The Game of the Name

Malware Naming, Shape Shifters and Sympathetic Magic

David Harley BA CISSP FBCS CITP
Director of Malware Intelligence, ESET

ESET LLC, 610 West Ash Street, Suite 1900, San Diego, CA 92101
dharley@eset.com; +1 619 204 6461

CFET 2009
3rd International Conference on
Cybercrime Forensics Education & Training

Abstract

Once upon a time, one infection by specific malware looked much like another infection, to an antivirus scanner if not to the naked eye. Even back then, virus naming wasn't very consistent between vendors, but at least virus encyclopaedias and third-party resources like vgrep made it generally straightforward to map one vendor's name for a virus to another vendor's name for the same malware.

In 2009, though, the threat landscape looks very different. Viruses and other replicative malware, while far from extinct, pose a comparatively manageable problem compared to other threats with the single common characteristic of malicious intent. Proof-of-Concept code with sophisticated self-replicating mechanisms is of less interest to today's malware authors than shape-shifting Trojans that change their appearance frequently to evade detection and are intended to make money for criminals rather than getting adolescent admiration and bragging rights.

Sheer sample glut makes it impossible to categorize and standardize on naming for each and every unique sample out of tens of thousands processed each day.

Detection techniques such as generic signatures, heuristics and sandboxing have also changed the ways in which malware is detected and therefore how it is classified, confounding the old assumptions of a simple one-to-one relationship between a detection label and a malicious program. This presentation will explain how one-to-many, many-to-one, or many-to-many models are at least as likely as the old one-detection-per-variant model, why "Do you detect Win32/UnpleasantVirus.EG?" is such a difficult question to answer, and explain why exact indication is not a pre-requisite for detection and remediation of malware, and actually militates against the most effective use of analysis and development time and resources. But what is the information that the end-user or end-site really needs to know about an incoming threat?

Introduction

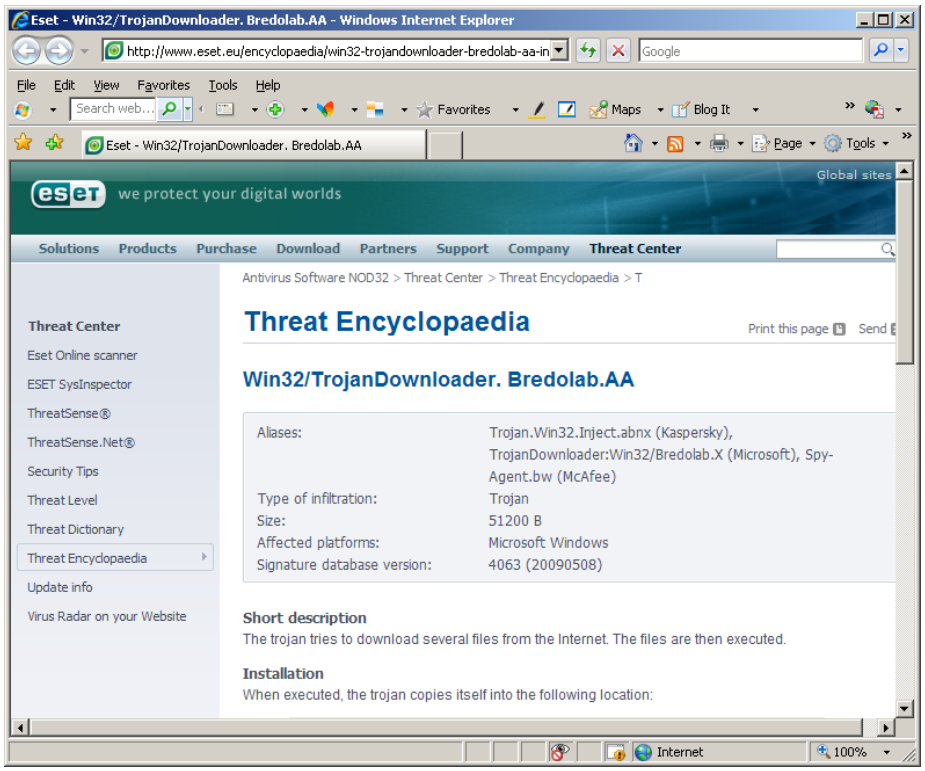
Damon Knight's short story "Babel II" (a science fiction story from 1953: strange how often Sci-Fi crops up in this field!) tells of a world where the protagonist's encounter with an alien he calls the "Hooligan" results in a state of affairs where speech and writing is scrambled so that no human being can understand the speech of any other human being: all written material has also been rendered unintelligible.

Unfortunately, the way in which we (the anti-malware industry) identify malware in terms of naming has become more and more like the North American city of Knight's story.

In the early days of anti-virus, it didn't matter so much. One infection by specific malware looked much like another infection: not to a human observer perhaps (unless you happened to be one of the relatively few people with the knowledge and resources to inspect a disk's boot sector and see that something wasn't right, for instance), but certainly to an antivirus scanner.

It's perfect true that there were complaints even in the early 1990s or earlier about inconsistent virus naming between vendors, but at least virus encyclopaedias and third-party resources like vgrep [1] made it generally straightforward to map one vendor's name for a virus to another vendor's name for the same malware. In fact, vendors still try to maintain a correlation in their descriptions databases between their naming and that used by other vendors (see Figure 1). Furthermore, vgrep (Figure 2), a utility made available under the auspices of Virus Bulletin (the most influential periodical in the anti-malware industry) for online and offline correlation of virus names, is still in existence, though of debate value in today's threatscape . (There are also other tools which have never been publicly available.)

Figure 1: Cross-Reference Between Vendor Detection Names



So what has changed? In the early 1990s, the virus problem was pretty well contained, and Trojans were hardly a problem at all. Most malware spread fairly slowly, and didn't change shape too often. There was a class of malware called polymorphic viruses that struck fear to the hearts of the media, and these did at one point pose a significant problem, in that they drove some of the early, signature-dependent products out of the market place. However, those vendors who were able to adjust their technologies to face these threats may even have benefited by the need for adjustment, since they were obliged to explore more sophisticated scanning algorithms. Conversely, the sophistication of that generation of viruses didn't actually pose the anticipated long-term threat that caused hardened computer journalists to tremble, as they turned out to be more vulnerable to 2nd and 3rd generation algorithms than the less elegant techniques (in some ways) such as server-side polymorphism used by modern malware authors. [2]

Figure 2: Vgrep

Full analysis of heavily armoured Proof of Concept (PoC) malware like Whale (not that *anything* else was quite like Whale) [3] didn't pose too many problems either. They may have been eye-wateringly time-consuming to fully analyse, but it turned out that you didn't actually need, in general, to fully understand every last subroutine present in a virus to write a detection algorithm for it. Note that I'm avoiding the use of the term "signature" here: not only because I share the traditional AV researcher's dislike of the term, but because sophisticated malware requires more sophisticated algorithmic scanning than pattern detection based on exact or near-exact identification (see Table 1).

Table 1: (Near)-Exact Identification

Near-Exact Identification	Recognition of malware where the identification is only good enough to ensure an attempt to remove the virus will not result in damage to the host by using an inappropriate disinfection method. Every section of the non-modifiable parts of the malware body is not uniquely identified.
Exact Identification	Recognition of malware, especially a virus, when every section of the non-modifiable parts of the malware body is uniquely identified.

Modified from "A Dose By Any Other Name" [4]

Even when most malware authors were more interested in PoC than in ROI (Return on Investment), a war of attrition was already taking place between Us and Them. As each side evolved new tricks and technologies, the other would be working on countermeasures and new approaches of its own. Now, however, the pace is much quicker. When I was an AV administrator in the early 90s, it was often still considered enough to update a product once a month or even once a quarter. As Internet connectivity became more commonplace and malware started to spread a little faster, it also became more common to make available daily updates to counter one or more malicious programs, but we were still seeing a few updates to counter a few new viruses and major variants. [5]

Today, the ESET labs expect to see 100,000 unique new samples (*not* variants) a day. Other labs see comparable volumes, though they may count differently, for reasons related to differences in detection design and implementation as well as laboratory procedure. Why are they not variants? Because most of them are not rewritten, recompiled code, but multiple examples of malware families which may or may not share base code: however, where code is shared between sub-variants, the binary is modified using runtime packers, compressors and obfuscators so as to hamper traditional signature detection. Clearly, we're not going to find unique names for each sub-variant, and it's often quite unnecessary (in scanner maintenance terms) even to classify them with the thoroughness that's necessary if we're to assign them to a specific malware family. The following description addresses a threat detection called INF/Autorun which usually makes the top three detections in ESET's ThreatSense.Net® reports [6] and doesn't correspond to a specific malware family in terms of shared code. Such a broadly heuristic detection is impossible to match to detections from other companies, because different companies use different heuristic algorithms.

> This detection label is used to describe a variety of malware using the file autorun.inf as a way of compromising a PC. This file contains information on programs meant to run automatically when removable media (often USB flash drives and similar devices) are accessed by a Windows PC user. ESET security software heuristically identifies malware that installs or modifies autorun.inf files as INF/Autorun unless it is identified as a member of a specific malware family.

Signature-focused anti-malware packages have to release astonishing numbers of updates per day – hence the need for some vendors to implement "in-the-cloud" technology in order to accelerate the speed at which they must generate signatures and make them available. Extensive use of generic signatures (which detect whole families of variant or sub-variant rather than just single variants) and advanced heuristics (which may make use of malware

recognition based on behaviour analysis and other techniques not reliant on resemblances based on shared code) introduce a whole extra layer of complexity. The number of updates released on a daily basis may vary from vendor to vendor to an extraordinary degree. AV-Test's update frequency chart showed, on the 30th July 2009 at 10.28 GMT, the number of updates released over 7 days by over 40 vendors. That number varied between one and 1,916 [7], but don't be fooled into thinking there is a direct correlation between update frequency and detection performance. In general, the products that make the most use of proactive detection techniques for the detection of unknown malware will tend to release infrequent updates, whereas products with many highly specific detections (signatures, if you like) need far more frequent updates.

Within those updates, where detection names are actually made available, one detection name, even one that sounds quite specific (Win32/Nastybot.AXN for example – this is not a *real* example!) may include detection for hundreds or thousands of individual variants and sub-variants. These may or may not include shared base code: they may look very different to a signature-reliant scanner because of the copious use of packers and obfuscators, yet be classified by the same name by a heuristic scanner because they're identified by the same broad-brush heuristic. Conversely, two closely-related malicious programs may be picked up by different heuristic rules and therefore allocated another identifier, depending on a number of factors such as specific characteristics, where the malcode is in its life-cycle (as a lab spends more time processing its variants, it may re-allocate in accordance with later information), and on the infection vector. For example, variants of Conficker might be identified according to their use (or non-use) of the Autorun infection vector, resemblance to other variants, exploitation of MS08-067 and so on, possibly resulting in a different identifier in each case.

Samples may often be identified generically, for example as a Trojan, bot or agent: for example, programs using packing and obfuscation techniques characteristic of a class of malware, and the name. Often, the generic nature of such a detection is indicated by its name: see Table 2 for some examples.

No Strings Attached

For an earlier paper [4] Pierre-Marc Bureau analysed the number of detection strings generated when scanning a set of samples detected as "Autorun" by our products. At that time, NOD32 included the substring "Autorun" in 296 different strings. Testing a number of scanners against the same sample set showed many different names for the samples according to vendor (as expected) and wide variation in the *number* of labels: one vendor used 675 different labels, but another used fewer than 100. These data should not be seen as representing any kind of measurement of effectiveness, and nor should the number of daily updates associated with individual vendors: rather, they should be seen as representing differences in identification factors such as the extent to which each vendor uses signatures and different flavours of heuristic (passive, active and so on).

This evolution in scanning technology has led us a long, long way from the early days of anti-virus, where most detections were static strings unique to a virus or variant. Instead, detection techniques now include more complex algorithms, to heuristic approaches that identify unknown malware as malicious by using:

- Static analysis to determine whether it would behave maliciously if executed
- Dynamic analysis to see whether a program behaves maliciously when executed in an isolated environment (sandboxing, emulation, active heuristics, and so on)

Born To Be WildList

In the present decade, sample numbers have escalated exponentially, analysis and detection has become increasingly generic, and naming, even for some WildList malware, has become nearly useless because of the difficulty of mapping samples to names.

Why should we still care about the WildList? It's true that this resource (http://www.wildlist.org) has become marginalized (not only in the public eye but as regarded by some sectors of the anti-malware industry) by its focus on replicative malware, which now comprises a tiny percentage of the totality of malware that now poses a threat to the end-user, and the fact that it is perpetually behind the curve in terms of threat currency [4]. However, the organization is working on those limitations it still has relevance in detection testing and certification, where it provides a "level playing field" collection of well-validated malware that all mainstream anti-virus software can reasonably be expected to detect, providing a limited but comparatively reliable potential metric.

However, the WildList also provides a dramatic illustration of how irrelevant naming has become. Consider the two tables that follow: Table 3 is extracted from the July 1999 WildList: it includes a small but representative sample of some of the better known examples of the 132 viruses reported in the main WildList that month, and the "list date" and "reported by" are omitted.

Table 3: Some Names from July 1999 WildList

Name of Virus	Alias(es)
AntiCMOS.A	Lenart
Byway.A	Dir2.Byway
Empire.Monkey.A	Monkey
O97M/Tristate.C	O97/Crown.B
W97M/Melissa.A	Maillissa
WM/Wazzu.A	Wazzu
X97M/Laroux.HJ.	Bayantel

Table 4 is extracted from the June 2009 WildList (the July WildList isn't yet available at the time of writing). The "Aliases" column isn't included, because no alias is given for any entry, though the field has been retained in the current WildList report.

Table 4: Some Names from June 2009 WildList

Name of Virus
W32/Agent!ITW#100
W32/Autoit!ITW#100
W32/Autorun!ITW#260
W32/Bagle!ITW#137
W32/Conficker!ITW#1
W32/Ircbot!ITW#474
W32/Koobface!ITW#24
W32/Netsky.Q
W32/Onlinegames!ITW#110

Even in the limited context of WildList (or, more specifically, the samples included in WildCore, the corresponding sample set, specific instances of malware are rarely identifiable by name only. This is not only due to differences in naming conventions. Continuous malware modifications may not qualify as new variants with individual name, so there is no guarantee that any sample that hits an individual's desktop is identical to any single entry in the current WildList, even though it is flagged with the same generic name. Even if it *is*, there's no way, in general, for that individual to know *which* without access to the physical sample set, as distributed to trusted individuals.

Naming of Parts

"Identification of malicious code remains paramount: precision in *naming* is almost irrelevant." [4]

One of the reasons my generation of researchers dislikes the term signature detection is that it is seen as equivalent to identifying the presence of malicious code with some precision (exact identification or near-exact identification). However, most modern anti-malware detects a wide range of unknown malware or variants using less rigid analytical tools variously described as heuristics, behaviour analysis, dynamic analysis and so on. Precision in naming a malicious program does not provide an accurate measurement of how effectively a scanner detects and deals with a malicious object. In fact, all modern anti-virus scanners are essentially heuristic: that is, they don't plod wearily, byte by byte, through each file they scan before they decide it's infected or uninfected, malicious or innocent.

I may have given the impression that there has never been a meaningful naming convention, but in fact a CARO [8] standard has existed since 1991 but has been "mostly to do with what you cannot use as a name [3]." As there is no CARO reference collection, CARO naming is not sample-based and doesn't offer a catalogue of specific sample/identifier matches [9].

The CME initiative (Common Malware Enumeration: see http://cme.mitre.org/) was intentionally "divorced" from the detail of single specific samples. Instead, it aimed to represent each threat by a *collection* of one or more relevant samples "... so that someone with their own threat sample will be able to find the correct CME identifier associated with the sample [10]." However, the initiative foundered "on the rocks of reality" [4], or perhaps became enmeshed in a Sargasso Sea of conflicting expectations, and is no longer updated.

Consider (Table 5) the final CME entry on the web site, CME-711, a collection of samples generally associated with the (now more or less defunct) Storm botnet:

Table 5: CME-711

Aladdin	Win32.Small.dam
Authentium	W32/Downloader.AYDY
AVIRA	TR/Dldr.Small.DBX
CA	Win32/Pecoan
ClamAV	Trojan.Downloader-647
ESET	Win32/Fuclip.A
Fortinet	W32/Small.DAM!tr
F-Secure	Small.DAM
Grisoft	Downloader.Tibs
Kaspersky	Trojan-Downloader.Win32.Small.dam
McAfee	Downloader-BAI!M711
Microsoft	Win32/Nuwar.N@MM!CME-711
Norman	W32/Tibs.gen12
Panda	Trj/Alanchum.NX!CME-711
Sophos	Troj/DwnLdr-FYD
Symantec	Trojan.Peacomm
Trend Micro	TROJ_SMALL.EDW

Table 6 shows a fairly typical VirusTotal report, in this instance for a recent Microsoft zero-day exploit. Virus Total provides a service for estimating the likelihood that a submitted file is malicious by running a battery of scanners against it to see whether any of them identify it as malware. Vendor identifiable and other data have been removed, as it's common to mistake such a list for an indicator of comparative performance, which can be highly misleading, and I prefer not to promote that misconception (http://blog.hispasec.com/virustotal/22).

Table 6: Naming Variations in a VirusTotal Report

a variant of Win32/AutoRun.Agent.OS
Heuristic.LooksLike.Win32.Suspicious.A!92
Mal/Packer
Packed/NSPack
Packed/NSPack
Packer.NSAnti.Gen (v)
PAK_Generic.005
Startpage.EJD
StartPage-HR
StartPage-HR
Suspicious File
TR/Crypt.FKM.Gen
Trj/CI.A
Trojan.Dropper
Trojan.MulDrop.32247
Trojan.Win32.AvKiller.kv
Trojan.Win32.MulDrop.31605
Trojan-Dropper.Agent
Trojan-Dropper.Agent!IK
TrojanDropper.Agent.aiw
Trojan-Dropper.Win32.Mudrop.awn
VirTool:Win32/Obfuscator.EH
W32/Behav-Heuristic-063
W32/Mudrop.AWN!tr
W32/OnlineGames!Generic
W32/OnlineGames!Generic
W32/Packed_Nspack.A
Win32.Packed.Klone.ap03
Win32/Zlob.KC
Win32:Rootkit-gen
Win32:Rootkit-gen
Win-Trojan/Obfuscator.31829

If we break down the information in this list to see what we can deduce from the names, there is *some* information there. Several vendors are detecting it simply from the fact that it uses a packer/obfuscator; some from the fact that it appears to have rootkit functionality, some from its functionality as a Trojan dropper (a program that simply installs other malware). However, this doesn't tell us anything very specific.

CME identifiers, the WildList, and that VirusTotal report simply refer the casual onlooker back to an aggregation of names without a reference sample – not that the average computer user could do anything useful with a reference sample – though VirusTotal does give us hash values that have some limited use even without access to a sample.

But when an identifier can be applied equally appropriately to malware from completely unrelated malware families, in what way is the identification useful? We've already seen that naming is no longer about sample identification, at least in the sense of giving exact

information to the end user. Unfortunately, nearly all vendors continue to provide “Top Ten” threat lists that while internally consistent don’t relate usefully to other vendor lists. This often has the unfortunate result that a customer cannot tell whether Company A can detect a virus according to the name given to it by Company B, and have to ask us whether we detect “the media virus du jour” [4]. No wonder so many people outside the industry assume that we don’t know what we detect.

Table 7 shows the kind of (loose) format often used on industry and other mailing lists (such as AVIEN’s [12] for the exchange of information regarding upcoming threats.

Table 7: Threat Information

Message subjects (for mailborne malicious links or code)
Malicious links
Filenames
MD5 and/or SHA1 signature of suspicious file
Summary of detections reported when the actual sample was submitted to VirusTotal (or a similar resource).

For such information to be useful to the recipient, though, he may need a sample to check against the hash value (for the sake of debate, I’ll ignore the fairly small risk of a hash collion), and certainly needs knowledge of both naming conventions and detection technology that isn’t found in the general computing population.

Heuristic and/or generic detections reported by multiple scanners against distinct individual samples give some idea of the difficulties of establishing useful identification of an infection purely on the basis of a name supplied by a scanner. The following example (Table 8), from a previous paper[4], is taken from reports of newly-emergent (presumed) malware reported at a stage in their life-cycle that for most or all scanners predates sample analysis and, therefore, (near-) exact identification.

Table 8

Win-Trojan/Downloader.62976.M
TR/Crypt.XPACK.Gen
W32/Downldr2.BLMC
Downloader.Agent.AETI
Trojan.Downloader.Exchanger.D
(Suspicious) - DNAScan
Trojan.DownLoader.50204
Suspicious File
Win32/Collet.AA
Trojan-Downloader.Win32.Agent.mik
Win32.SuspectCrc
Trojan:Win32/Tibs.gen!G
Win32/Agent.ETH
Trj/Downloader.SZE
Troj/Exchan-C
Downloader
Trojan.Crypt.XPACK.Gen

Source: A Dose By Any Other Name [4]

Conclusion

Does all this matter to the end user? Only if the user thinks it does.

A conscientious administrator might always want to know exactly what is happening and what, up to a point, is being detected, even if it only tells him *what* was being blocked at the gateway, for instance. Many people feel they can't do their jobs properly without that low-level information. But we're no longer in that relatively unpressured threat landscape where Lovebug could occupy virtually all our attention in terms of outbreaks by appearing as several variants over a few hours. Now we have many thousands of new, unique variations a day (irrespective of whether they're technically variants).

Inevitably, we have to use some form of filtering (whether it's generic, heuristic, behavior analysis or whatever) to detect proactively: in effect, to detect what we can't identify exactly. It doesn't usually matter in principle to the end user, or the home user. Unless, of course, the detection is a False Positive (FP) (or, more to the point, an FP that gets noticed!), or a borderline detection, or a disinfection more damaging than the infection.

We all need to effect some mindset adjustments. The anti-malware industry and major testers and reviewers are used to recalibrating, but expectation management in the user community is a *real* challenge.

Too often the industry plays the game of pretending that we know exactly what we've just detected, because that's what the customer expects. And it often doesn't work because:

- We detect some things with label X and miss other things with the exact same label
- The more curious users and others, like the wider security community, go to our web sites and often find information that's either so generic that they don't know what it means, or that's highly detailed information on something that was analysed in depth earlier in its evolution, so it's all about what it *used* to look like, not what it looks like now.

ESET has been working for a while on our own top ten lists so that they focus on interpreting trends, rather than detailed analysis of a single malicious program. However, like other vendors, we have to strike a balance between giving a realistic picture of the threatscape and the unrealistic expectations of most of our audience. I used this quotation from Ursula Le Guin in a previous paper [4], but I'll use it again because it succinctly summarizes the popular assumption that to name malware is to understand and detect it: "… the name is the thing…and the truename [sic] is the true thing. To speak the name is to control the thing [13]."

As I also said in that paper, malware naming is, in real life, closer to a very different conceptualization, "When a man shows another man a particular part…and he can't recall the proper label for that part…He calls it a doodad or a hingey or a whatchamacallit….A doohingey can be the name of a scrub mop or a toupee. It's a term used freely by everybody in a certain culture. A doohingey isn't just one thing. It's a thousand things [14]."

We can do a lot to mitigate the effects of new variations (variants, minor variants, subvariants), even malcode we haven't met before. We don't have to generate a three-page analysis of each sample, fortunately. But most customers aren't going to understand this better unless we improve communication with them.

In brief, why is consistent naming a problem? If it's *not* a problem, identification by naming is just background noise, and we can filter most of what plagues us without naming every last subvariant. (Of course, for PR purposes we need to reassure our customers occasionally that we really are filtering it, or they might start to think they don't need the product…) When it gets tricky is when we need to remediate what we didn't detect…

Often we're asked: "Do you detect {name of media malware du jour}?" Unfortunately, the question requires a two phase answer. The short answer is (unless we've seriously dropped the ball, or the malware in question is so localized we haven't seen it even though another vendor has) usually "yes in principle..." But the longer answer is along the lines of "Yes, in principle, but we can't confirm that we detect a specific sample unless you can give us that sample, or at least a hash to see if it matches something in our database." Which isn't, unfortunately, the "yes, of course we do" reply that we could often make in the 1990s that customers have been conditioned to expect as a measurement of vendor competence.

Naming in the sense of a catchy name like Lovebug or Storm is mostly important to the media. That's not a criticism: it's hard to hang a good story off a label like "probably a variant of Win32/Statik". But naming is mostly smoke and mirrors. Malicious programs located minutes apart in the bitstream may have nothing in common but a name. It works the other way too. A single sample may be named so differently by different vendors that there's no way to identify it as the same program except by a hash value. You could say that nowadays, the hash *is* the name. At any rate, when precise identification matters.

Naming isn't as helpful as it's assumed to be, at least, not for the end user. Variant naming can't usually be tied to a single sample. The right question isn't "what is it called?" which was never the most important issue and is now almost unanswerable. The right question is "what do I do about this?"

As long as the anti-malware industry lets the rest of the world dominate the question and answer process without trying to explain what the issues really are, we make life more difficult for our customers as well as for ourselves.

The glut problem (AV-Test now claims to have a sample database exceeding 22 million samples! [15]) can't be fixed by near-exact identification. Proactive and generic detection technologies are not shortcuts for lazy programmers, but essential components of an anti-malware toolkit. It's not the name that matters, but the detection.

References

[1] Vgrep: Virus Bulletin http:// http://www.virusbtn.com/resources/vgrep/index

[2] David Harley, "Viruses and Worms" in "Maximum Security 4th Edition" ed. Anonymous, SAMS 2004.

[3] Alan. Solomon & Dmitry Gryaznov, Dr. Solomon's Virus Encyclopaedia, S&S International, 1996

[4] David Harley & Pierre-Marc Bureau, "A Dose By Any Other Name", in "Virus Bulletin Conference Proceedings", Virus Bulletin 2008

[5] David Harley, Robert Slade & Urs Gattiker, "Viruses Revealed", Osborne, 2001

[6] http://www.eset.com/threat-center/threat_trends/Global_Threat_Trends_June_2009.pdf

[7] Update Frequency of Anti-Virus Software, AV-Test, http://www.av-test.org/index.php?menue=7&lang=0&sort=down&order=updates

[8] "Current Status of the CARO Malware Naming Scheme", Dr Vesselin Bontchev, at http://www.people.frisk-software.com/~bontchev/papers/naming.html

[9] "How Scientific Naming Works", Joe Wells, at http://www.wildlist.org/naming.htm

[10] "The CME Process: Scope, Identifiers, and Guidelines for Deconfliction", Desiree Beck, at http://cme.mitre.org/cme/process.html

[11] http://www.virustotal.com/analisis/743cfae926c2f3ee518c36243985ece9a36911b91800a7bdc416929c6ba5f439-1246631982

[12] Anti-Virus Information Exchange Network: http://www.avien.net

[13] "The Rule of Names", Ursula K. Le Guin, Ziff-Davis 1964

[14] "Doodad", Ray Bradbury, in "Astounding", Abner Stein Ltd. 1943

[15] http://www.sophos.com/blogs/gc/g/2009/07/24/avtestorgs-malware-count-exceeds-22-million/