Man, Myth, Malware and Multi-Scanning

David Harley, ESET N. America

Julio Canto, VirusTotal/Hispasec Sistemas

CFET 2011

5th International Conference
on Cybercrime Forensics Education and Training

About the Authors

Consultant and author David Harley[1] has been working closely with the security company ESET[2], where he holds the title Senior Research Fellow since 2006, and a former Director of the Anti-Malware Testing Standards Organization[3]. Julio Canto is Senior Software Engineer at Hispasec Sistemas[4], who are the guys behind VirusTotal[5], probably the best known multi-engine online scanning service. They have wanted to do a paper on this topic for quite a while, since cooperating on a problem with a security organization misusing the service[6] as a substitute for real testing, and were delighted when CFET[7] gave them the chance to do so.

Abstract

Malware multi-scanning: everybody's doing it. AV companies use batteries of competitor products for comparative analysis and other laboratory procedures. Blackhats are increasingly likely to use internal or third-party "black" laboratory resources for the testing of malware tweaked to increase resistance to anti-malware analysis and forensics, as the blackhat economy strengthens and parallels conventional business models. Public multi-scanner sites intended for the evaluation of the risk from individual files are also used and misused for many purposes, such as:

- Indirect distribution and gathering of samples
- The estimation and guesstimation of malware prevalence and of public exposure to risk from "undetected" malware
- The "ranking" of products by detection performance, and the subsequent generation of marketing collateral
- Pseudo-validation and classification of samples by testers.

Public sites have evolved and matured to meet the different needs of anti-malware vendors, a wide range of home and end users, other security researchers, and the media. However the range of myths and misconceptions around what is and isn't appropriate use has outpaced those developments. This paper and presentation will look at the history and range of multi-scanner usage in all these contexts, but will focus primarily on the inappropriate substitution of multi-scanning for (a) performance ranking and pseudo-testing, and (b) sound sample validation and classification.

This paper will consider five key points:

- Firstly, what's out there? We consider the multiplicity of public multi-scanner sites, in-house AV resources, specialist AV community resources and blackhat resources that are currently known to be in use as an anti-forensic measure.
- Secondly, we consider the sane and sensible uses for multi-scanning, including pre-validation sample processing, in-house comparative analysis, and risk assessment of individual files at public sites.
- Thirdly, we consider the misuse of public and private multiscanner facilities for pseudo-testing: is it a good idea to use multi-scanners for product ranking by detection performance?
- Fourthly, we look at pseudo-validation, addressing the issue of automation versus avoidance in sample validation and classification
- Finally, we address the implications for the anti-malware and product testing industries.

Everybody's Doing It.

Malware multi-scanning, that is.

AV companies use batteries of competitor products for comparative analysis and laboratory procedures. Obviously they like to know how well their products compare with others, in detection as in other respects. Marketing departments are particularly interested in such data, but mainstream companies usually refrain from using them in direct marketing comparisons. It's considered bad karma. Besides, who in their right mind believes what one security company says about other companies' performance, based on its own tests? Admittedly, that doesn't stop some companies – especially those on the fringes of the AV sector – from presenting their internal threats as authoritative anyway.

Virus labs – or anti-malware labs, to be more precise – are also likely to use multiple scanning for pre-filtering samples: for example, a lab might assume that a sample which is detected as something malicious by more than one scanner needs closer (possibly manual) analysis.

Other security researchers are fond of using multi-scanning as a stick to beat the AV industry over the head with: "we have this variant of SpyEye and no antivirus detects it." The vexed question of whether multi-scanning is an *authoritative* means of assessing detection performance with multi-scanning is central to this paper.

Meanwhile, Blackhats are increasingly likely to submit their own malware to internal or third-party "black" laboratory resources as part of the process of tweaking it to increase resistance to anti-malware analysis and forensics. All this is symptomatic of the ways in which the blackhat economy is strengthening in parallel with conventional business models.

Really Useful Engines

Public multi-scanner sites intended for the evaluation of the risk from individual files by checking them against multiple scanner engines are also used and misused for many purposes, such as:

- Indirect distribution and gathering of samples (though reputable sites only share samples with trusted parties)
- The estimation and guesstimation of malware prevalence and of public exposure to risk from "undetected" malware
- The "ranking" of products by detection performance, and the subsequent generation of marketing collateral
- Pseudo-validation and classification of samples by testers. Why pseudo-validation? We'll come back to that.

This is VT's own view of the ways in which its service is used[8]:

- Independent researchers and end users use it to check potential threats

- CERTs, security firms, anti-malware vendors use it for initial sample classification
- General public, independent investigators, CERTs, and security firms use it for real time distribution of samples to the wider security community.
- Legitimate software developers may use it as a check for possible false positives

Public sites have evolved and matured to meet the different needs of anti-malware vendors, a wide range of home and end users, other security researchers, and the media. However those developments have been outpaced by a whole range of myths and misconceptions around what is and isn't appropriate use. We'll focus here on the inappropriate substitution of multi-scanning for performance ranking and pseudo-testing, and for sound sample validation and classification.

Here's an overview of the VirusTotal service.

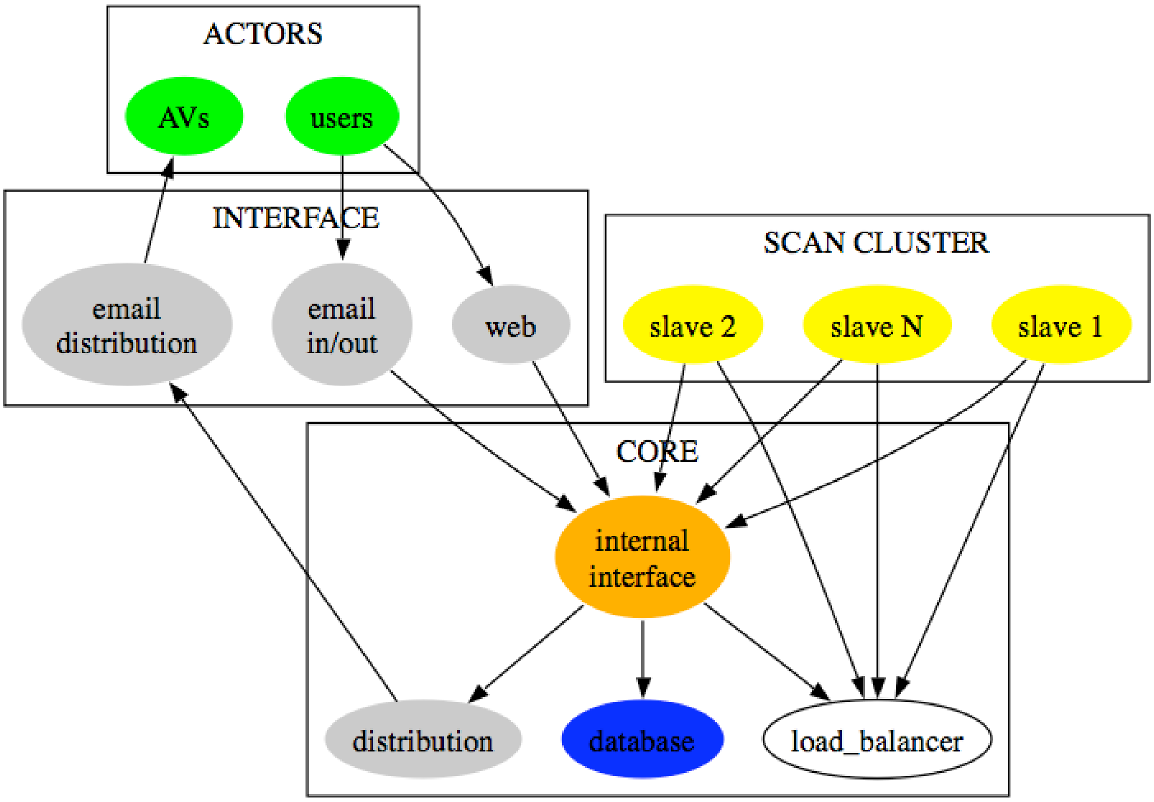


Figure 1: Virus Total General Design

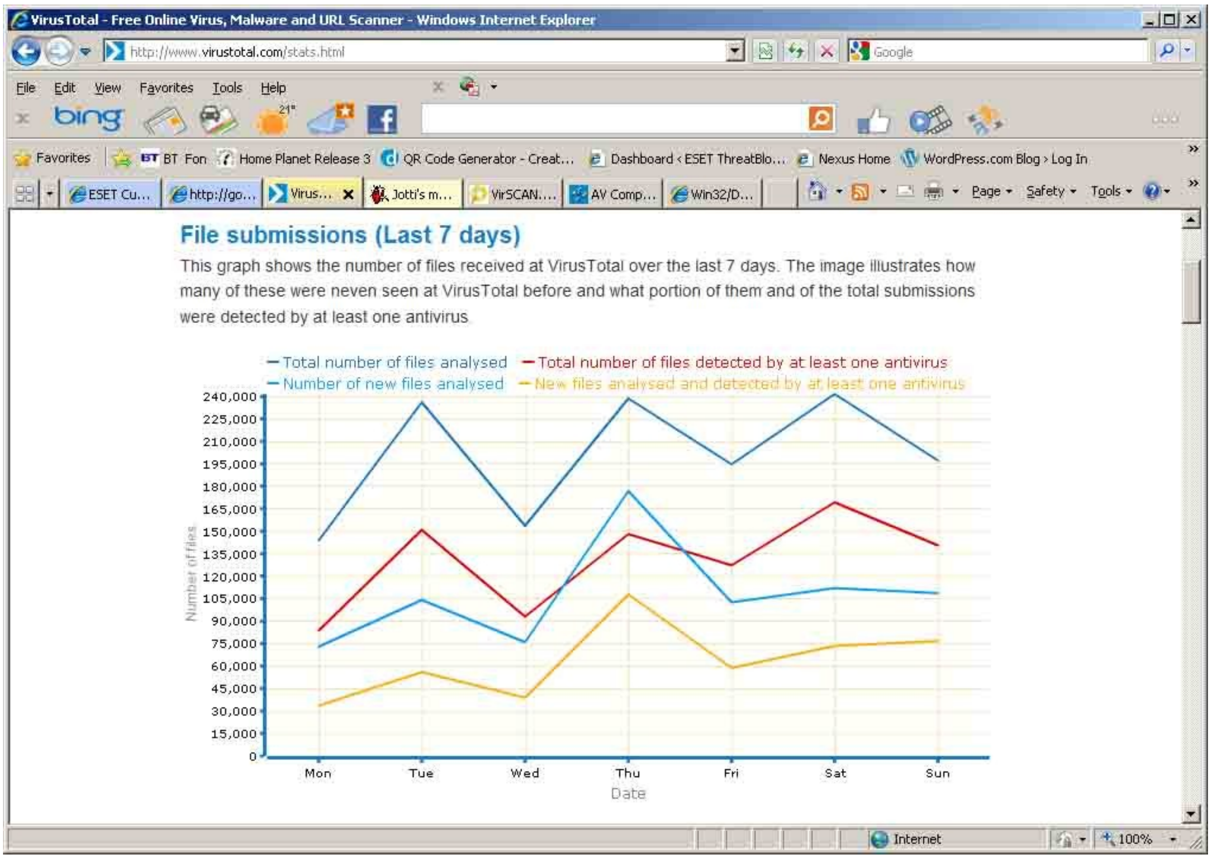
VirusTotal isn't the only multi-engine online file-scanning service – Jotti[9] and Virscan[10] come to mind as alternatives – but it's probably the best known, and has particularly good cooperative relationships with the AV industry.

- It was made public in June 2004, when it processed about 5,000 samples. In October 2009, it processed about well over 3 million samples
- It started with 11 engines, now it was using 43 as of July 2011.

- VT started with a single computer, and moved on to a cluster of machines
- It sends tens of thousands potentially infected files to the AV industry every day so they can study them and update their products as necessary. Virscan.org and Jotti also share samples with the industry.
- Multi-user sites may also offer other off-the-radar services that the AV companies find useful, too.

Let's think about the implications of that sharing process.

Figure 2 is a screenshot taken a few days before the presentation showing the number of files received over the last seven days. It is, obviously, a lot of files. The proportion of "innocent" to malicious files out of all those submitted isn't known unfortunately: clearly, the percentage of unequivocally malicious files affects in some sense the "value" of the samples to a vendor. Since you can't assume that every sample (or anywhere near every sample) is malicious, you have to factor in the time and resources spent filtering. (The same applies, in principle, to all sample-gathering resources of course.) Still, we are talking about a very useful resource. And criminals are aware of its value.



File submissions (Last 7 days)

This graph shows the number of files received at VirusTotal over the last 7 days. The image illustrates how many of these were neven seen at VirusTotal before and what portion of them and of the total submissions were detected by at least one antivirus

VirusTotal - Free Online Virus, Malware and URL Scanner - Windows Internet Explorer
http://www.virustotal.com/stats.html
Google
File Edit View Favorites Tools Help
bing
Favorites
BT Fon
Home Planet Release 3
QR Code Generator - Creat...
Dashboard ‹ ESET ThreatBlo...
Nexus Home
WordPress.com Blog › Log In
ESET Cu...
http://go...
Virus...
Jotti's m...
VirSCAN...
AV Comp...
Win32/D...
Page
Safety
Tools
— Total number of files analysed
— Total number of files detected by at least one antivirus
— Number of new files analysed
— New files analysed and detected by at least one antivirus
240,000
225,000
210,000
195,000
180,000
165,000
150,000
135,000
120,000
105,000
90,000
75,000
60,000
45,000
30,000
15,000
0
Number of files
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Date
Internet
100%

Figure 2: A Week of VT File Submissions

Figure 3 is an example of the sort of statistics that VT used to provide regarding proportions of innocent files to apparently infected files. Red means detected by at least one engine, blue stands for no detection by any of the N Engines. I've certainly noted days in the past where there were more blues than reds. Of course, you might consider that's just a measure of how ineffective AV is, but that would be an unsafe assumption, in my (DH's) humble and impartial opinion. ☺

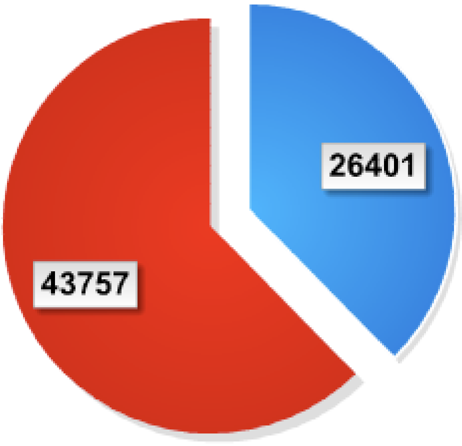


Figure 3: Proportion of Received Samples Diagnosed as Infected

The list below shows the ten most-queried files, rendered as MD5 hashes, on the day before these data were downloaded. If you have a suspicious file, you can check it on VirusTotal by generating the hash and submitting that rather than the file, using md5, sha1 and sha256. Though in fact, md5 is considered problematical nowadays because of the potential for hash collisions, and like other organizations, VT is moving away from it. If you're a real malware geek analyst you may be able to identify some of these from their hashes, but some of us are merely human...

- f922d46646a9ee75f79d09eb02b3964c
- 611c916c9a3847b08d104d57f24ec97e
- ce338fe6899778aacfc28414f2d9498b
- bb7df04e1b0a2570657527a7e108ae23
- a82c6a51306dc92d1788209e7b65efe9
- d89746888da2d9510b64a9f031eaecd5
- 0829f71740aab1ab98b33eae21dee122
- cd44bcdce1dff168cfcfef14a4eb572b
- 4bd992dae2dbbe35b4ec51458103f729
- fb480a631062375a7e1d1db8da01f5fe

The final results of the analysis of a sample are received, so one can see the evolution as detection increases. That's different to Permalinks, which can mislead over time. ("You don't detect this: I can see it from the VT permalink." "But that was a week ago!")

There's a limit to how many requests per minute can be made, so as to limit potential abuse.

Reading the Results

VT is a complicated service to deliver, and interpreting the results isn't always straightforward. These are obvious issues (there are some others we've omitted, as this isn't just about VT, but multi-scanning in general). Some engines have very paranoid heuristics: at ESET, there is a standing joke that a certain product's principle heuristic is that if it's executable, it should be flagged as suspicious. Packers and obfuscators are a problem for all AV companies: even packers that were obviously intended from the start for malicious use sometimes turn up wrapped around legitimate software. That makes it awkward when companies use the presence of a packer as a heuristic. If they get it wrong it's a false positive. If they play safe and don't flag it, they look bad next to a product that *does* flag it.

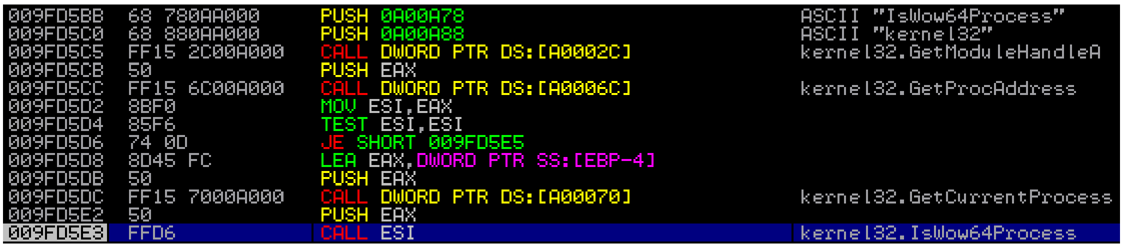
Then there are programs that you can't describe unequivocally as malware, but that may nevertheless be described as 'Possibly Unwanted' or 'Possibly Unsafe' Applications (other vendors may use slightly different terminology such as Possibly Unwanted Software – PUS – or Possibly Unwanted Programs - PUPs).

'Unwanted' covers all sorts of greyware, adware, whatever. "Possibly Unsafe" is a category used at ESET to cover legitimate tools that may be misused by malware[11].

Down a Branch Line: TDSS

Let us introduce you to TDSS, now somewhat notorious as the indestructible botnet, much to the embarrassment of one of ESET's competitors.

Figure 4 is actually a snippet of dropper code that checks on the type of operating system that it's running in, but it could have been anything, because this isn't about the specifics of TDL4 code[12].



```
009FD5BB  68 780AA000     PUSH 0A00A78                        ASCII "IsWow64Process"
009FD5C0  68 880AA000     PUSH 0A00A88                        ASCII "kernel32"
009FD5C5  FF15 2C00A000   CALL DWORD PTR DS:[A0002C]          kernel32.GetModuleHandleA
009FD5CB  50              PUSH EAX
009FD5CC  FF15 6C00A000   CALL DWORD PTR DS:[A0006C]          kernel32.GetProcAddress
009FD5D2  8BF0            MOV ESI,EAX
009FD5D4  85F6            TEST ESI,ESI
009FD5D6  74 0D           JE SHORT 009FD5E5
009FD5D8  8D45 FC         LEA EAX,DWORD PTR SS:[EBP-4]
009FD5DB  50              PUSH EAX
009FD5DC  FF15 7000A000   CALL DWORD PTR DS:[A00070]          kernel32.GetCurrentProcess
009FD5E2  50              PUSH EAX
009FD5E3  FFD6            CALL ESI                            kernel32.IsWow64Process
```

Figure 4: TDSS

ESET's researchers in Russia have paid a lot of attention to the evolution of TDSS[13]: not only its technical properties (which are fascinating), but also its distribution. TDL3(+) was distributed by a group called Dogma Millions using a Pay Per Install (PPI) model, and subsequently Gangsta Bucks (Figure 5) took up the torch for TDL4, using a very similar model.

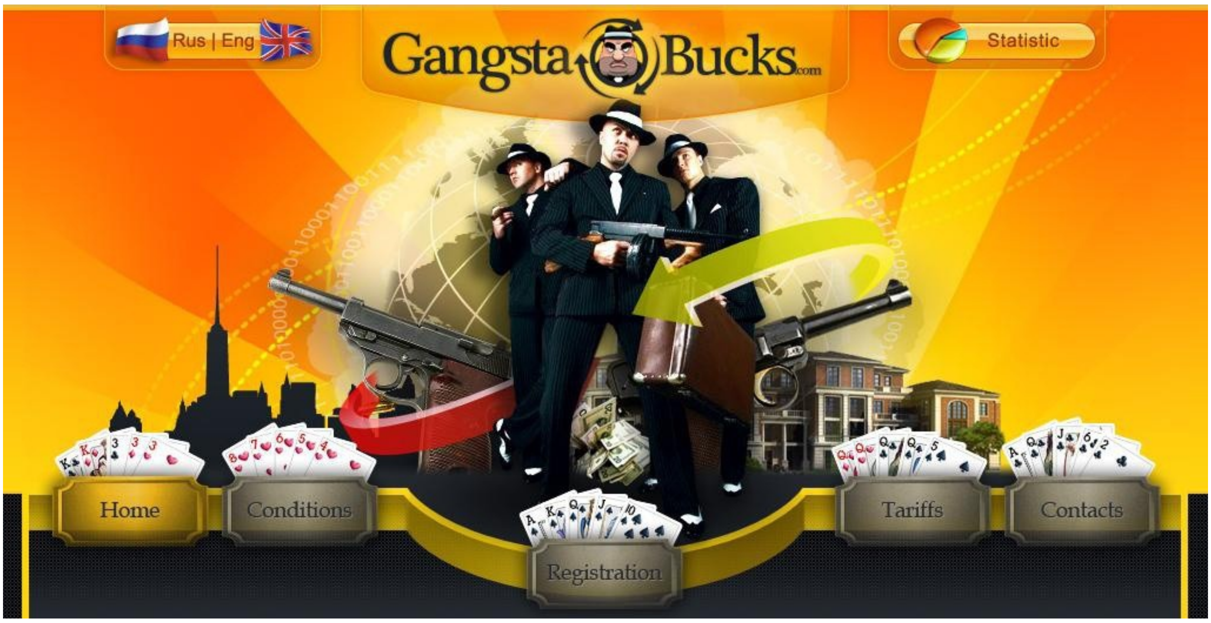
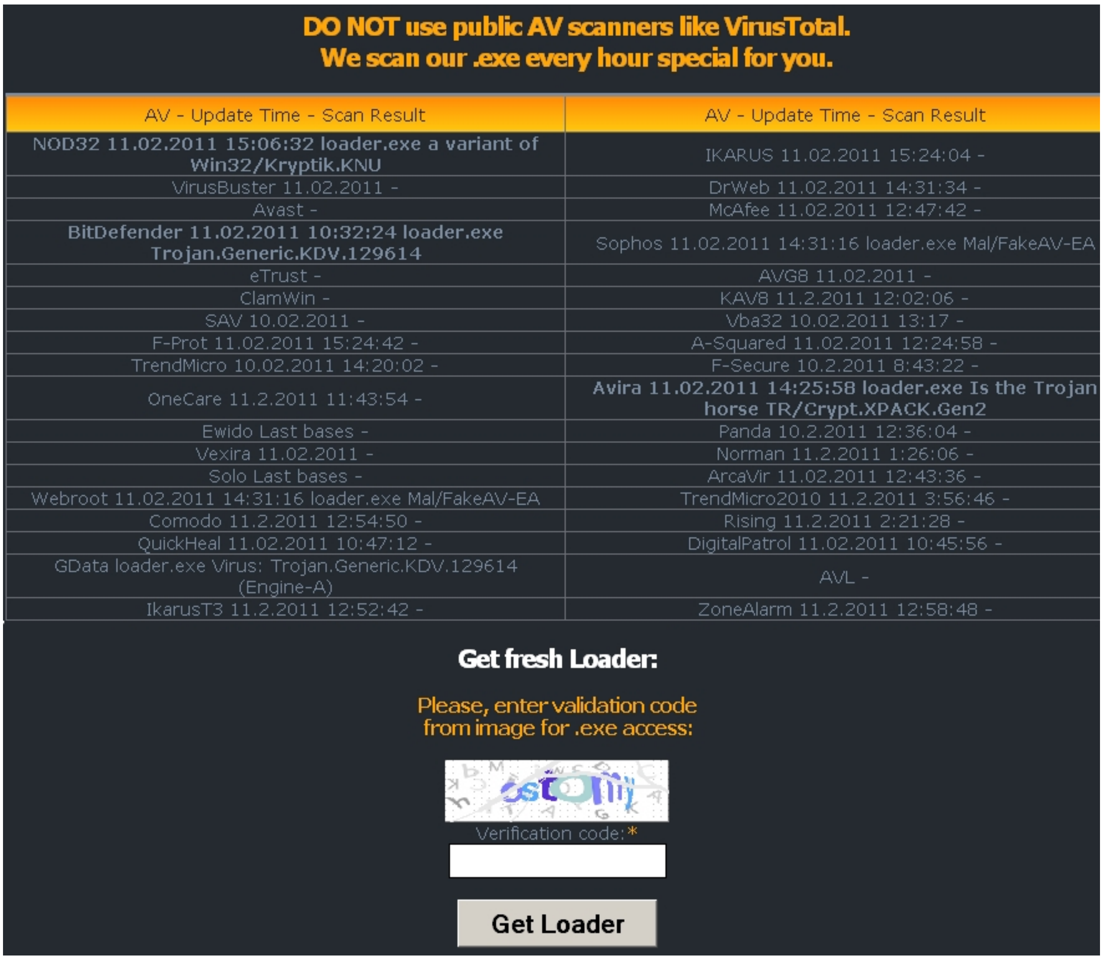


Figure 5: Gangsta Bucks

An authorized partner was able to download the current version of the Trojan downloader and also to receive statistics relating to detection by antivirus software. As soon as the downloader was known to be detected by most antivirus software products, the partner received the new “fresh” (repacked) version of malware to distribute.

Now note the instruction at the top of the screenshot in Figure 6. Affiliates/partners are fined, or kneecapped or something if they disobey. What you’re looking at here is something like Virus Total for blackhats. There are services whose selling point is that they *don’t* share samples with security companies, making them a useful resource for criminals wanting to test current detection of new variants/sub-variants/repacks.



DO NOT use public AV scanners like VirusTotal.
We scan our .exe every hour special for you.

AV - Update Time - Scan Result	AV - Update Time - Scan Result
NOD32 11.02.2011 15:06:32 loader.exe a variant of Win32/Kryptik.KNU	IKARUS 11.02.2011 15:24:04 -
VirusBuster 11.02.2011 -	DrWeb 11.02.2011 14:31:34 -
Avast -	McAfee 11.02.2011 12:47:42 -
BitDefender 11.02.2011 10:32:24 loader.exe Trojan.Generic.KDV.129614	Sophos 11.02.2011 14:31:16 loader.exe Mal/FakeAV-EA
eTrust -	AVG8 11.02.2011 -
ClamWin -	KAV8 11.2.2011 12:02:06 -
SAV 10.02.2011 -	Vba32 10.02.2011 13:17 -
F-Prot 11.02.2011 15:24:42 -	A-Squared 11.02.2011 12:24:58 -
TrendMicro 10.02.2011 14:20:02 -	F-Secure 10.2.2011 8:43:22 -
OneCare 11.2.2011 11:43:54 -	**Avira 11.02.2011 14:25:58 loader.exe Is the Trojan horse TR/Crypt.XPACK.Gen2**
Ewido Last bases -	Panda 10.2.2011 12:36:04 -
Vexira 11.02.2011 -	Norman 11.2.2011 1:26:06 -
Solo Last bases -	ArcaVir 11.02.2011 12:43:36 -
Webroot 11.02.2011 14:31:16 loader.exe Mal/FakeAV-EA	TrendMicro2010 11.2.2011 3:56:46 -
Comodo 11.2.2011 12:54:50 -	Rising 11.2.2011 2:21:28 -
QuickHeal 11.02.2011 10:47:12 -	DigitalPatrol 11.02.2011 10:45:56 -
GData loader.exe Virus: Trojan.Generic.KDV.129614 (Engine-A)	AVL -
IkarusT3 11.2.2011 12:52:42 -	ZoneAlarm 11.2.2011 12:58:48 -

Get fresh Loader:

Please, enter validation code
from image for .exe access:

Verification code:*

Figure 6: Instructions to Affiliates

A CSI Digression: Sidetracked by Forensics

This is an extract from a list of detections reported by an ESET scanning product used by a forensic investigator examining a PC in the course of an active investigation.

- \Local Settings\Temp\db.exe - a variant of Win32/TrojanDownloader.VB.OCD Trojan
- \Local Settings\Temp\srfto8sd44.exe - a variant of Win32/Adware.Coolezweb.AZ application
- \Program Files\AskSBar\SrchAstt\1.bin\A2SRCHAS.DLL - Win32/Toolbar.AskSBar application
- \WINDOWS\system32\mswjr.exe - Win32/TrojanClicker.VB.NIM trojan
- \WINDOWS\system32\msxm192z.dll - a variant of Win32/PSW.WOW.NNZ trojan
- \WINDOWS\system32\ytasfwecxsruue.dll - Win32/Olmarik.LE trojan
- \WINDOWS\Temp\fyrfaydxcp.exe - a variant of Win32/Kryptik.AFI Trojan

While the investigator wasn't able to share any information about the case, we assume that the concern was that the suspect might try to use some version of the Trojan defence in court[14].

Basically the analyst wanted information relating to the types of infections found on a computer involved in an on-going police investigation, wanting to know if the *malware* could be responsible for incriminating files on the PC, rather than the *suspect* being responsible. He told ESET which version he was using, including the engine version number and signature database and date, and the number of infected files it had flagged. He'd checked the ESET Threat Center[15] and googled the names of the malware as flagged by NOD32 but wanted definitive answers as to their effects.

This isn't really the right question to ask: it's like trying to substitute a handful of more-or-less generic descriptions for real (dynamic) forensic analysis and perhaps an expert witness. It wasn't clear whether the system was actually infected, or simply contained malware, and whether there was active malware on the system from which the image was copied, lacking detail such as registry keys which might indicated better the target system's infection status. Clearly, if the files hadn't executed, but were simply sitting there passively, the Trojan defence may not be viable. No execution, no malicious action to blame any illegal downloads on. In fact, it's not unknown for criminals to put some malware onto a system specifically in the hope of using the Trojan defence.

Unfortunately, it's not possible, in principle, to give an authoritative answer on what a specific binary does on the basis of the detection name[16] used by most modern malware. One of the reasons for that is that individual binaries are often single components of a complex attack executed remotely, using the compromised computer for purposes which are liable to change over time[17]. The detection names in the previous slide are too generic to identify a specific binary.

It's for this reason that AV informational databases are, from a forensic point of view, not very helpful. Even if anti-malware laboratories had the resources to document every malware sample they process (hundreds of thousands of unique binaries are received daily), listing every possible payload in detail would not be feasible.

More often than not, the only way to establish exactly what the binaries in question do is by dynamic analysis of the malware in situ, and even that wouldn't necessarily say what it would have done at another point in time, or even on the machine from which the image was taken (as opposed to the one used for analysis).

A virus lab might, given time, give reasonably accurate information on the behaviour *common* to or *commonly found* in malware that is associated with those detection names, but the analysis could still only be approximate, since it can only be based on files with the same detection name, which are likely to have broadly similar functionality. However, some detections are too generic for a description to be much help. That's because they're often based on the programmatic behaviour of the malware rather than the payload mechanisms or even the family relationships between binaries from the same source.

It might have been different if it were possible to obtain the infected files from the target system. A copy of the disk image would be even better, but sometimes the nature of the investigation makes that infeasible or at least very complicated. The accuracy of any analysis in the lab would, of course, depend on urgency, and would be affected by the kind of information needed.

This was David Harley's rough summary of what the malware listed above was (some content shared with the investigator has been redacted):

- \Local Settings\Temp\db.exe - a variant of Win32/TrojanDownloader.VB.OCD Trojan Payload: downloads another file from a malicious site.

- \Local Settings\Temp\srfto8sd44.exe - a variant of Win32/Adware.Coolezweb.AZ application: Adware. Displays indeterminable 3rd-party advertising.

- \Program Files\AskSBar\SrchAstt\1.bin\A2SRCHAS.DLL - Win32/Toolbar.AskSBar application: Probably greyware/possibly unwanted toolbar.

- \WINDOWS\system32\mswjr.exe - Win32/TrojanClicker.VB.NIM Trojan Probably accesses a remote resource, very possibly for advertising/malvertising

- \WINDOWS\system32\msxm192z.dll - a variant of Win32/PSW.WOW.NNZ Trojan: Probably a World of Warcraft password stealer

- \ \WINDOWS\system32\ytasfwecxsruue.dll - Win32/Olmarik.LE Trojan; \WINDOWS\system32\ytasfwwientxtq.dll - Win32/Olmarik.KW Trojan: associated with TDSS (Alureon).

- \WINDOWS\Temp\fyrfaydxcp.exe - a variant of Win32/Kryptik.AFI Trojan: the Kryptik detection is programmatic, based on certain characteristics of the code: the payload could be practically anything.

Detection Pseudo-Testing

Let's consider the misuse of public and private multiscanner facilities for pseudo-testing: is it a good idea to use multi-scanners for product ranking by detection performance?

When we were preparing this paper, we came across a Wikipedia entry on VirusTotal[19] (which Hispasec didn't put there). At the time it stated that VT "Free antivirus testing". VT is *not* a tool for AV testing, and it *especially* isn't designed for comparative testing. In fact, we changed the entry to the rather more accurate "Free checking of suspicious files using multiple antivirus engines." It is, of course, perfectly possible that some monkey with a laptop keyboard will change it back, but it hasn't happened so far.

Stealth Testing

This is a little PR exercise from a security company, clearly suggesting that VirusTotal had carried out a test for them and that they'd wiped the floor with every AV company whose engine VT was using in 2007. This is was a "test" so secret, VT itself had no idea it was happening…

> *"… independent security-industry benchmark website VirusTotal.com attempted to simulate a malicious attack using a long-known source of malicious code on computers. Competing with 32 rivals, only [XXX] detected and blocked the malicious code in VirusTotal's tests..."*

Bernard Quintero responded in a blog entry[18] that:

- VirusTotal had not conducted any experiment or test related to AV comparative analyses.

- VirusTotal had no notice whatsoever of the malicious code they refer to in this piece of news.
- VirusTotal had never tested nor tried XXX's security solutions.

He said: “All anti-malware products have detection problems due to the tremendous proliferation and diversification of malware nowadays. Likewise, any product may detect a new sample on its own, either because of its heuristics or because they are the first ones to generate a specific signature. This is why it seems totally inadequate and opportunistic to claim the superiority of a product based on the result of a sole malware sample.“

He went on to make some very telling arguments as to why Hispasec specifically advises against VT’s use as a tool for comparing anti-malware scanner performance.

“VirusTotal was not designed as a tool to perform AV comparative analyses, but to check suspicious samples with multiple engines, and to help AV labs by forwarding them the malware they failed to detect.

To use VirusTotal to perform AV comparative analyses involves many implicit methodological errors :

- VirusTotal AV engines are command-line versions, so depending on the product, they will not behave quite like the desktop versions: for instance, in cases when desktop solutions use techniques based on behavioural analysis and are augmented by on personal firewalls that may decrease entry points and mitigate propagation, and so on.
- - In VirusTotal, desktop-oriented solutions coexist with perimeter-oriented solutions; heuristics in this latter group may be more aggressive and paranoid, since impact of false positives is less visible in the perimeter.”

It’s hard to improve on that, so we won’t try.

In general, it is not an easy task to perform a responsible and reliable AV comparative analysis; it requires a selection of malware/malicious URLs that is representative, statistically valid, and correctly classified[20]. Besides, for the correct evaluation of desktop AV products, it would be necessary to execute those samples one by one in real environments with each of the resident products to see their detection and prevention capabilities. This is an example of what AMTSO calls ‘whole product testing’[21].

How a “Test” Can Be Independent

Not to mention apparently impartial, yet (Virust)otally : useless

- Sample set: found, presumed malicious objects (honeypots, honeynets, mailboxes)
- Methodology: files submitted to Virus Total
- Validation: files submitted to Virus Total…

This is that horrible example of inappropriate use I mentioned earlier[6]. A reputable security organization ranked products according to their detection capability. But they didn’t actually test anything themselves. They collected some samples from their honeypot and submitted them to the

VirusTotal website, where they were submitted to a battery of command-line scanners, and used the results to rank the ‘Most Effective Antivirus Tools Against New Malware Binaries’.

SRI no longer ranks products by spurious detection totals. But we should talk a bit about why it was such a bad idea[22], because there are other organizations and companies doing somewhat similar ‘analysis’ using VT or another public site.

This *isn’t* a problem with VirusTotal: It’s an inappropriate expectation, based on a failure to gather all relevant information. These aren’t really “true positives”: they’re simply samples that one or more companies have identified as possible (“suspicious”) or actual malware: in other words, the companies that are likeliest to flag false positives are also the companies likeliest to score higher with this methodology.

A product may use advanced behavior analysis, without flagging a file as malware just because a runtime packer has been used, for example, because that doesn’t in itself prove that the file is malicious. In fact, files submitted to VT and forwarded to AV companies are *not* necessarily malicious.

Because VirusTotal uses on-demand command-line scanners (what is often called static analysis), a product that uses dynamic behavior analysis using emulation or a virtual machine *could* be seriously disadvantaged. Because the scanning in use relies largely on signature detection, heuristic analysis that involves allowing the malware to attempt to execute (in a safe virtual environment, of course) is not necessarily invoked (depending on the product), so the scanner doesn’t recognize malware that in the *real* world it *would* have recognized. Actually, it’s not quite that simple: I’ll expand on that shortly.

- Best result: several scanners identify a known malicious file. The chances are that’s accurate, though the industry has had problems in the past with “cascaded false positives”, where other companies have picked up one company’s FP without checking it independently, or haven’t checked it properly.

- A less good result: no scanner identifies it as known malicious. That sounds reassuring, but doesn’t prove lack of malice: as they say in the Scottish justice system, it’s “not proven”. As those who sell other kinds of solution are all too keen to remind us, anti-virus detection rates with new or repacked malware are nowhere near 100%.

- Least good (worst case): a single detection might be one company ahead of the curve, but might be a false positive. If 2-3 scanners identify it as malicious, that likelihood decreases – “more is better” – but…

- In many cases, if you actually check the identifications, they’re some variation on “suspicious”. But you already knew that the file was suspicious: that’s why you submitted it. ☺

We’re being a little disingenuous here. Suspicious means something slightly different to an AV product to what it might mean to a customer. In many cases, it means that a fairly coarse-grained heuristic has been applied and something doesn’t seem quite right. It doesn’t mean it’s wearing a mask, a striped t-shirt and carrying a bag marked “SWAG”. But even security professionals don’t necessarily realize that.

Pack Up Your Troubles

Here are some variations on packer-based heuristics used by various scanners according to context and level of paranoia:

- It’s packed
- It’s packed with a known “black” packer
- It’s packed with a custom variant packer
- It’s packed, but was already a small executable

These are all legitimate blocking criteria, but this approach isn’t exactly known malware detection. More like blacklisting a whole class of object. In fact, it’s more like the defenses that evolved in the heyday of mass-mailers, i.e. blocking attachments like .EXE, .SCR, .ZIP and so on. In other words, it’s blacklisting not of a single malicious object, or a family of malware, but a whole class of executable objects. That’s fine if you know that’s what you’re doing, but it’s not a malware-specific detection method, and potentially, it institutionalizes false positives.

This isn’t the main issue though.

Most AV scanners work like this, roughly speaking:

- Passive scanning: they check for signatures, generic signatures, passive heuristics. Roughly equivalent to static (code) analysis. This is a *very* rough equivalent to on-demand scanning, but less so than formerly, since on-demand scanning is likely to include some form of emulation or sandboxing.
- Active/Dynamic Scanning: analysis of behaviour by observing code executed in a (hopefully) safe environment.
- Rough equivalent to dynamic analysis. Emulation, Virtual Machine, sandboxing… As we’ve seen, this expands the ability of an on-demand scanner to overcome the problem of execution context (http://smallbluegreenblog.wordpress.com/2009/05/15/execution-context-in-anti-malware-testing/), so there’s less distinction between on-demand and on-access scanning. But there’s wide differentiation in detail between products, and that militates against precise comparative evaluation.

Static Testing

This approach scans a file/object without allowing it to execute, which will penalize some products unfairly. It’s very convenient testing practice, as it can be done simply by running a command-line scanner against a selection of static samples (usually files). And it’s near enough platform independent, if detection database is standard across platforms.

Dynamic Testing

Dynamic testing (especially if based on on-access scanning and taking into account the need to incorporate realistic execution context) is a more accurate evaluator of performance. But it's resource-intensive, time-intensive, and consequently expensive to implement (or at any rate, to implement properly)[23]. You can't usually run dynamic tests with two million samples unless you're running a horrendously expensive longitudinal test, which means you have to select your samples very carefully to ensure that they're "representative". More testers are moving in this direction, but they're still having to compromise with the expectations of commissioning magazines etc. who want large sample sets over very short periods. Not a realistic expectation.

- It can put products that use active/proactive techniques at a serious disadvantage.
- No execution, no behaviour to observe/analyse.
- In such a case, result doesn't reflect detection capability.

This is one of the reasons public multi-scanning sites aren't really a good tool for comparative analysis (VT, for one, was never meant to be). It's also the reason why the testing industry needs to go over to dynamic testing (as is happening with the best testing organizations) and losing its reliance on on-demand scanning.

In fact, on-demand scanning isn't always directly equivalent to static analysis. Some (in fact many) of the engines used at VT use emulation for doing some detections, emulating a given number of cycles of 'execution' of the sample to see if something suspicious is detected (NOD32, Norman, F-Prot and others afaik). That's closer to dynamic analysis than to old-fashioned static analysis. That introduces a bias in favour of those products, but of course there are other factors that cause contrary biases.

What VT – in fact, *any* multi-scanning site using on-demand scanners – *doesn't* have is behaviour analysis once the sample is executed and of course, it doesn't emulate the 'entry vector' of the sample (http, email, etc) so other measures like checking suspicious behavior in the system, or accessing bad URLs, or whatever will not work. This again introduces biases against certain products.

How not to use VT

- VirusTotal uses a group of very **heterogeneous engines**. AV products may implement roughly equivalent functionality in enormously different ways, and VT doesn't exercise all the layers of functionality that may be present in a modern security product.
- VirusTotal uses **command-line versions**: that also affects execution context, which may mean that a product fails to detect something it *would* detect in a more realistic context.
- It uses the **parameters that AV vendors indicate**: if you think of this as a (pseudo)test, then consider that you're testing vendor philosophy in terms of default configurations, not objective performance.
- Some products are targeted for the **gateway:** gateway products are likely to be configured according to very different presumptions to those that govern desktop product configuration.

- Some of the heuristic parameters employed are very sensitive, not to mention paranoid

Conclusion

VirusTotal is self-described as a TOOL, not a SOLUTION: it’s a highly collaborative enterprise, allowing the industry and users to help each other. As with any other tool (especially other public multi-scanner sites), it’s better suited to some contexts than others. It can be used for useful research or can be misused for purposes for which it was never intended, and the reader must have a minimum of knowledge and understanding to interpret the results correctly. With tools that are less impartial in origin, and/or less comprehensively documented, the risk of misunderstanding and misuse is even greater.

References

1. http://en.wikipedia.org/wiki/David_Harley

2. http://www.eset.com/

3. http://www.amtso.org/

4. http://www.hispasec.com/en/

5. http://www.virustotal.com/

6. David Harley: 'VirusTotal is not a Comparative Analysis Tool!'; http://blog.eset.com/?p=150 (2008)

7. http://www.canterbury.ac.uk/social-applied-sciences/computing/conferences/CFET2011/Home.aspx

8. VirusTotal: 'Virustotal.com in depth: how to use it properly'

9. http://virusscan.jotti.org/en

10. http://www.virscan.org

11. Aryeh Goretsky: 'Problematic, Unloved and Argumentative: What is a potentially unwanted application (PUA)?'; http://go.eset.com/us/resources/white-papers/Problematic-Unloved-Argumentative.pdf (2011)

12. Aleksandr Matrosov, Eugene Rodionov, David Harley: 'TDSS part 1: The x64 Dollar Question'; http://resources.infosecinstitute.com/tdss4-part-1/ (2011)

13. http://blog.eset.com/?s=TDSS; http://www.eset.com/resources/white-papers/TDL3-Analysis.pdf

14. David Harley: 'SODDImy and the Trojan Defence'; http://go.eset.com/us/resources/white-papers/SODDImy-and-the-Trojan-Defence.pdf (2010)

15. http://www.eset.com/threat-center

16. David Harley: 'The Game of the Name: Malware Naming, Shape Shifters, and Sympathetic Magic'; http://go.eset.com/us/resources/white-papers/cfet2009naming.pdf (2009)

17. David Harley, Pierre-Marc Bureau: 'A Dose by any other Name'; http://go.eset.com/us/resources/white-papers/Harley-Bureau-VB2008.pdf (2008)

18. Bernard Quintero: 'AV Comparative Analyses, Marketing, and VirusTotal: A Bad Combination'; http://blog.hispasec.com/virustotal/22 (2007)

19. http://en.wikipedia.org/wiki/VirusTotal.com

20. Anti-Malware Testing Standards Organization: Fundamental Principles of Testing; http://www.amtso.org/amtso---download---amtso-fundamental-principles-of-testing.html (2008)

21. Anti-Malware Testing Standards Organization: ‘Whole Product Testing Guidelines’; http://www.amtso.org/amtso---download---whole-product-testing-guidelines.html (2010)

22. David Harley: ‘The Curious Art of Anti-Malware Testing’; http://go.eset.com/us/resources/white-papers/Curious_Act_Of_Anti_Malware_Testing.pdf (2009)

23. David Harley: ‘Execution Context in Anti-Malware Testing’; http://smallbluegreenblog.wordpress.com/2009/05/15/execution-context-in-anti-malware-testing/ (2009)