much population density as the availability of resources for the population – and for this reason we must rule out this option for bots.

Log poisoning

This is a technique I have seen carried out by some banks when their servers have been under attack. Log poisoning involves the infection of several computers, which start flooding the server with false data. The poisoning is smooth but ineffective. The processing of the log data is automated and, therefore, it will only lead to an increase in information storage and processing.

In any case, the problem with each of the previous points is that we are talking about the death of the trojan. Let's look at the situation from the perspective of the whole ecosystem. Although anti-virus programs detect and delete the trojan, the self-updating trojan changes constantly, rather like the mythological hydra [5] – when Heracles cut off its head it grew back two more.

The best way to shut this ecosystem down is to shut down the servers. While this may sound simple, it can be very difficult to carry out. These servers are usually hosted in countries like China or Russia, where taking them down is more than an awkward task; it's almost impossible. The server hosting the infection kit is usually different from the server to which data is sent and there may even be multiple servers.

Furthermore, there's no use in just closing the server to which the trojan sends the data (which is usually the one which hosts the control panel), as the users are still infected, and in many cases downloaders are active on their machines. This means that the hacker can easily install a new version of the trojan that sends the data to a new server.

What else can be done? The most effective solution would be to kill (metaphorically speaking) the bot's creator, which bring us to the final point:

Law enforcement

I know that this subject will bring a smile to many readers' faces (as it does to mine), as the general feeling is that those behind crimeware are light years away from being tracked down by law enforcement agencies. This bears some truth, though the aim of this article is not to go into the situation in depth. Nevertheless, many people agree that some law enforcement provided with the necessary means, international collaboration agreements and proper legislation could be the best solution. However, we should be mindful of the fact that we will never be able to terminate all bot ecosystems and other malware completely, just as the police will never be able to stamp out all crime.

REFERENCES

[1] http://en.wikipedia.org/wiki/Skynet_%28fictional%29.

[2] http://pandalabs.pandasecurity.com/MPack-uncovered_2100_.aspx.

[3] http://www.websense.com/securitylabs/blog/blog.php?BlogID=94.

[4] http://pandalabs.pandasecurity.com/Ice_2800_Pack_2900_-for-the-summer.aspx.

[5] http://en.wikipedia.org/wiki/Lernaean_Hydra.

OPINION

FRIENDLY WHITELISTING AND OTHER INNOVATIONS: A RESPONSE

Ian Poynter
Bit9, Inc., USA

In the August 2007 issue of *Virus Bulletin*, Dr Vesselin Bontchev wrote a thought-provoking opinion piece about whitelisting as an endpoint security technology (see *VB*, August 2007, p.4). As Chief Security Officer for *Bit9*, a company highlighted by Dr Bontchev in his article, I feel compelled to continue this conversation and explore some of the questions he has raised.

EVOLUTION OF WHITELISTING

Dr Bontchev's piece is well reasoned, and I agree with most of his insights. However, these points apply to the whitelisting technology of five to 10 years ago – not the quite different whitelisting technology of today. In the scenarios he describes, Dr Bontchev accurately assesses many of the challenges of a whitelisting security model – ranging from managing the whitelist, to scalability concerns, to user and political issues.

But, just as anti-malware technology has evolved dramatically over the last decade, so too has whitelisting. In 2003, *Bit9* obtained a two-million-dollar grant from the National Institute of Standards and Technology's Advanced Technology Program (NIST ATP) to research and solve exactly the sorts of difficulties Dr Bontchev identifies. That project, and subsequent research, led to many surprising changes in the capabilities of whitelisting technologies that have already simplified whitelisting and endpoint security in general.

THE FUNDAMENTAL PROBLEM WITH TODAY'S SECURITY PRACTICE

Let's start with a fundamental reality that Dr Bontchev describes – anti-virus scanners are clearly the 'weakest line of defence', yet users are not implementing more advanced techniques because they are too complex to keep up to date.

I have found this to be true first-hand throughout my career in the security industry. But it is not the fault of the user. I believe it is our responsibility – our obligation as security researchers – to do better. After all, if we can't make our technology accessible to users, who can?

I joined *Bit9* because I wanted to find a better way – a more effective security model for the future of secure computing. In the paragraphs below, I describe two of the most significant advances that together have had a transformational effect on implementing a scalable, effective whitelisting approach.

INNOVATION 1: AUTOMATICALLY MANAGING A LOCAL WHITELIST BASED ON TRUST

Ask an enterprise to manage a local whitelist of executable files, and they will fail almost immediately. Software environments are too dynamic, the applications are too cross-dependent, the files themselves are too cryptic, and it is practically impossible to make sense of what's what.

Individual file-based management of whitelists is impractical, but fortunately there are new alternatives.

Trust

Let me illustrate with an example of trust. Suppose you run a large company with tens of thousands of employees, contractors, consultants and temporary workers. You are trying to ensure that no one bad gains access to your building. You could:

1. Run a background check on every person, every time they enter the building.
2. Allow the badges to be issued to authorized individuals, which are validated upon entry.

Clearly option 1 is too time-consuming and resource-intensive. Option 2 is the obvious choice. But in order to be successful, you must create a trusted process for the issuance and validation of those badges.

You trust your Human Resources department. You may trust your contractor to issue badges on your behalf. You may apply a more rigorous screening to the temporary workers, but ultimately once they get a badge, they are sanctioned. With an efficient badge verification system at the door, you can effectively screen individuals entering and leaving the building without hindering the free flow of authorized personnel.

The IT department of an organization does essentially the same thing when deploying software. They purchase products from manufacturers, download patches from *Windows Server Update Services*, and create software in internal development groups. They put processes in place to

make sure everything is tested before it is deployed. As a result of this process, software is inherently trusted at the moment it is pushed out to a computer.

Automation

An organization's whitelist can be built – and maintained – in an automated fashion by identifying and leveraging these existing business processes. For example, by tying your whitelist to a deployment server such as *Microsoft Systems Management Server* (*SMS*), the very act of pushing out software automatically adds it to the local whitelist. Cryptographic verification can then be done centrally and automatically. This structure means whitelisting itself becomes a transparent business process.

In this whitelisting model, trust is local, not global. No third party decides which software or processes are appropriate. No third-party policy updates are required. And no untrusted software, particularly zero-day, can install or run. Businesses find this extremely attractive because they have control over their systems and processes – a highly valued attribute in today's culture of compliance.

This leads us to the second significant innovation in whitelisting technology.

INNOVATION 2: ACCURATE IDENTIFICATION AND AUTHENTICATION OF SOFTWARE

We all recognize that users – and sometimes even IT professionals – don't have a clue what to do with a message such as:

'The program blah.exe is communicating over port 1900'

What is blah.exe? Is it part of an application I own? Is it malware masquerading under a common filename? When and how did it get onto this machine?

These questions demonstrate why determining the identity and authenticity of software is so important when whitelisting.

To that end, *Bit9* has built an online 'Global Knowledgebase' for software identification and analysis, to ensure the integrity of a local whitelist.

Knowledgebase

People may be confused about the purpose of this knowledgebase – which has sometimes mistakenly been called a 'Global Whitelist' – so I want to be perfectly clear: the *Bit9* Knowledgebase does *not* dictate policies for 'good' and 'bad' software to any consumers of the service. *Bit9* has no interest in telling companies what should or shouldn't run on their computers or anyone else's.

In fact, the *Bit9* Knowledgebase operates more like a search engine for trusted information about software. It collects and reports on files, products, publishers, security assessment results, vulnerability reports, and more. These attributes are becoming richer and more descriptive over time as *Bit9* builds out the back-end of the service.

The entire knowledgebase is indexed by cryptographic hash values for each file – so a user of the service can look up a specific file, link to the application(s) of which it is a part, and access all the relevant information that has been collected.

Risk assessment

So how is the Knowledgebase used? A customer building their whitelist would first identify their sources of trust as described above. This takes care of the overwhelming majority of applications that users install or update – and our customers verify this. But of course, additional applications will trickle in. In those cases, the *Bit9* Knowledgebase is used to assess the risks associated with allowing that software to run.

Once again, this technique puts the user in control and they can choose when and if any particular software, vendor, or process should be added to the whitelist.

By taking advantage of the context provided by the *Bit9* Knowledgebase, I believe that over time, confusing messages such as the one about program 'blah.exe' can be all but eliminated.

CONCLUSION

I'd like to offer just one final thought to those who think that whitelisting will never replace anti-virus scanning. Never is a long time. And as long as we have zero-day attacks and false positives, we will need to keep progressing forward.

Judging by the milestones we have achieved and the real-world, practical feedback from our customers who are deploying whitelist-based security, it seems one cannot ignore the fact that users are finding whitelists both more effective and easier to manage than the anti-malware software they had before.

Like most security professionals, I recommend a layered, defence-in-depth approach for the best results. But when it comes to the endpoint and the significant progress we've made, I wouldn't be surprised if whitelists ultimately became the *de facto* standard for preventing malicious software on computers.

CONFERENCE REPORT

VIVA LAS VEGAS!

Andrew Lee
ESET LLC, UK

David Perry
Trend Micro, USA

VB sent roving reporters Andrew Lee and David Perry to the Nevada desert for five days of non-stop hardcore computer security at the Black Hat Briefings and hacker convention DEFCON 15. Andrew Lee reports on Black Hat, and David Perry rounds up the fun and games (and serious business) at DEFCON.

In a valley surely more deserving of the title 'silicon valley' than any other – surrounded as it is by nothing but solid rock and silicon dioxide – lies Las Vegas, a place so unreal it is equally deserving of the title 'virtual reality'. A town through which millions of dollars pass daily, and therefore a town where security is ever present, Las Vegas somehow combines feelings of freedom, fun and security in a town where attempted fraud (and worse) is a daily reality. Sound familiar? I can't help being reminded of the world wild web, where the constant struggle to protect and secure is matched by an equally energetic and opposite struggle to compromise.

Perhaps fittingly then, it is here that the largest computer security conference in the world takes place. Each year several thousand people meet over a few days in the height of summer, under the blazing heat of the sun, in the middle of a desert to learn about the latest happenings in the computer security world. Patrons of all stripe, from hackers, crackers and slackers (well, you have to have three rhymes) to feds, pros (and possibly a few cons!) and all the assorted personnel that you'd expect at a major security conference.

It's hard to grasp the scale of the conference, dwarfed as it is by the simply huge Caesar's Palace venue, but you start to realize it in the crowded corridors between presentations and the endless line to get into the lunch room.

The Black Hat conference is preceded by various training workshops, which are available for an additional fee. There is a good variety of topics on offer, and the quality is high, if a little pricey. This year's offerings included workshops on everything from reverse engineering to building secure websites.

In a somewhat unusual move, the conference proper kicked off with two simultaneous keynote addresses (in different locations). I can only guess that a few stragglers heard Tony Sager of the NSA speaking, as it was standing room only (with overspill into at least two other rooms) during Richard Clarke's keynote.

Richard Clarke, former US government member and advisor on security to four consecutive US presidents from Regan to G.W. Bush, was an engaging speaker and pushed all the right buttons, although ultimately he gave little more than an extended plug for his new novel. His vision of a dystopian (and less than secure) future was peppered with amusing anecdotes, subtle (and not so subtle) digs at the current US administration, and an exhortation to get involved in the debate – to depoliticize it so that real progress can be made. One point that I felt was right on the button (if rather obvious) was that many of the new technologies being developed are based on the assumption that 'cyberspace' is secure, and that as we build more of our economy to be reliant on connected systems, we face an increasing likelihood of more costly and more disastrous compromise.

So, with the tone set for the conference (and suppressing a smile at the irony of there being no copies of Mr Clarke's novel in stock ready for signing after his speech), I headed off to the first session I'd picked from the day's eight-track program.

One of the features, and probably the greatest appeal, of Black Hat is that it is unashamedly hardcore in its technical depth. None of the sessions on offer are for the faint of heart, and the wares on offer here range from the presentation of new exploits to forensics designed to discover exploitation. The first session I attended, Dan Kaminsky's excellent 'Design reviewing the web', introduced the rather wonderfully named 'Slirpie', which enables DNS rebinding attacks, and effectively allows VPN access to protected networks via a lured web browser. This session also brought with it my favourite quote of the conference: 'When something wasn't designed to be a security technology, don't be surprised if it doesn't act as one.'

Another feature of Black Hat is the frequent conflict one experiences in choosing sessions. With so much on offer, it's impossible to see everything and, as with the famous Las Vegas resort buffet meals, one must pick and choose – but sometimes the choices are hard. Fortunately, Black Hat records all of the sessions on professional audio visual equipment, and makes the unedited footage available. For the next session I was torn between the unveiling of Phil Zimmermann's Z-Phone technology – a method of securing VoIP – and the next shot in the escalating spat between Joanna Rutkowska and the anti-malware industry with her pills of various colours. In the end, I reached a compromise and briefly attended both – most presentations are an hour

and fifteen minutes long, which makes this more feasible than at other conferences.

The rather provocatively titled 'Don't tell Joanna, the virtualized rootkit is dead' was really an extension of Peter Ferrie's excellent paper presented at last year's AVAR conference, in which he demonstrated a multitude of attacks against virtualization software. Here, the speakers (rightfully acknowledging Peter's work by including him on their panel) presented what seems to be a very reliable way of detecting virtualized rootkits. I didn't see the culmination of this presentation, but it was enough to raise my interest in the presentation later in the day by Rutkowska and Alexander Tereshkin on subverting the *Vista* x64 kernel, promising new details on virtualized malware. Ultimately, Rutkowska and Tereshkin's presentation was to prove a little disappointing, as it did not directly answer the questions raised by the earlier presentation, but it did serve to show that there are plenty more fireworks to come from that department.

Other presentations of note were a marathon two-and-a-half-hour session on tactical exploitation given by HD Moore and Val Smith, in which there was pretty much standing room only, and an interesting examination of the 'Tidal waves of malware' given by Stefano Zanero.

The second day kicked off with a speech by one of my favourite authors, Bruce Schneier. Always controversial, provocative and amusing, Schneier presented some fascinating facts and figures garnered from experiments in human psychology and economics. What does that have to do with computer security, one might ask? Almost everything – the rather cogent point being that the way in which people behave has a huge impact on how they react to situations. A feeling of security does not equate to security, and vice versa. Some of the results Schneier showed were so counter-intuitive that there were occasional audible murmurs of surprise as he presented them. You may or may not agree with Schneier on everything, but we ignore him at our peril.

With a nine-track program, there was even more to choose from on day two than on the first day, and I kicked off with a duo of presentations in the Reverse Engineering track, the first – 'Covert debugging', featuring Danny Quist and Val Smith of *Offensive Computing* – demonstrating a tool for removing armouring from malware, and the second, on a similar theme, presented by *IBM ISS*'s Mark Vincent Yason, looking at some advanced unpacking techniques.

Later in the day I took in the Anti-Spyware Coalition panel discussion, an interesting presentation on how *Microsoft* is using the knowledge it gains from exploits to improve its *MSRC* updates, and Mikko Hypponen's presentation on the status of cell-phone malware.

Discussion of the content and good and bad points of each presentation I managed to get to would be redundant, but I hope that I've conveyed the flavour of the conference, and encouraged more *VB* readers to attend.

When I go to Black Hat, I feel challenged, and constantly stimulated by the incredible inventiveness of people in this industry and the wider security world. Such events can take us out of the everyday of work that so often blinkers our view of the world, and can put us right into the boiling flow of technology, innovation and madness that erupts from the volcano of modern computer technology.

Such conferences are also excellent opportunities to network, meet new people and rediscover old friends. In some ways, as much can be learnt over a frozen Margarita taken in the fractionally cooler late Las Vegas evening as in the most technical sessions. Finally, I have to thank the person who got me the rarer-than-hens'-teeth entry pass to the *Microsoft* party in the exclusive *Pure* nightclub – cheers!

Black Hat is traditionally followed by DEFCON, which is pretty much the same as Black Hat, but with a scruffier T-shirt – I'll leave it to fellow anti-virus miscreant David Perry to provide the details.

DEFCON 15 – THE BIGGEST PARTY IN HACKERDOM

I was addressing a company meeting at the Hard Rock Hotel in Las Vegas on the morning of August 2nd, the day before DEFCON began. 'We have to discuss how to handle this,' I said, 'Who is going to pick up the badges?'

DEFCON badges are a sight to behold. Each year they are different: stamped metal, electronic, liquid-filled, holographic, lensatic, anime, and for the last couple of years they have been in limited supply. You have to get up early to acquire the genuine article, or end up with a lovely, but surely second-rate substitute.

The second interesting thing about DEFCON is the registration. Those who have attended many COMDEX, CeBit and N+I shows in the past will recall mighty questionnaire-style registration forms, intended to determine how best the organizers can sell your soul to marketing departments: What hotel are you staying at? How many people work in your company? What is your shoe size? What really happened on grad night? It's like a Russian visa application, only more intrusive – but not at DEFCON.

Waiting in line to register for DEFCON, you notice that nobody is wearing a tie. Most are wearing T-shirts (printed with slogans like 'ALL YOUR BASE ARE BELONG TO US' and 'THE SUN IS TRYING TO KILL ME') and jeans or shorts. Hair is in evidence of colours not found in nature.

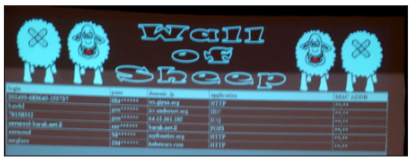
There's a guy dressed like Gandalf, multiple utilikilts, lots of unusual sunglasses. This crowd is dressed for a rock concert, or a Burning Man, and then you get to the end of the line.

'I'd like to register for DEFCON, please…' is barely out of your mouth when the goon (people working at DEFCON are designated goons) sticks out their hand. 'Hundred dollah', they intone, and you drop the Benjamin into their waiting mitt. 'Here,' they reply and shove a whole fistful of stuff at you. There's the badge, *batteries* for the badge, a CD of the presentations, a schedule and a collectable sticker.

The badge is a printed circuit board, painted white. The three logos of DEFCON (skull and crossbones, telephone dial and diskette) are visible as board cladding under the white paint (these are actually capacitance switches), as is the word 'DEFCON' across the top. Right down the middle is a scrolling LED display and at the bottom the word 'HUMAN' (or 'GOON', 'VENDOR', 'SPEAKER', etc.) is cut out of the board in bold letters. On the back are electronic components and available circuit pads for other components. When powered up, the badge lights up with the phrase 'I ♥ DEFCON 15'.

The badge itself is hackable. There are technical notes available for the badge and the keynote is all about it and what other components are available for sale so as to make a more complicated badge. There is a prize for the best hack. Some attendees spend the whole conference at a table drinking soda with a smoking soldering iron in one hand, hacking the badge. Others get involved in digital capture the flag (hacking a secured server). Still others compete at picking locks, or play 'spot the fed', a game where one attempts to identify one of the many federal agents lurking at the show. Others hang out at the dunk tank, play hacker Jeopardy or buy surplus electronics, hacking tools, black T-shirts and lockpicks in the best dealer room *ever*.

But this is a technical conference, and a damn serious one at that. Six separate tracks run non-stop for 12 hours each of the first two days, and six hours after that. Too many speakers for anyone to see – six times over! There is a wide range of topics, from 'Security by politics: why it will never work' to 'SQL injection and out-of-band channelling' to 'Creating and managing your security career'. Big names behind some of these include Dan Kaminsky, Gadi Evron, Steve Christy, and a large percentage of those who have just presented at the smaller, more sedate, more expensive (and, just *entre nous*, less fun) Black Hat Briefings. There are panels of federal agents, the EFF and disclosure experts. There are also some presentations by people who don't know they are repeating decades of other people's work. This field of study attracts more people every year, and



DEFCON welcomes N00BZ, and then PWNS them to the Wall of Sheep!

Best noted trend: more malware was discussed and analysed this year at DEFCON than in years past. Once considered the lowest form of hacker life, malware (in all its many forms) and the prevention, detection, removal and analysis of the same has, in many ways, become the lead story at DEFCON. My favourite presentations included those on trojans, botnets and web-based threats. But, as noted before, there was too much to see.

The best answer for this is the conference DVD. I went for the ultra deluxe total coverage DVD, which covers every presentation of both Black Hat and DEFCON with synchronized audio. This left me free to cover all my (daunting) bases on both shows. The DVD costs around $500.

A show like this is great for networking, for meeting old friends and making new ones. To quote *Linux* guru James Dennis 'the real action is always in the hallway track', and I spent a good deal of time there as well. It's amazing how many people have made computer security their life's work, and how many newcomers join our ranks each year. DEFCON is not a commercial trade show, and it is not an academic conference. It is an unabashed celebration of the hacking world and all that it entails. I surveyed many people, and asked their opinions about both Black Hat and DEFCON, and the consensus was clear. DEFCON is more bang for your buck, but Black Hat is better for the professional connections you can make.

Everyone's favourite thing at this DEFCON was the outing of an undercover *NBC Dateline* reporter, who was then followed out to the parking lot by jeering geeks with video cameras. All good fun, to be sure, and for those who missed it there are several *YouTube* videos of the excitement (such as http://youtube.com/watch?v=nCvmkxO5hoQ). But there is something here that just doesn't seem right. Remember that DEFCON doesn't want to expose or even know the identity of any of its attendees, protecting the anonymity of the 'hacker community'. So, the question looms: why can't an undercover reporter have anonymity, too? It's popular to deride and blame the media nowadays (not without reason), and hackers as a group have been misrepresented to a fantastic degree, but is turnabout really fair play? A famous engraving from Francisco Goya bears the following inscription:

THE SLEEP OF REASON BEGETS MONSTERS.

Persevere,

David Michael Perry

TECHNICAL FEATURE

OPENOFFICE SECURITY AND VIRAL RISK – PART ONE

Eric Filiol and Jean-Paul Fizaine
Army Signals Academy, France

The prevalence of macro viruses for *Microsoft*'s *Office* suite has declined since its peak in 2000, and recently we have seen the evolution of office software towards free, open source software in the form of the *OpenOffice* suite, which provides compatibility with many different operating systems. This two-part article presents an up-to-date evaluation of the security of *OpenOffice* (release 2.2.x), based on the results of a study undertaken during the summer of 2006. Some worrying security weaknesses have been identified, which could be used by malware to spread through innocuous-looking documents by exploiting the feelings of trust engendered by document encryption and digital signature.

This paper will discuss on a purely technical basis the pros and cons of both open and proprietary solutions as far as security is concerned. There is no such thing as a perfect solution – therein lies the complexity of computer security.

1. INTRODUCTION

For the last two years, the *OpenOffice* suite has been positioned as an open and free alternative to existing commercial office software suites. Equipped with a lot of sophisticated functionality, *OpenOffice* represents a credible, high-quality solution. Since version 2.x.x, the software has undergone a rich evolution to provide the user with an ergonomic environment. Several development environments are available as well as dedicated tools that greatly enhance the suite's overall functionality. But the existence of such an environment and capabilities raises questions as to the security of *OpenOffice* with respect to malware.

An in-depth study of the *OpenOffice* environment (releases 2.0.2 and 2.0.3 under *Linux*, *Mac OSX* and *Windows*) was conducted between June 2005 and July 2006 in the Virology and Cryptology lab at the French Army Signals Academy. The results were initially published in July 2006 [1]. This study has been technically validated by some proof-of-concept codes, but the most worrying results were only hinted at in [1] and have, to date, only been published in French [2]. This paper presents the latest technical results, for the first time in English. They have been updated since the last *OpenOffice* release (2.2.x).

The material presented here refers essentially to macro security and to the OpenDocument format (ODF) with respect to its built-in encryption and digital signature capabilities. These alone represent most of the *OpenOffice* security issues we have identified. Proof-of-concept code will not be presented here, since it is not relevant to the understanding of the article. Due to a lack of space, we have limited the extent of the technical details presented (file dumps in particular), however, they are all available upon request (for IT security professionals only).

Aside from ours [1], there have been only a few studies of *OpenOffice* security. The main ones are as follows:

- In 2003, Rautiainen [3] presented a short analysis of macros with respect to *OpenOffice* versions 1.x.
- In June 2006, *Kaspersky* claimed to have detected the first *OpenOffice* virus called StarDust. However, no technical evidence was available to support this claim, and *OpenOffice.org* later denied the self-reproducing nature of the malware.
- In May 2007, a multi-platform *OpenOffice* virus, called BadBunny, appeared. This malware seems to be a direct illustration of the risk highlighted by our own work, in particular with respect to some advanced programming languages: Python, Js and Ruby.
- In June 2007, a deep, comparative study on the security of both OpenXML and OpenDocument formats was published by P. Lagadec [4]. This study is a follow-up to [1] with respect to the viral hazard and information leakage in documents.

Apart from the search for software flaws, security analysis rarely considers functional evaluation, in other words the core algorithmic choices, or formal analysis: flow matrix, state matrix, protocol analysis. Unfortunately this is true for both open and proprietary software.

2. OPENDOCUMENT FORMAT STRUCTURE

We will not recall the structure of the OpenDocument format here – a detailed description can be found in [1]. Let us just summarize that an *OpenOffice* document is in fact a simple, compressed ZIP archive. It is thus possible to decompress it and to access and manipulate all the different document components (data, meta-data, macros etc.) very easily. Our study was conducted for ODF documents under *Linux*, *Windows* and *Mac OS*. For the sake of clarity (use of command lines) and without the loss of generality, we present here the results for the *Linux* study.

2.1 Unprotected document

Among all these components the most important one in terms of security issues is undoubtedly the manifest.xml file

located in the META-INF directory of the ZIP archive. This file describes the complete document ODF structure and all the data which are essentially relevant to the different security functionalities: macros, encryption, digital signature etc. The other files are:

- content.xml: this file is present in every *OO* document and simply contains the visible part of the document.
- meta.xml: this file contains all the document meta-information (author's data, access data etc.).
- styles.xml: this file contains the document formatting options.
- setting.xml: this file contains all the document configuration data (window size, printing parameters etc.).

Whenever one or more macros are used, a new directory is created:

```
./Basic:
total 8
drwxr-x-rx   4 lrv  lrv  138 Mar  2 01:47 Standard
-rw-r—r—   1 lrv  lrv  338 Mar  2 00:38 script-lc.xml
./Basic/Standard:
total 16
-rw-r—r—  1 lrv  lrv   350 Mar  2 00:38 script-lb.xml
-rw-r—r—  1 lrv  lrv  2049 Mar  2 00:38 a_macro.xml
./META-INF:
total 8
-rw-r—r—  1 lrv lrv  1465 Mar   2 00:38 manifest.xml
```

The directory called Basic (we will consider macros written in the default scripting language, OOBasic) contains the complete macro file tree of the document.

In addition, the manifest.xml file has been modified to record the macros and their data (access path in particular):

```
<manifest:file-entry manifest:media-type="text/xml"
 manifest:full-path="Basic/Standard/a_macro.xml"/>
<manifest:file-entry manifest:media-type="text/xml"
 manifest:full-path="Basic/Standard/script-lb.xml"/>
<manifest:file-entry manifest:media-type=""
 manifest:full-path="Basic/Standard/"/>
<manifest:file-entry manifest:media-type="text/xml"
 manifest:full-path="Basic/script-lc.xml"/>
```

There are a lot of possible scripting languages in which *OpenOffice* macros can be developed: OOBasic, JS, Python, Ruby etc. Whatever the language used, the general management scheme remains the same. Moreover, what has been presented for a single macro also holds true for complete libraries of macros [1].

2.2 Encrypted document

Whenever a user applies a document password, the document is encrypted. Let us consider a document containing a single macro. All encryption technical data are included as properties within XML tags. The encryption algorithm is Blowfish in CFB mode, the keys are derived from the PBKDF2 key management protocol, while the hashing algorithm is SHA1.

As an ODF file is in fact a ZIP archive, it is necessary to define which files in the archive are encrypted and which are not. To see where the encryption takes place, let us compare the manifest.xml file of the reference_file_encrypt.odt (encrypted) and the reference_file.odt (unencrypted) files respectively. The following is a dump (excerpt) of the diff command between the two files:

```
< <manifest:file-entry manifest:media-type="text/xml"
manifest:full-path="content.xml"/>

< <manifest:file-entry manifest:media-type="text/xml"
manifest:full- path="Basic/Standard/HelloWord.xml"/>

< <manifest:file-entry manifest:media-type="text/xml"
manifest:full-path="Basic/Standard/script-lb.xml"/>

-----

> <manifest:file-entry manifest:media-type="text/xml"
manifest:full-path="content.xml"
manifest:size="2626">

> <manifest:encryption-data manifest:checksum-
type="SHA1/1K"
manifest:checksum="ITmRG2GO+QEChZSdWuHnELeNmoU=">

> <manifest:algorithm manifest:algorithm-
name="Blowfish CFB" manifest:initialisation-
vector="UnteGYIbs8Q="/>

> <manifest:key-derivation manifest:key-derivation-
name="PBKDF2" manifest:iteration-count="1024"
manifest:salt="A1jwblqPdaNcWUpdgOF9Kg=="/>

> </manifest:encryption-data>

> </manifest:file-entry>

> <manifest:file-entry manifest:media-type="text/xml"
manifest:full-path="Basic/Standard/HelloWord.xml"
manifest:size="339">

> <manifest:encryption-data manifest:checksum-
type="SHA1/1K" manifest:checksum="zCCMsJxNl78Fzcpe/
CnNEHgo4Bs=">

> <manifest:algorithm manifest:algorithm-
name="Blowfish CFB" manifest:initialisation-
vector="auXTuBEHXHQ="/>

> <manifest:key-derivation manifest:key-derivation-
name="PBKDF2" manifest:iteration-count="1024"
manifest:salt="l/McRciiGElm7EIyxQDvRQ=="/>

> </manifest:encryption-data>

> </manifest:file-entry>

> <manifest:file-entry manifest:media-type="text/xml"
manifest:full-path="Basic/Standard/script-lb.xml"
manifest:size="350">
```

```
> <manifest:encryption-data manifest:checksum-
type="SHA1/1K" manifest:checksum="kL8H/
WhawMbDZeY47uBLZGY30qQ=">

> <manifest:algorithm manifest:algorithm-
name="Blowfish CFB" manifest:initialisation-
vector="5Y8OYH/JTkc="/>

> <manifest:key-derivation manifest:key-derivation-
name="PBKDF2" manifest:iteration-count="1024"
manifest:salt="UIv7yfLKliAq8yN5ukoI3g=="/>

> </manifest:encryption-data>

> </manifest:file-entry>
```

All the archive files are encrypted, apart from the manifest.xml file. This is very surprising, since all the attacks we have noted are possible simply by editing and modifying this particular file.

2.3 Digitally signed document

A digital signature requires a certificate. *OpenOffice* relies on external components in order to obtain certificates. We will not focus on how to import those certificates here (see the *OpenOffice* inline help for more details), other than to say that we used X509 certificates for our research. We should mention that the ODF specifications [5] do not provide details about the use of digital signatures. We thus had to analyse how the signature is applied. For that purpose, let us consider an encrypted *OpenOffice* document containing a macro called ref_mac_enc_sig.odt.

The following is the most relevant data. More detailed data will be provided on request:

```
ZZR:~/Research/Analysis/OpenOffice.org/Work/
Attaque_odf_190507/Struct_study lrv$ unzip
ref_macro_enc_sig.odt -d ref_macro_enc_sig_ext

Archive:  ref_macro_enc_sig.odt

 extracting: ref_macro_enc_sig_ext/mimetype

   creating: ref_macro_enc_sig_ext/Configurations2/
statusbar/

 extracting: ref_macro_enc_sig_ext/Configurations2/
accelerator/current.xml

   creating: ref_macro_enc_sig_ext/Configurations2/
floater/

   creating: ref_macro_enc_sig_ext/Configurations2/
popupmenu/

   creating: ref_macro_enc_sig_ext/Configurations2/
progressbar/

   creating: ref_macro_enc_sig_ext/Configurations2/
menubar/

   creating: ref_macro_enc_sig_ext/Configurations2/
toolbar/

   creating: ref_macro_enc_sig_ext/Configurations2/
images/Bitmaps/

  inflating: ref_macro_enc_sig_ext/META-INF/
macrosignatures.xml

  inflating: ref_macro_enc_sig_ext/META-INF/
documentsignatures.xml

 extracting: ref_macro_enc_sig_ext/content.xml

 extracting: ref_macro_enc_sig_ext/Basic/Standard/
HelloWord.xml

 extracting: ref_macro_enc_sig_ext/Basic/Standard/
script-lb.xml

 extracting: ref_macro_enc_sig_ext/Basic/script-
lc.xml

 extracting: ref_macro_enc_sig_ext/styles.xml

  inflating: ref_macro_enc_sig_ext/meta.xml

 extracting: ref_macro_enc_sig_ext/Thumbnails/
thumbnail.png

 extracting: ref_macro_enc_sig_ext/settings.xml

  inflating: ref_macro_enc_sig_ext/META-INF/
manifest.xml

ZZR:~/Research/Analysis/OpenOffice.org/Work/
Attaque_odf_190507/Struct_study lrv$
```

We can see that two new files have been added into the archive: META-INF/macrosignatures.xml and META-INF/documentsignatures.xml. The manifest.xml file contains the relevant data for those two files:

```
<manifest:file-entry manifest:media-type=""
manifest:full-path="META-INF/macrosignatures.xml"/>

<manifest:file-entry manifest:media-type=""
manifest:full-path="META-INF/documentsignatures.xml"/>

.................
```

The two new files are not encrypted. This point is essential to the understanding of the attack mechanisms of encrypted documents. The document.xml file contains the detailed data that are required during the digital signing process itself (not given here). Every file in the archive is unsigned while the macros' signatures are declared within the macrosignatures.xml file. To be more precise, the digital signature is applied to any file containing or relating to macros [6, 7].

3. A FORMAL ANALYSIS OF THE OPENOFFICE DIGITAL SIGNATURE

OpenOffice.org's security is based on two essential mechanisms: password-based encryption and digital signature. Both aim to prevent illegitimate use or manipulation of a document. In the context of document malware, any weakness with respect to either of these mechanisms could be exploited in a powerful way to fool the user's trust in cryptographic protection.

Since there are a lot of ways of using encryption and signatures to protect an *OpenOffice* document, we will use a formal graph-based approach to describe them all. Every node in our graph describes a user's action. A given path in our graph describes a sequence of such actions to encrypt

and/or sign a document. This is a very powerful approach for detecting security flaws – in other words, cases where the security mechanisms are supposed to have been applied while in reality they have not been (the document is not encrypted or not signed).

Our graph-based formalization aims to identify weaknesses in the signature process, in particular with respect to macros. To summarize, we will show that the signature of the document's visible content and the signature of the macros are mutually exclusive: we cannot sign them at the same time. Using technical examples, we will prove in subsequent sections that this constitutes a serious design flaw that can be exploited efficiently by malware.

Every node describes a possible status for the document:

- D: modified document.
- MD: modified document with macro.
- ED: saved document.
- EMD: saved document with macro.
- SED: document is signed and saved.
- MSD: document with a macro added *after* the document has been signed.
- EMSD: document with a macro added *after* the document has been signed, but *before* the document is saved.
- SEMD: signed and saved document with a macro.

The nodes are connected by labelled arrows. The labels describe a user's commands/actions applied to the document:

- add M: add a macro.

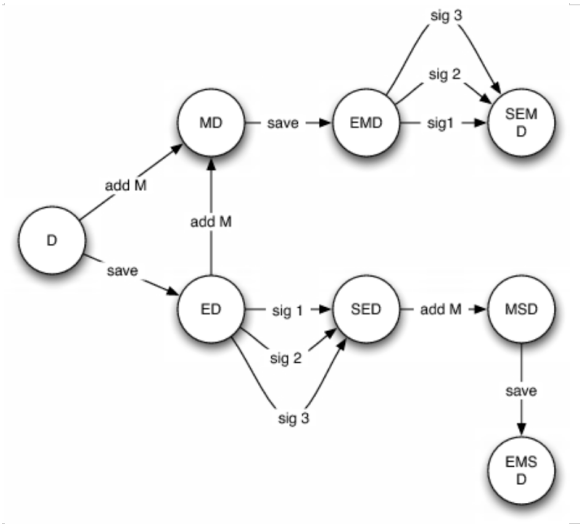


Figure 1: OpenOffice signature graph.

- save: save document.
- sig 1: sign through the *File* → *Digital Signatures...* menu.
- sig 2: sign through the *Tools* → *Macros* → *Digital Signature...* menu.
- sig 3: sign through the second bottom right box in the *OpenOffice* GUI.

The corresponding graph is depicted in Figure 1.

In Figure 1, we notice that the graph is divided into two connected components. The first sub-graph is made up of nodes MD, EMD and SEMD, while the second one contains nodes ED, SED, MSD and EMSD. This supposes two different possible uses of a digital signature that can be applied at any time in the life of the document. However, our experiments have proved that it may be quite different. Let us explain why.

When considering signature *and* encryption at the same time, our approach remains essentially the same. The set of nodes is generalized as follows:

- D: modified document.
- MD: modified document with macro.
- (SE)MD: encrypted and saved document with macro.
- S(SE)MD: signed, encrypted and saved document with macro.
- (SE)D: encrypted and signed document.
- M(E)D: a macro is added to an encrypted and saved document.
- SM(E)D: a macro is added to an encrypted, saved and finally signed document.
- S(SE)D: modified document which has been saved, encrypted and signed.
- MS(E)D: a macro is added to an encrypted and signed document.
- EMS(E)D: signed then encrypted document with macro.

We thus obtain the graph depicted in Figure 2. It is very powerful for identifying the potential misuse of digital signatures in *OpenOffice*. Such misuse could be by malware, as we will show in the next sections. Two classes of design flaw have been identified. The first class deals with the lack of built-in document integrity management. The second class refers to problems that may occur when macros and/or document are signed. We will not discuss the critical issue of trust macros. They are presented in [1].

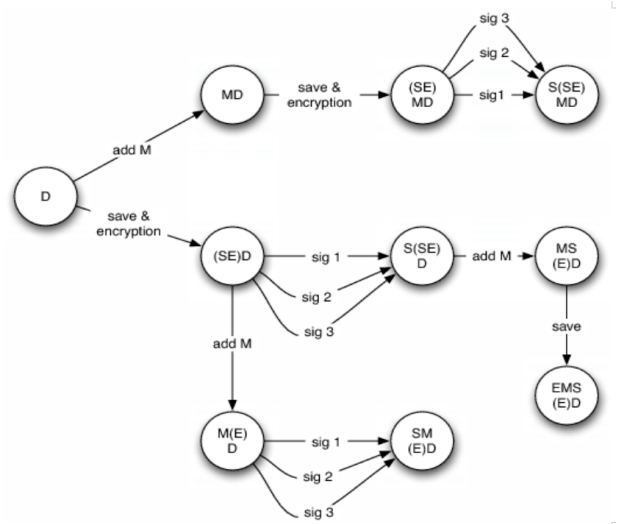


Figure 2: OpenOffice encryption and signature graph.

4. SECURITY ISSUES IN OPENOFFICE ENCRYPTION AND SIGNATURE

The main problems with *OpenOffice* encryption and signatures do not lie in the cryptographic tools themselves but in their implementation and management. The main consequence is that a malicious program may be able to identify interesting blocks of instructions and thus adapt itself to the target document. This is particularly worrying with respect to macros which constitute execution points that can be subverted by a malicious program, despite the apparent use of a digital signature.

In this section, we will consider some cases where it is possible for malware to bypass a digital signature or to exploit it. All these experiments have been successfully carried out without causing any integrity violation alert and while preserving the document usability.

Let us first mention that the use of a digital signature can be identified within the archive in two ways:

- An additional file, denoted documentsignatures.xml, is created in the META-INF directory of the archive.
- An entry is added into the META-INF/manifest.xml file:

```
<manifest:file-entry manifest:media-type="" manifest:full-path="META INF/documentsignatures.xml" />
```

4.1 Signed OpenOffice document with an unsigned macro

Let us first create a signed document containing a macro. Contrary to the user's belief, the macro itself is not signed. As a consequence (see next sections), a malicious program can modify a macro without triggering any integrity violation alert, thus undermining the user's feeling of security with respect to the digital signature.

The listing of the archive clearly shows that there is a signature which is applied to the document, but not to the macro. The META-INF/documentsignatures.xml does not refer to any macro signature:

```
ZZR:~/Research/Analysis/OpenOffice.org/work/Attaque_odf_190507 lrv$ unzip doc.odt -d doc_dir
Archive:  doc.odt
  extracting: doc_dir/mimetype
     creating: doc_dir/Configurations2/statusbar/
   inflating: doc_dir/Configurations2/accelerator/current.xml
     creating: doc_dir/Configurations2/floater/
     creating: doc_dir/Configurations2/popupmenu/
     creating: doc_dir/Configurations2/progressbar/
     creating: doc_dir/Configurations2/menubar/
     creating: doc_dir/Configurations2/toolbar/
     creating: doc_dir/Configurations2/images/Bitmaps/
   inflating: doc_dir/META-INF/documentsignatures.xml
   inflating: doc_dir/content.xml
   inflating: doc_dir/Basic/Standard/Hello.xml
   inflating: doc_dir/Basic/Standard/script-lb.xml
   inflating: doc_dir/Basic/script-lc.xml
   inflating: doc_dir/styles.xml
   inflating: doc_dir/meta.xml
   inflating: doc_dir/Thumbnails/thumbnail.png
   inflating: doc_dir/settings.xml
   inflating: doc_dir/META-INF/manifest.xml
```

4.2 Signed OpenOffice document with a signed macro

In this case, the signature is applied independently to the visible part of the document and to the macro itself. The only way to apply the signature to both of them simultaneously is to successively follow path sig2 *and* paths sig1 *or* sig3 in the graph shown in Figure 2.

To illustrate this, let us first sign the macro:

```
ZZR:~/Research/Analysis/OpenOffice.org/work/Attaque_odf_190507/App_dsig_doc_macro lrv$ unzip doc1.odt -d doc1_dir
Archive:  doc1.odt
  extracting: doc1_dir/mimetype
     creating: doc1_dir/Configurations2/statusbar/
   inflating: doc1_dir/Configurations2/accelerator/current.xml
     creating: doc1_dir/Configurations2/floater/
     creating: doc1_dir/Configurations2/popupmenu/
     creating: doc1_dir/Configurations2/progressbar/
```

```
   creating: doc1_dir/Configurations2/menubar/
   creating: doc1_dir/Configurations2/toolbar/
   creating: doc1_dir/Configurations2/images/Bitmaps/
  inflating: doc1_dir/META-INF/macrosignatures.xml
  inflating: doc1_dir/content.xml
  inflating: doc1_dir/Basic/Standard/Hello.xml
  inflating: doc1_dir/Basic/Standard/script-lb.xml
  inflating: doc1_dir/Basic/script-lc.xml
  inflating: doc1_dir/styles.xml
  inflating: doc1_dir/meta.xml
  inflating: doc1_dir/Thumbnails/thumbnail.png
  inflating: doc1_dir/settings.xml
  inflating: doc1_dir/META-INF/manifest.xml
```

Only the macro is signed. Let us then sign the document content itself. The listing clearly shows two different signature files:

```
ZZR:~/Research/Analysis/OpenOffice.org/work/
Attaque_odf_190507/App_dsig_doc_macro lrv$ unzip
doc1.odt -d doc1_dir
Archive:  doc1.odt
 extracting: doc1_dir/mimetype
   creating: doc1_dir/Configurations2/statusbar/
  inflating: doc1_dir/Configurations2/accelerator/
current.xml
   creating: doc1_dir/Configurations2/floater/
   creating: doc1_dir/Configurations2/popupmenu/
   creating: doc1_dir/Configurations2/progressbar/
   creating: doc1_dir/Configurations2/menubar/
   creating: doc1_dir/Configurations2/toolbar/
   creating: doc1_dir/Configurations2/images/Bitmaps/
  inflating: doc1_dir/META-INF/macrosignatures.xml
  inflating: doc1_dir/META-INF/documentsignatures.xml
  inflating: doc1_dir/content.xml
  inflating: doc1_dir/Basic/Standard/Hello.xml
  inflating: doc1_dir/Basic/Standard/script-lb.xml
  inflating: doc1_dir/Basic/script-lc.xml
  inflating: doc1_dir/styles.xml
  inflating: doc1_dir/meta.xml
  inflating: doc1_dir/Thumbnails/thumbnail.png
  inflating: doc1_dir/settings.xml
  inflating: doc1_dir/META-INF/manifest.xml
```

A deeper analysis clearly shows that the two signature files have not been created at the same time:

```
ZZR:~/Research/Analysis/OpenOffice.org/work/
Attaque_odf_190507/App_dsig_doc_macro/doc1_dir2/META-
INF lrv$ ls -l *
-rw-r--r--   1 lrv  lrv  5875 Jul 12 14:45
documentsignatures.xml
-rw-r--r--   1 lrv  lrv  5657 Jul 12 14:30
macrosignatures.xml
-rw-r--r--   1 lrv  lrv  2602 Jul 12 14:45 manifest.xml
```

This constitutes a critical design weakness since the user must be aware of the fact that a specific signature process must be applied to sign *OpenOffice* macros. The main consequence is that most of the time it is possible to modify (infect) macros without violating the document's integrity.

5. ATTACKING ODF: A FORMALIZATION

ODF is based on the XML technology. All the information is contained within XML tags, thus giving a semantic value to the information. All the XML tags are then organized within a tree structure that makes the extraction of the information far easier. These structures enable ODF to be formalized in a very powerful way by means of automata and language theories. For the sake of brevity we will not present the whole of our formalization work. The reader can refer to our technical report [8].

In order to identify all possible attacks that can be operated by malware against an *OpenOffice* document by exploiting integrity and/or signature management flaws, we have applied a formal model based on a graph-theoretic approach again. The graph is defined as follows:

- Each node represents the document status at time *t*, before any action is applied to it.

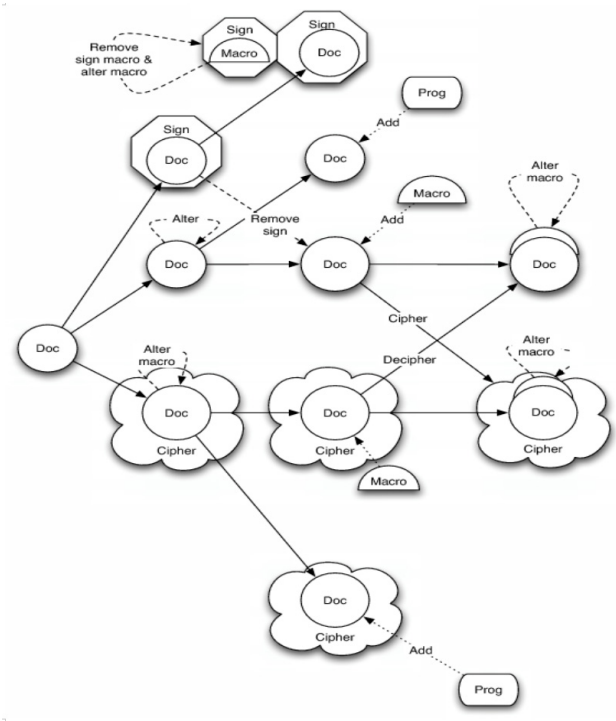


Figure 3: OpenOffice attack graph.

- The arrows represent the different possible attacks that can be performed once a given action (command) has been applied.
- The node surrounding area defines a given feature for the document.
- An action is applied to a node feature and thus defines which attack has been performed:
  - Add: add a property or a macro to the document.
  - Alter: modify a document property or component.
  - Cipher: the document is encrypted by using *OpenOffice* built-in tools.
  - Decipher: the encrypted document is deciphered by using *OpenOffice* built-in tools.

We thus obtain the attack graph depicted in Figure 3, which describes things at the lowest level. It is also possible to combine different document statuses to exhaustively describe all possible attacks at every possible level.

In part two of this article (which will appear in the October 2007 issue of *Virus Bulletin*) we will look at security issues in *OpenOffice* integrity management and draw conclusions about the overall security of *OpenOffice*.

REFERENCES

[1] De Drézigué, D.; Fizaine, J.-P.; Hansma, N. In-depth Analysis of the Viral Threats with OpenOffice.org Documents. Journal in Computer Virology, (2)-3, 2006.

[2] Filiol, E.; Fizaine, J.-P. Le risque viral sous OpenOffice 2.0.x. MISC – Le journal de la sécurité informatique, vol. 27. 2006. http://www.miscmag.com/.

[3] Rautiainen, S. OpenOffice Security. Proceedings of the Virus Bulletin International Conference 2003.

[4] Lagadec, P. OpenOffice/OpenDocument and MS Open XML Security. PACSEC 2006 Conference. http://pacsec.jp/psj06archive.html.

[5] Open Oasis. OpenDocument Specifications v1.1. http://www.oasis-open.org/specs/.

[6] W3C. Specification signature W3C. http://www.w3.org/Signature/.

[7] XML. http://www.xml.com/pub/a/2001/08/08/xmldsig.html and http://www.w3.org/TR/xmldsig-core/.

[8] Filiol, E.; Fizaine, J.-P. Security Analysis of the ODF Security: a Formal Approach. ESAT Technical Report 2007-21.

PRODUCT REVIEW

BITDEFENDER TOTAL SECURITY 2008

John Hawes

Despite being a fairly young company, *BitDefender* – set up in Bucharest, Romania by its parent *Softwin* as recently as 2001 – has already established itself as an important player in the security market. Taking over *Softwin*'s *AntiVirus eXpert* (*AVX*) technology, *BitDefender* has expanded its range of offerings to include home-user and corporate solutions, desktop, server and gateway products, and support for *Linux* and *FreeBSD*. These products have been promoted vigorously and effectively, pushing the technology and the brand into the public consciousness. The company has expanded from its Romanian base to set up branch offices in France, Spain and the USA. With strong brand recognition and a reputation for solid, dependable detection rates and impressive heuristics, the company has become one of the most trusted names amongst mid-sized security firms.

BitDefender's products have been regular participants in *VB*'s comparative reviews since the *AVX* days, and after a slow start soon found their stride – first achieving VB100 certification in 2003 and failing only once since, when stability issues in a tricky *Linux* test got the better of the product in 2004. The product's appearance in our tests has been less dependable of late, having skipped both *Vista* comparatives this year, but *BitDefender*'s record remains strong. Its products are also certified by the standard bodies *ICSA Labs* and *West Coast Labs*, and highly rated by test centres such as *AV-Comparatives*, where they regularly score 'Advanced' and 'Advanced+' ratings – particularly in retrospective tests looking at proactive detection of new threats – and *AV-Test*, where the *BitDefender* name is common among the top five ranked products.

BitDefender Total Security 2008 (*TS2008*) is the latest version of the company's home-user product, released shortly before the publication of this review. It focuses strongly on the needs of the home user, and the interface – previously criticised for being a little technical and daunting – has been redesigned and simplified with the inexpert user in mind.

The full suite covers a wide range of security issues, with standard anti-malware, firewall and anti-spam functionalities augmented with backup facilities, system tune-up, phishing protection and more. A pared-down sister suite, *BitDefender Internet Security 2008*, is also available, offering everything apart from the backup and tune-up facilities, and a pure anti-malware product completes the range. The product supports *Windows XP* and *Vista*, 32-bit and 64-bit, and is available in English, French, German and

Spanish, with more languages including Japanese due to be added by the end of the year.

WEB PRESENCE, INFORMATION AND SUPPORT

The main home of *BitDefender* on the web, www.bitdefender.com, is a fairly slick and attractive place, with a fixed width design which leaves wide swathes of greyish-white space in the browser. The company's trademark red, black and steely grey colour scheme looks glossy and elegant without the cutesy cartoonish look which has become popular on many security sites lately.

The first link on the navigation bar leads to information about the company itself, revealing that its slogan is 'Securing your every bit' and that it is justly proud of having won the *PC World* '#1 Best Buy' title in March 2006. The company goal is to become one of the top six AV solutions by 2010, and a quick glance through its history shows steady progress in this direction. The company's technological history is even more impressive, boasting firsts in web and messaging scanning, automated updating, behavioural technologies, and bundling firewalls with AV products. Further information on the technology provides detailed overviews of its capabilities and is aimed at potential partners – of whom there are already many, the *BitDefender* engine being one of the most popular for inclusion in multi-engine offerings.

Curiously, the *VB* logo is missing from the otherwise rather crowded awards page – an oversight which I'm sure will soon be corrected. Further information for would-be partners and information on the product range is also available, as is an online shop and a download area providing evaluation versions, removal tools and updates. A free version of the home-user AV product is available, and the *Linux* scanner is also free to download.

Of most interest to me on the site, however, were the sections labelled 'Support' and 'Defense Center'. The latter area contains a malware encyclopaedia, which is not the most exhaustive I have seen, with entries numbering in the hundreds rather than the tens of thousands that would be needed to cover the ever-expanding range of nasties out there. The encyclopaedia also suffers from a rather defective search system – common terms such as 'mytob', 'netsky' and 'bagle' returned 'no documents', despite most of them being easy to find with a little browsing. However, when an individual item of malware is picked out of the encyclopaedia, the information provided goes into great depth, and is put across with admirable clarity.

A set of free-to-download removal tools is also provided, covering a lengthy list of common infections, along with an online scanner, which boasts of being a 'fully functional antivirus product', offering full system and registry scanning and disinfection. Like so many of these offerings, it requires *Internet Explorer* to work, using ActiveX controls to access the system.

A news section, backed up by various email alert options, provides the latest stories on major malware outbreaks and developments, along with technological advances in *BitDefender*'s products. 'Real-time' malware statistics are also offered, drawing on feedback from products in use across the world – these can be divided into desktop and gateway figures, display numbers of files or systems infected, and can be tweaked to show smaller or larger time periods, providing some interesting insight into what is hitting users hardest at any given time. Finally, an advice page offers 'Ten Commandments for Your Computer Sanity', a nice simple set of rules setting out the basics of safe computing.

In the Support section, the visitor is first guided towards information for the relevant type of user (home, small business or corporate) and the relevant product. Links are then provided to downloads, a rather limited knowledgebase (the *TS2008* section contained only an explanation for the 'I/O errors' entry in log files), a 'documentations' section, (which had I assumed would provide manuals but the linked PDFs were in fact rather simple datasheets), and support contact details. Unlimited 24/7 support is provided with the *TS2008* licence, and advice can be sought by phone, via email or via an online chat system.

This part of the site, like several others, is adorned with customer recommendations, the most impressive of which is from a Mr David Perry, who apparently was so pleased with his *BitDefender* purchase that he threw his popular rival product 'on the burn pile and watched it melt'.

INSTALLATION, OPERATION AND DOCUMENTATION

Installation of the system follows a pretty standard path. After an initial, rather lengthy period of extracting files, there came the usual EULAs, warnings about removing other security products, and a message promising 'expert protection' and boasting of the product's certification by *VB* and others, before the installation proper began. During this process, a licence key can be entered (although an evaluation period is also available, which seems not to limit the functionality of the product), a user login for the *BitDefender* site can be set up, and the product runs an update and a scan of the local system to ensure it is clean before installing. After the initial extraction period, the process is speedy and keeps the questions to a minimum,

allowing novice users to run through it without anything tricky to worry about.

The interface is a pretty simple thing, modelled somewhat along the lines of the *Windows Security Center* – just four big, fat icons, representing the four main areas of the product, show the status of the parental controls, the security provision, the backup and the tune-up facilities. In my case all but the security icon started off with a big grey cross through them to indicate that they had yet to be configured. The security icon was marked with a red exclamation mark – a warning that the product had yet to update. A red button to one side of the icons offers the chance to 'Fix all issues', while along the bottom some tabs provide various tasks for each area.

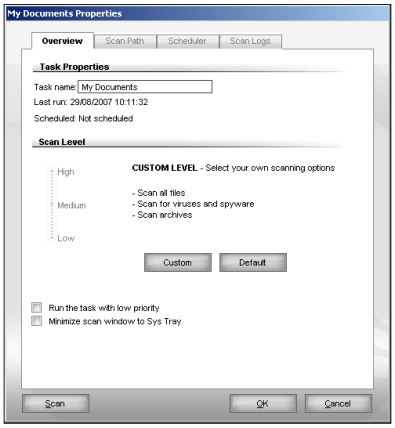
Clicking this, or any of the chunky buttons for the individual areas, brings up a list of the various items considered to be vital to the security of the system. Those which have yet to be configured or run, or which have encountered some trouble, are marked with a red 'Fix' link. This brings up the appropriate scan, config page or wizard, while the 'Fix all' links bring these up in chains, moving on to the next as each one completes.

The bottom of the page lists some 'quick tasks', divided into tabs for the different modules. The security tab, displayed by default, offers updating and a series of scans. The scan options were 'Documents', which does a quick check of the registry, the 'My Documents' folder and files on the desktop; a 'Full system scan'; and a 'Deep system scan'. Both the 'Full system scan' and the 'Deep system scan' apparently scan all local drives, the difference being that the 'Full' scan doesn't include archives, while the 'Deep' one does.

The 'Backup' tab within this area offers access to the backup and restore wizards, the first of which is activated by the 'Fix' buttons mentioned earlier – this presents lists of items that can be backed up, and a place to store them, while the restore system does the same in reverse. There is also a full backup utility linked from here, which opens a new window containing some in-depth controls for managing backups. The 'Tuneup' tab offers the same series of wizards provided earlier by the 'Fix' buttons, which can be run again as often as the user requires – disk defragmenting, removal of unwanted data and registry cleaning can all be run through via more, fairly straightforward wizards.

Leaving these for later, I ran through a couple of the malware scan jobs, which zipped along in quite good time even when scanning a whole system. My concerns that there would be no way of tuning or tweaking this side of the product were soon allayed when I spotted a small button in the bottom corner of the main GUI marked 'Settings'. This opened a new window, less glossy than the main interface or the wizards, which proved to be an in-depth and fully featured configuration area, with options to set up, schedule and run scans, tweak the settings of the on-access monitor, and also to control the other aspects not covered by the simplistic main GUI – the firewall, spam-filtering, and parental and privacy controls.

Much of this area is not merely less glossy, but perhaps inevitably less accessible to the non-technical user, presenting quite a lot of fairly serious data with little by way of explanation. However, the front page for each sub-section presents a slider with which the security settings can quickly and simply be changed from a default in the middle to the highest or laxest security levels, with some statistics on activities carried out (files or mails processed, web connection activity etc.) – in some cases in the form of nice little real-time graphs. The real nitty gritty of the configuration is then accessed via tabs, the setup thus providing the inexpert user with a way of adjusting the settings without the need for too much research, while the more skilled (or foolhardy) can delve into the depths of configuration as they please.



For novices who do need to make more involved changes to the configuration, and even those simply running through the wizards, something that is conspicuous by its absence is help. I mentioned earlier that the website offered little by way

of documentation, and no full manual was provided (at least with the download version tested here). A help file is available, but can only be accessed from a link on the front page of the main interface. Further tips, advice, clarification and so on would make a handy addition to some of the more difficult or important areas, and at the very least links to the appropriate pages of the help system should be considered.

The help itself seems reasonably comprehensive, although in some places the language or layout is a little awkward. There are a lot of very long pages, with whole chapters stored in a single lot (although broken up for the links in the contents tab) and many very large screenshots, which necessitate some sideways scrolling, even at fairly high resolutions.

The help focuses mainly on running through the steps required by each task, but each section also has a brief explanation of the reason for the functionality and advice on how and when to use it. These are mostly just brief introductions to each section, but in the 'advanced' chapters a series of 'insights' give a nice broad overview of the problems solved by each individual component, presented in a pleasantly user-friendly manner.

MALWARE DETECTION AND PROTECTION

With the interface thoroughly explored, I moved on to some basic tests of the product's malware detection. Malware scanning can be achieved in numerous ways: using the big 'deep' and 'full' buttons on the main interface, using a context-menu option, dragging files to the little semi-transparent status box, or simply trying to open something and seeing if the monitor finds anything suspicious there. They all seem to run pretty smoothly, and with some good stability. Several attempts to overwhelm the program – by running several processes opening hundreds of infected samples in quick succession while a manual scan chugged along over some more samples, and still more piled in over the network – failed to bring the thing to its knees or let anything slip past it, although some slight slowdown in the system was inevitable.

Running scans over the standard *VB* test sets produced unsurprising results – *BitDefender* has always had excellent detection rates in VB100 tests, and the few samples usually missed are all rather obscure and elderly (and even these are, apparently, being worked on by diligent lab staff). Even the very latest batches of samples available failed to defeat the scanner, or its heuristics, which flagged just about everything thrown at it in some style. A couple of items not spotted by the scanner were quickly picked up as they tried to install themselves in the registry and drop things into system folders.

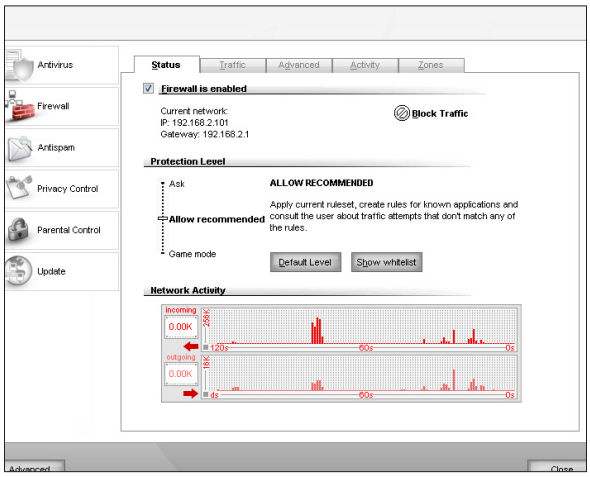
Cleanup was also pretty solid – the default settings on access are to disinfect or remove malicious files, and this was done without difficulty on the few items tested. Unfortunately the product was only on the *VB* test bench for a few days, and little time was available for in-depth testing of the behaviour-blocking, heuristics and removal prowess of the product, but we hope to be able to introduce more regular testing of these aspects of products in the future, and I expect to see *BitDefender* well up in the rankings.

Scanning speeds were pretty good throughout, perhaps making some slight improvement over times recorded in earlier VB100 testing, though as both the systems and the test sets have been altered several times in recent months no direct comparison can be made. On-access speeds seemed particularly impressive, and I rarely noticed any slowdown on more up-to-date systems – some older machines did suffer a little under the weight, especially when the interface was running, lagging for several seconds between screens and so on, but in this day and age few people will be using such tired old hardware.

OTHER SECURITY FUNCTIONS

Of course, the malware scanning provided under the 'Security' tab is far from everything the product has to offer. Also covered by general security, and configured from the advanced settings area, are a range of items including the firewall, anti-spam, and privacy and parental controls.

The firewall page starts off fairly simply, with a status indicator and a nice plain slider to tweak its strictness. In normal modes, the firewall seems to offer a sensible level of protection from the off, automatically including a well populated whitelist of trusted software which seems to include most standard items without pestering the user too much. Adding new rules for connections is a straightforward task, again with further configuration available on an



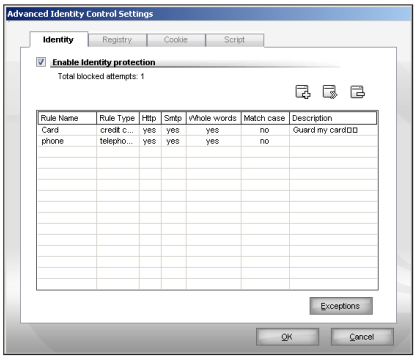
'advanced' tab. The whitelist can also be bypassed easily, using the slider to set security to a higher level, and a handy button allows instant blocking of all traffic.

A graph of traffic on the main page is supplemented by a nice table of activity on a separate tab, showing all connected processes, the ports they are using and the traffic passing through them. Another sensible idea is to class unsecured wireless connections automatically as untrusted – a warning pops up when trying to connect to an unsecured router, informing the user that such connections can be used by adding a rule, but suggesting that securing the connection point would be a smarter move.

The firewall also offers 'Game mode', which shuts off messaging and updating to minimize impact on the system or user during game sessions, and relaxes the connection rules to block only those actions which have specific rules defining them. This is 'strongly discouraged', according to the manual, though there is little such discouragement in the product itself. It does at least allow gamers – especially experienced firewall tweakers – to implement some level of protection without impacting too much on their fun.

The anti-spam module integrates with a range of mail readers, providing a nice little toolbar with which to manage messages and contacts, with mails easily marked as coming from friends or annoyances, and other addresses likewise. Addresses are imported via a simple wizard, and a little training system can be pointed at existing folders of good or spam mail to get an idea of what kind of a mail user you are (I was confused for a moment when it asked me to 'select witch folder' – I thought maybe this was an unusual term for a list of known bad folk, but later decided it must be a typo). The system can also be set to drop messages in Asian or Cyrillic character sets automatically, and to use heuristic filtering techniques to improve performance.

The next section, labelled 'Privacy control', is a bit of a mixed bag. The first part lets the user store a selection of important data – addresses, telephone, bank account and credit card numbers etc. – which are then watched for and, if the information is spotted leaving via the web or email, the action is blocked. This kind of system always makes me feel a little uncomfortable, but the data is apparently kept well encrypted and the blocking seems effective. A whitelist of sites that are allowed to receive such data can be created. There is also a phishing filter provided, with its own little toolbar added to *IE*, which watches for suspicious sites and again allows the creation of a user-defined whitelist.



A registry monitor blocks unauthorised changes to the system's startup list and other settings, while cookie and script blockers control which sites can and cannot drop cookies and run active content, again using sets of allow/deny rules. Finally, there is a system info tab providing lots of information about the local system, including startup items, browser components and helper objects, active processes and their imports, and much more besides. Only minimal explanation is provided here, and this would only be of much use to users with considerable insight into their systems.

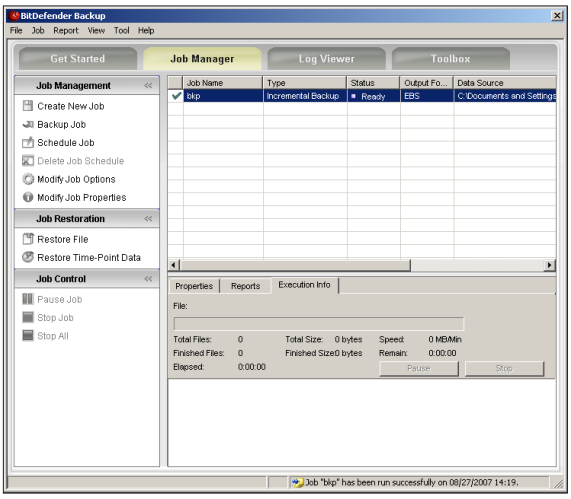
Parental controls are a more straightforward area – unlike many such things, there is no long list of areas which should be filtered, just per-user settings for children, teenagers and adults (the teens are allowed access to drugs and hacking-related content, but not gaming, according to the on-screen summary). Additional filtering is available for URLs and keywords, and a time-control mechanism completes the set of parental controls.

FURTHER FUNCTIONALITY

The product ensures it lives up to the 'Total' part of its name with yet more functionality – the backup and system maintenance features. Such items are becoming increasingly common in security suites, with several major products rolling in some level of backup or cleaning alongside more traditional security measures. Here, they are kept somewhat separate from the security area, being assigned two of the four fat buttons on the main console.

The backup feature provides a simple wizard with which incremental or full backups of ranges of files and data can be run to local drives, across networks or to removable devices. This seemed fairly straightforward, and another fat button in the 'quick tasks' tray also allows for speedy restoration of the backed-up data. Another link leads to a much more in-depth interface, where these backups can be configured in more detail, including compressing and encrypting the backup targets and a vast range of other options. There are facilities for viewing logs of progress and errors monitored, and even a CD-burning system to create permanent backups.

The final part of the product is the 'Tuneup' section. This again provides a range of options, although this time they appear only to be manipulable from the main interface and are somewhat less mature than the rest of the offerings. 'Defragmentation' has a very basic wizard, just offering a list of available drives, and it seems to check drives to see if



defragging would be of any benefit, but then has a go at tidying up even when it has informed the user that it is unnecessary. It certainly spends quite some time processing all drives, even when the same process has been run just moments earlier. An Internet file removal tool is next – this is pretty straightforward too, with no options at all: you start it, and it removes temporary files and cookies. Both tools seem a little pointless, as such functionality is already readily available, but I suppose having it all in one place is handy for some.

Less common is the file shredder, which trashes deleted files securely using the US Department of Defense method – which is generally considered fairly solid, although is not quite as fanatically thorough as methods offered by another product reviewed here recently (see *VB*, March 2007, p.13). It's simple to use – just point it at a file or folder and click 'go' – and pretty speedy.

Another deletion tool, the 'Duplicate files remover', is a little more worrying. Pointing the thing at my C: drive, it produced a list of 'duplicates' in groups, with the default action being to remove all but the newest. Unfortunately this included lots of important place-markers and other useful parts of the *Windows* system, the removal of which would have trashed various (admittedly mostly minor) functionalities. There was a warning that there were system files and folders included in the scanned area, but it would perhaps have been safer and more sensible to have excluded these automatically, or at least to have set the default to do nothing in sensitive areas and force the user to check them through. The tool seems useful for checking over large stashes of media, music files or pictures where duplicate copies of large files can easily start cluttering up disk space, but perhaps some more guidance would be useful.

Finally, there is a registry cleaner, which attempts to identify unneeded or 'orphan' registry entries and remove them. Again this sort of technology is notoriously tricky and prone to false positives, but can be useful; thoughtfully *BitDefender* also provides a 'Registry restore' tool, positioned just as prominently as the cleaner, to undo any catastrophic cleaning efforts.

CONCLUSION

Overall, *BitDefender*'s latest product leaves a pretty positive impression. It is hard to do justice in such a small amount of space and such a short time to the huge range of functions included here – several months would probably be needed to put all this through its many paces.

The redesigned interface is pretty successful – its simplicity is impressive, offering peace of mind for the inexperienced home user without the need for a degree course to operate it. In some places a little more clarity would be useful, particularly where the links to the help pages are concerned, but for the most part the basic console leaves little room for confusion. The duplicate remover could perhaps do with a little tweaking to eradicate any danger of causing damage, but the other tools require little more than clicking 'fix' and occasionally 'next'. Despite my initial fears, there is a vast wealth of in-depth configuration available beneath the simple interface – enough to satisfy even the most hardened techie. The only thing I missed was the option to peruse logs of on-access monitoring, which is perhaps unlikely to be needed by anyone but a tester.

The main purpose of a security suite is to provide security, and this *BitDefender* does in spades, effortlessly spotting and blocking malware, controlling access to and from the network, flagging suspect websites and keeping an eye on emails. The cleaning angles may need a little more work to achieve perfection, but the backup tool is a splendid offering, particularly the grown-up version with the vast wealth of extra options. Hopefully, as this sort of thing becomes more and more common in security products, users will realise the importance of keeping secure copies of their important data. And once this message sinks in, with the gentle encouragement of products like this, backed up by powerful detection and protection, users' data and systems will be more thoroughly secured.

Technical details:

BitDefender Total Security 2008 was tested on:

AMD K7, 500Mhz, with 512MB RAM and dual 10GB hard disks, running *Microsoft Windows XP Professional SP2.*

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

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 discuss all aspects of electronic crime and ways to combat it. For more details see http://www.antiphishing.org/ecrimeresearch/index.html.

RSA Conference Europe 2007 takes place 22–24 October 2007 in London, UK. Conference tracks include 'developing with security', 'deployment strategies', 'enterprise defence', 'hackers & threats', 'policy & government', and 'security solutions'. Discounted registration rates apply until 21 September. For full details see http://www.rsaconference.com/2007/europe/.

Black Hat Japan, takes place 23–26 October 2007 in Tokyo, Japan. Online registration is now open. For more information see http://www.blackhat.com/.

The CSI 34th Annual Computer Security Conference will be held 5–7 November 2007 in Washington, D.C., USA. The conference program and online registration are now available at 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/.

The Chief Security Officer (CSO) Summit 2007 will take place 28–30 November 2007 in Amsterdam, the Netherlands. The summit, entitled 'Security strategy to steer your business', offers participants the opportunity to tackle fraud management challenges in a hassle-free environment, surrounded by colleagues. A speaker panel will share direct experiences, successes, and tips gained from managing successful security projects. For details see http://www.mistieurope.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). Enquiries 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. Online registration will be available from 1 September 2007. See http://www.rsaconference.com/2008/US/.

Black Hat DC 2008 will be held 11–14 February 2008 in Washington, DC, USA. Other 2008 dates include Black Hat Europe 2008, which takes place 25–28 March 2008 in Amsterdam, the Netherlands, and Black Hat USA 2008, which takes place 2–7 August 2008 in Las Vegas, NV, USA. For details see http://www.blackhat.com/.

The 5th Information Security Expo takes place 14–16 May 2008 in Tokyo, Japan. For more details see http://www.ist-expo.jp/en/.

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

SENDER AUTHENTICATION CHECKS ON THE RISE

A report by email marketing software provider *Lyris* has revealed that use of the Sender Policy Framework (SPF) email authentication check is on the rise among ISPs. The company conducts quarterly research studies of deliverability rates for permission-based email marketing messages, and the Q2 2007 study revealed that SPF authentication checks are now among the top ten content triggers checked by ISPs to determine whether an email is legitimate. Stefan Pollard, of marketing solutions provider *EmailLabs*, says, 'This is the first time we've seen SPF checks start to creep into content filter tests, which means that receivers are starting to verify that a sender's SPF authentication record is accurate.' Pollard urged legitimate email marketers to ensure that their records are accurate and kept up to date.

EVENTS

The 11th general meeting of the Messaging Anti-Abuse Working Group (MAAWG) will take place 8–10 October 2007 in Washington D.C., USA (members only). The 12th general meeting, open to members and non-members, will be held 18–20 February 2008 in San Francisco, CA, USA. See http://www.maawg.org/.

TREC 2007, once again featuring a spam filter evaluation track, will be held 6–9 November 2007 at NIST, MD, USA. For details see http://plg.uwaterloo.ca/~gvcormac/spam.

Inbox/Outbox will take place 27–28 November 2007 in London, UK. Seminars will cover eight key email-related themes including security and content filtering. See http://www.inbox-outbox.com/.

FEATURE

FIGHTING SPAM USING TAR PITS

Tobias Eggendorfer
Independent researcher, Germany

Spam and viruses are the biggest threat to Internet usage according to a recent survey. One thing they have in common is that they can be distributed by email. For a spammer, email is the main tool, while for a virus writer, email is just one of many propagation vectors.

One of the most important considerations for spam prevention techniques is to keep the number of false positives and negatives to a minimum. A false negative is a spam message put into a user's inbox, wasting his time and increasing the risk of his accidentally deleting an important message because it got lost among hundreds of spam messages. A false positive is a 'ham', i.e. a non-spam message, that is moved to the spam folder. In a business environment this could result in the loss of business as a result of an order not getting the required attention, or it could put a company at risk of being sued for not fulfilling an order in time, thus increasing the economic risks associated with a false positive by several orders of magnitude.

Most anti-spam and anti-malware techniques are symptomatic cures for the epidemics, with few attempting to tackle the root of the problem. Anti-malware programs try to identify malware as soon as it is transferred onto a computer, be it by email, by a downloaded program file or introduced through a remote exploitable security hole. A causal therapy would be to implement an operating system with security in mind and dedicate know-how and labour time to security testing. *OpenBSD* is an example of how effective this can be.

SYMPTOMATIC THERAPY

One anti-spam technique that attempted to tackle the root of the problem was sender authentication and identification – which included Domain-ID, Sender-ID and SPF. Although well intended, it was obvious to many that these technologies would fail, because all known options break important email functionality. Email forwarding is a very important function, but it is virtually impossible with these security measures in place. With people moving around the world, travelling and

changing jobs very quickly, having access to their communication systems through any service they choose is a must. In a paper presented at the Conference on Email and Anti Spam (CEAS), a *Google* employee explained that the vast majority of *GoogleMail* users have their email forwarded from some other account to *GoogleMail*, thus complicating sender authentication. It is safe to assume that a lot of web mail service providers encounter the same problem.

But removing important functionality was not the only problem for sender authentication, another being the fact that spammers could register their domains with perfectly valid SPF records in DNS. Identifying spam using any of the sender authentication techniques is like playing Russian roulette – some will survive.

Furthermore, patents and the attempt to gain influence, power and money also became involved when these techniques first became available. It is not unlikely that this prevented a lot of developers and administrators from implementing this technology and the lack of uptake of the method also contributed to its failure.

GREYLISTING

Another approach to getting rid of spam without implementing a filter is greylisting. Greylisting takes advantage of the fact that most bulk mailers only have a limited subset of SMTP implemented. In particular, they often lack the functionality to deal with temporary error conditions on the server side. A temporary error is a failure situation that might be resolved within a short period of time without any intervention from the administrator. Examples are a user over-quota condition, temporary failure to connect to a company's LDAP directory to identify whether a user exists locally, or an overload condition on the server.

The SMTP standard has foreseen these issues and suggests that MTAs retry to send those messages for a certain period of time. In its default configuration, the quite common MTA *sendmail* is set up to retry to deliver a message for five days. However, spammers' bulk mailers do not resend messages. Therefore, greylisting systems respond to an incoming mail message with a temporary error code and will store a tuple consisting of the client's IP and the envelope from and to of the mail message. If, after a certain period of time, this client reconnects and tries to deliver a message with a corresponding tuple, the mail will be accepted. According to the proponents of greylisting, this system reduced spam by 80% when it was first introduced.

Now, however, the efficiency of greylisting is dropping, as bulk mailers have learned to try to resend their messages.

Greylisting.org explains greylisting's new main advantage:

> 'This delay in new sender contacts also gives you a lot of extra power. This may be an hour, but in this hour there is a large chance that the mass mailer/spammer has been identified by the more conventional anti-spam software. Thus, when he retries it, is likely that we will know him for what he really is!'

Obviously, there are other ways to leave a message waiting until the spam filter has been updated.

Also, some providers claim that greylisting would waste their resources: each message needs to be stored for a longer period of time, thus forcing a large provider to add terabytes of storage space to accommodate those waiting messages.

Greylisting is also incompatible with a setup often found within larger providers. If instead of one outgoing mail server a server farm is used, then often the resend attempt comes from a different IP than the original. Greylisting proponents argue that this should not be an issue, as there are only a few relevant providers using this technology and those can be whitelisted manually. This might be feasible for a small environment with limited worldwide contacts, but not for an international environment.

Furthermore, there are a lot of mail services that are incompatible with greylisting, e.g. messages sent by *Yahoo Groups* didn't make it to the recipient if he used greylisting. Once again, the supporters of greylisting came with the solution of whitelisting those IPs and also explained that those companies must be using a broken MTA. But a problem introduced by a new technology can't be an existing system's fault – a new safety feature should be compatible with existing technology.

Taking these issues into account, and adding the ever decreasing effectiveness of greylisting, it seems that greylisting is not the solution to the spam problem, either.

PREVENTING SPAM

By looking at the problems with some of the existing attempts to reduce spam, we can draw up a list of a number of requirements for a new concept. The most important are: compatibility, efficiency, and, last but not least, free availability without involving patents.

One method to prevent spammers from spamming is simply to prevent them from collecting email addresses. One of their sources is malware-infected computers. Some malware searches the local hard disk for email addresses. Obviously, in order to prevent this method of address harvesting we need to prevent the malware from getting onto the machine – a problem that is not yet fully resolved and beyond the scope of this article.

However, another major source of email addresses is the Internet, both the Usenet and the World Wide Web. The latter has become more important over the years and with the advent of forums and newsgroups being mirrored to it, is more promising for an email harvester.

One thing that can be done to prevent the harvesting of email addresses from a web page is to obfuscate the addresses in such a way that they are unreadable to harvesters, but compatible with any browser technology and barrier-free. The latter requirement is not met by the often suggested use of a graphical representation of the email address. We did some analysis on the efficiency of obfuscation methods and found a rather simple one to be very effective. Simply by adding a white space after every other letter, the address is blown up and it becomes very difficult to find it automatically in any document, e.g. finding 'us er @ ex am ple. co m' in this text is not trivial, as the left and right boundaries are hard to identify.

Ongoing research confirms this still to be a secure and efficient way of obfuscating an email address. We therefore developed an *Apache* module that obfuscates addresses on the fly during output using this method, thus making secure obfuscation a matter of installing and enabling it in the *Apache* configuration file.

HTTP TAR PIT

Obfuscation is a rather egoistic approach: it helps the user protect his inbox, but it is of no use to the wider Internet community. Therefore, we looked for a way to stop harvesters while they are in the process of collecting email addresses. To do so, we developed an HTTP tar pit.

In brief, the HTTP tar pit creates random web pages linking back to itself and thereby traps the harvester in an infinite loop. Obviously, the links need to be different every time, because no decent spider would return to a page it has visited before. Our tar pit creates random file names with random, yet plausible, file extensions. The server is configured to redirect every request for one of those random URLs to the tar pit script. This is done by using the ErrorDocument-method of *Apache* and a reset of the HTTP status code from '404 Document not found' to '200 OK'.

As some harvesters implement a maximum link depth on certain domains in order to avoid endless loops, we use DNS wild cards to create random sub domains. This resets the harvester's link counter and thus keeps him in the tar pit. To further increase the effects, we run several interconnected tar pits on multiple machines with multiple IPs, thereby further obfuscating their existence.

Because the HTTP tar pit offers more new links to itself than there are new links on an average web page, the tar pit's links pollute the list of pages to visit that is maintained by the harvester. Thus, the more often the tar pit is visited, the more efficient it gets until it takes up almost 100% of the harvester's links to visit.

Besides catching spammers in an endless link loop, the tar pit also stutters each byte slowly to the client to delay the communication further and catch the harvester for even longer. This stuttering needs to be carefully adjusted to the time-outs harvesters use, because they should not disconnect too quickly, but stay in the trap.

When implementing this HTTP tar pit, we also took into consideration the 'good' spiders used by search engines. If they were trapped, their operators might even sue for compensation. Fortunately, the W3C Robots-Exclusion-Standard offers a method to tell spiders not to analyse certain pages. Therefore, we set up a robots.txt, which is ignored by almost all harvesters, to protect the good spiders. If any search engine spider were to ignore this information, we would not be liable for them becoming trapped.

An argument often put forward against using robots.txt is that it would be easy for harvesters to start conforming to the same standard. But if they did, this would just mean that preventing harvesters from collecting an email address would be as simple as adding the specific page containing the mail address to robots.txt's list of pages not to visit.

ADDING AN SMTP TAR PIT

With a view to being an attractive tar pit to spammers, the tar pit should offer email addresses, since harvesters output every new address found to their user interface. This serves as a kind of progress meter, and it would stop listing new addresses as soon as the harvester was mostly visiting links within the tar pit. The human operator of the harvester would notice its reduced effectiveness, start investigating it and ultimately find out that he has run into an HTTP tar pit and blacklist its URL.

However, just printing out random email addresses from the tar pit is not a good solution, as this could lead either to spamming random genuine addresses or to bounce spam if the addresses are nonexistent and the sender address was forged. We therefore decided to set up an SMTP tar pit and list addresses that point to this tar pit. The SMTP tar pit adds another level of frustration to the spammer.

Like its HTTP counterpart, an SMTP tar pit also delays communication between the spamming client and the tar pit server. With SMTP, creating endless loops of links is impossible, but by stuttering the server's responses byte by byte and adding artificial overhead by creating extra long responses with lots of SMTP's continuation lines, the slow

down is remarkable. On a regular connection, delivering a message usually takes a matter of a fraction of a second, whereas on a SMTP tar pit it might take up to one hour.

Supporters of SMTP tar pits therefore claim that they would block a spammer's sending process and thereby protect the Internet from a spam run. Although this is not true, as bulk mailers are able to connect to multiple servers at the same time and are tar pit-aware (i.e. disconnect quickly if they recognize an SMTP tar pit), in our setup this did not matter, because we just needed an SMTP server to take care of the email addresses published by the HTTP tar pit.

Adding that SMTP tar pit to the HTTP tar pit increased the HTTP tar pit's efficiency by several orders of magnitude. We found harvesters staying in the tar pit and looping infinitely for several weeks and making hundreds of thousands of visits during that time.

IDENTIFICATION OF HARVESTERS

Since most visits to a tar pit are by email harvesters, the tar pit also offers a simple method to identify the IP addresses from where harvester activity occurs. This piece of information might help in protecting other web pages: if the IP addresses the harvesters use are available to those web servers, they could block access to the harvesters.

To do this, we built another *Apache* module, this time an input filter that looks up the client's IP in a database of known harvesting IPs populated by our (by then distributed) network of combined HTTP and SMTP tar pits. If an IP is listed there, access to the protected page is forbidden and an error message is displayed. The harvester is then prevented from collecting email addresses from this page, because it cannot access it.

Since it is possible that humans might accidentally click into a tar pit, we decided to impose the website ban only for a certain amount of time, depending on the frequency of visits to the HTTP tar pit. We also chose to reassess the listing of the IP address after 24 hours, as we realized that a lot of harvesting on our tar pit network was done from dynamically assigned IP addresses. Blocking those IPs for longer than absolutely necessary might be annoying for the user to whom the is IP assigned after it has been used for harvesting – even though, from an anti-spammer's perspective, it would be helpful if the harvesting activity were to result in complaints to the provider.

Currently, however, this harvester identification does not offer 100% protection, because the harvester has to have visited a tar pit prior to a protected page. Obviously, this is out of our control.

SMTP TAR PIT SIMULATOR

By doing research into SMTP tar pits, we found that spammers would quickly disconnect if they realized the remote server was a tar pit. We decided to take advantage of this behaviour by setting up an SMTP tar pit simulator, first on a bridge and later as a patch for the widely used mail server *sendmail*.

Our tar pit simulator behaves like an SMTP tar pit for a certain amount of bytes sent, i.e. it will stutter the first 60–120 bytes to the client slowly, and will then open up the connection to full speed. Our tests showed that approximately 80% of the connections spammers made to our servers were dropped during the stuttering period. We did not find any ham sender that disconnected – meaning that, to our knowledge, the system does not generate any false positives.

Although it is not a perfect solution, it reduces the workload of a spam filter on the mail server significantly, either allowing it to do more computing-intensive mail analysis or to be run on cheaper hardware. Reducing spam by 80% would mean bringing spam levels back to those of 2001.

The advantage of the tar pit simulator is that spammers could only adapt to it if they accept a higher risk of being trapped in a real tar pit. So it is an economic decision for them as to whether to disconnect quickly to avoid being trapped in real tar pit or whether to wait for longer in case it is 'only' a simulator. The longer they wait, the higher their loss if they end up in a real tar pit.

Therefore, the more unpredictable the simulation time, the harder it is for spammers to adapt. We suggest that this time be randomized.

A prerequisite for the tar pit simulator to work is the existence of SMTP tar pits in the Internet, even though they themselves are not effective in fighting spam. This is another reason to combine the HTTP tar pit described above with an SMTP tar pit and not a plain mail server.

CONCLUSION

To sum up, both SMTP and HTTP tar pits offer interesting ways of getting rid of spam. Our HTTP tar pit prevents the collection of email addresses from web pages and it helps to identify harvesting IPs. This means that access to web pages can be blocked dynamically, thus protecting them from harvesters. Finally, a simulated SMTP tar pit might reduce the amount of spam a mail server has to deal with by 80%. This would provide a significant relief to the local mail infrastructure.