A DOSE BY ANY OTHER NAME

Pierre-Marc Bureau, David Harley
ESET Research, 610 West Ash Street, Suite 1900, San Diego, CA 92101, USA

Tel +1 619 876 5458, +1 619 876 5467
Email {pbureau, dharley}@eset.com

ABSTRACT

Years ago, when alt.comp.virus was still useful, 'Name that virus' was a popular virtual party game, and virus names were, if not standardized, at least easy to cross-reference with tools like VGrep [1]. In 2008, the numbers have escalated exponentially, analysis and detection have become increasingly generic, and naming, even for some WildList malware, has become nearly useless because of the difficulty of mapping samples to names. The CME (Common Malware Enumeration) initiative [2], while attempting to achieve something many people wanted, seems to have foundered on the rocks of the reality. Yet we continue to provide 'top ten' threat lists that have virtually no commonality or consistency across different vendors and sites, so that our customers continue to ask whether we detect the media virus du jour, and the slashdotty community point to us and giggle at our incompetence in failing to provide information about what we detect. Are all our solutions going generic? Are there ways to resolve this issue so that our customers can understand what's happening and regain some faith in the industry without being hung up on the question 'Do you detect virus X?' We think so, and will discuss some possible approaches in this paper.

INTRODUCTION

'*What's in a name? That which we call a rose*
By any other name would smell as sweet.' [3]

For some years in the 1990s, the newsgroup alt.comp.virus provided a (moderately useful but totally unmoderated) meeting ground between the anti-virus community (in a broad sense), the VX (Virus eXchange) crowd of virus writers and wannabes, and people looking for information about viruses or AV products. Before the signal-to-noise ratio there got so bad that most researchers gave up on the group, it was not uncommon for someone to initiate a game of 'Name that Virus' where one or more characteristics of a specific virus were described, and others would attempt to identify it from the description. At the time, while inconsistencies in virus naming between products were already a source of some potential confusion, a naming convention did already exist following a 1991 meeting by CARO members [4]. However, the 'naming standard…[was] mostly to do with what you cannot use as a name' [5]. As Joe Wells subsequently pointed out [6], because there is no CARO reference collection, CARO naming is not sample based and constitutes a guideline on format rather than a catalogue of specific sample/identifier matches. The WildList, on the other hand, is very much sample based, being tasked primarily to 'report exactly which viruses are spreading in the wild, to collect samples of those viruses, and to provide the viruses to bona fide antivirus researchers…' In consequence, its approach to naming is pragmatic, and does not claim to name viruses 'correctly' or 'authoritatively'. At the moment, however, WildCore remains at least nominally bound to replicative malware rather than the wider range of non-viral malware that tends to dominate today's threatscape. CARO and its members remain important players in defining naming standards [4] and have extended the scope of the scheme far beyond replicative malware, but acknowledge that no product is fully compliant with it.

CME was intentionally 'divorced' from the detail of single specific samples, representing each malware threat by 'a collection of one or more "samples" … [which] will likely contain multiple files (i.e. not consist of a single executable binary file)... so that someone with their own threat sample will be able to find the correct CME identifier associated with the sample [7].' However, it seems to have lost all impetus. Furthermore, that intention to relate to threat samples is compromised by the rapidity with which many malware families release new variants and/or use new packer variants, changing many characteristics on a (more than) daily basis.

BOTS ON THE LANDSCAPE

The attempts of 'notoriety-hungry nerds of yesteryear' [8] to produce sophisticated Proof of Concept (PoC) viruses have given way to profit-oriented initiatives directed by 'anonymous career criminals'. Self-replicating malware that spreads far and fast has decreased dramatically in market share compared to (for instance) shortish spam runs pointing to URLs poisoned with trojans. These threats are reinforced with server-side polymorphism, repeated packing and obfuscation, recompilation, self-updating and so on, designed to increase their resistance to signature-focused detection. The sophisticated (but, arguably, more easily detected in the long term) polymorphic viruses that characterized the early 1990s have become of largely historic interest. And naming has become increasingly divorced from sample identification.

The WildList Organization is in the process of working on its limitations, which are well known: the WildList is based on a well-validated but small and purely viral sample set which is perpetually behind the curve in terms of threat currency. Even under those circumstances, differences in naming conventions and continuous malware modifications that may not qualify as new variants with individual names mean that specific instances of malware cannot be identified by name only: effectively, the usefulness of the WildList is largely restricted to the physical sample set, as distributed to trusted individuals. To the individual without such access, names like W32/Agent!ITW#33 or W32/Autorun!ITW#174 mean very little. This is in stark contrast to the previous decade, where most of the people likely to read the WildList *at all* had an idea of what Stoned.Michelangelo.A was, and could be reasonably confident that it mapped to a verifiable infection and detection.

Some mailing lists exchange information about suspected new threats or threat variants including such details as:

- Message subjects (for email-borne malicious links or code)
- Malicious links

- Filenames
- MD5 and/or SHA1 signatures of suspicious files
- Summary of detections reported when the actual sample was submitted to *VirusTotal* (or a similar resource).

Useful though such information can be (such reports are considered a considerable asset by many members of AVIEN [9], where advisories on this model have been in use for many years), it also offers a significant demonstration of the naming problem. An informal survey of generic and/or heuristic detections reported in this way illustrates how difficult it is to identify the exact nature of a specific threat for anyone who doesn't have:

- An actual sample to check against an MD5 (for instance).
- Knowledge of the naming conventions used by individual vendors for heuristic, generic, and malware-specific detections.
- Knowledge of what the terms heuristic, generic and malware-specific actually imply – see glossary for a simplistic set of definitions.

These requirements actually imply technical knowledge and access to resources not commonly found beyond the borders of specialist communities (and presuppose that the risk of additional problems – hash collisions, for example – is fairly small).

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 examples are 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.

Sample 1

- TR/Crypt.XPACK.Gen
- Trojan.Crypt.AP
- (Suspicious) - DNAScan
- Trojan.Crypted-16
- Suspicious File
- VirTool.Win32.LDE
- A variant of Win32/Nuwar.CG
- Troj/Dorf-BA
- Trojan.Peacomm
- Trojan.Crypt.XPACK.Gen

Sample 2

- TR/Crypt.XPACK.Gen
- Win32:Zlob-BVC
- (Suspicious) - DNAScan
- Trojan-Downloader.Win32.Exchanger.f
- Trojan.Crypt.XPACK
- Trojan-Downloader.Win32.Exchanger.f
- Trojan:Win32/Tibs.gen!G
- Probably a variant of Win32/Statik
- Troj/Agent-GVE
- Suspected of Downloader.Zlob.8

Sample 3

- 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

Clearly, these are snapshots of identifications made early in the detection process: any of these names may have changed dramatically once the samples were analysed within the relevant virus laboratories, allowing more precise identification and assignment to an appropriate malware family. What constitutes 'appropriate' assignment, however, can vary widely from one laboratory to another.

Similarly the 'latest' CME entry on the website at time of writing seems to be CME-711 [2], a collection of samples generally associated with the Storm botnet:

- Win32.Small.dam
- W32/Downloader.AYDY
- TR/Dldr.Small.DBX
- Win32/Pecoan
- Trojan.Downloader-647
- Win32/Fuclip.A
- W32/Small.DAM!tr
- Small.DAM
- Downloader.Tibs
- Trojan-Downloader.Win32.Small.dam
- Downloader-BAI!M711
- Win32/Nuwar.N@MM!CME-711
- W32/Tibs.gen12
- Trj/Alanchum.NX!CME-711
- Troj/DwnLdr-FYD
- Trojan.Peacomm
- TROJ_SMALL.EDW

The CME initiative might still have some use within the original framework of objectives as a means of describing classes of malware and cross-referencing more-or-less generic names, but that use would be compromised in many respects by the inability to access and cross-refer to individual samples in a reference collection. For the everyday user, a CME identifier, like one of those strings of *VirusTotal* detections, simply refers them back to an aggregation of names without a reference sample. (Leaving aside the question of what an everyday user could usefully do with a reference sample!) When identifiers are included that could be applied equally appropriately to malware from a completely unrelated malware family, what has Joe Average really learned from the identification?

These examples suggest some interesting aspects of present common practice in heuristic and generic identification:

- The presence of certain packing and obfuscation tools as a major heuristic results in the use of a single identifier across a wide range of individual samples.
- Some identical, highly generic identifiers are used in every case by some vendors, whereas other identifiers suggest three different malware families.
- An entire range of heuristic techniques may be masked by a single 'suspicious' or 'probable' identifier.
- Use of the name of a generic class of malware as an identifier.

These are entirely legitimate practices, but they don't actually give the scanner user any help in identifying *exactly* what has been found on (and possibly infected) their system. Even if it *were* possible to give such help on a newly discovered sample, how useful would it be to the user? Is it any more reasonable to expect precise identification of a malicious program from a heuristic detection than it is to expect such information from an email filter that blocks all .EXEs? We believe not, with one major reservation.

All the above detections were harvested in laboratory or pseudo-laboratory situations: that is, the samples were examined in the cybernetic equivalent of a Petri dish. They were not allowed to execute except within the constraints of a virtual environment, and even then, only where the scanner used some variation on dynamic analysis to examine the suspicious object. In a 'live' environment where the suspicious object may have already had the opportunity to infect a system, the question arises as to whether the scanner is capable of effective disinfection on the basis of a generic or heuristic detection. Where a detection label is *obviously* generic, the end-user may be able to evaluate the likelihood of an effective disinfection, but a highly generic identifier can be mistaken for a near-exact or exact identification because it can't be distinguished from a label format used for more-or-less exact identification based on a complete analysis. Thus, it may be incorrectly assumed that effective post-infective removal is a given. In real life, a generic detection cannot always guarantee a safe generic disinfection.

TESTING DIFFICULTIES

These difficulties in mapping identifier to sample have very serious implications for detection testing, among other issues. Poor scanner detection testing is often based on so-called 'validation' of samples by using a 'favoured' scanner to identify each sample. Of course, there are many reasons why this approach is methodologically invalid (scanner bias, inability to distinguish false positives, and so on). In addition, though, where the tester fails to recognize a generic or heuristic detection as such, the likelihood increases dramatically that a sample giving rise to a possible false positive by that product will be included incorrectly in the test.

In a not-unrelated issue, researchers have often expressed concern [10] that heuristic detections may cascade inappropriately throughout the industry as signature detections, because some vendors have failed to check that the program is actually malicious (or potentially unwanted, or whatever) let alone whether the detection is heuristic. As Bustamente [11] points out, it's not intrinsically incorrect to flag a program heuristically as 'suspicious' that shares significant characteristics with known types of malicious software, but serious problems arise when false detections are disseminated throughout the anti-malware (and anti-malware testing) industries. We would also contend that when a heuristic detection identifier is cascaded as a specific (signature) identifier, its value as an identifier is to some extent compromised, not only for the user and for a vendor using it as signature detection, but for the originating lab. At best, it increases the likelihood of confusion among end-users.

RECEIVED WISDOM

While the terms 'exact identification' and 'near-exact identification' aren't commonly used outside the industry, there is an implicit assumption in common usage that signature detection, if we must use that phrase, is somehow equivalent to identifying the presence of malicious code with some precision. However, modern anti-malware is usually capable of detecting a wide range of unknown malware or variants using a wide range of analytical tools variously described as heuristics, behaviour analysis and so on. The popular insistence that precision in naming a malicious program is a primary measurement of how effectively a scanner detects and processes a malicious object is mistaken. In fact, it's not really compatible with the way in which modern scanners work. We'll address the <irony>minor</irony> question of how we change this perception in due course. But for now, we'll move on to a simple and obvious proposition. All anti-virus scanners are heuristic.

Some, of course, use more complex heuristic analyses than others. But no reputable commercial scanner trudges wearily through a database of signatures, string by string, for each and every object it scans. (Near-)exact identification is resource-intensive, even where it's appropriate to the threat type, and is impractical as the main scanning tool in today's climate of malware glut, where no vendor has timely access to or time to process *every* malicious variant in existence. But even scanners that don't utilize particularly advanced heuristics don't, for instance, usually look for macro viruses in the MBR, or for static strings in impossible locations.

LOCATION, LOCATION, LOCATION [12]

A scanner might look for a signature of some sort in a place where it might be found in an unviable but infected executable, for instance, but that falls into the question of what a scanner might detect apart from infected, viral or otherwise malicious objects – garbage files, harmless test files, corruptions, intendeds, and so on. That *does* have a bearing on our discussion, because it introduces a particular naming complication. 'Does your product protect me from W32.FailedTrojan.DAM or PEVirus.NotAVirus?' (Suggested answer: 'No, because even if we knew what Company X meant by that name, the identifier suggests that it isn't damaging and protection isn't needed.') However, it's not our main concern.

Essentially, the amount of processing time a scanner expends on identifying and removing a likely or proven threat is regulated by the context in which it scans: to take a simplistic example, a gateway scanner doesn't have to waste time on disinfection if it can simply block. A product designed to run on platforms with limited exposure and functionality is unlikely to scan (malware-specifically, generically, or heuristically) for malware that isn't normally effective on that platform. (Mixed environments where use is made of multiple desktop platforms are a different issue, and we won't discuss them here.)

DO YOU DETECT VIRUS X?

We are often asked whether our scanner detects such-and-such a virus, and under what name. Sometimes we can be exact (especially with older viruses, but it's rare to be asked about obsolete malware). Often, 'such-and-such a virus' is actually a generic detection, and that poses problems of expectation management, because customers continue to believe that exact identification is a necessary demonstration of effectiveness at resolving a malware problem. Although, like most vendors, we respond to and therefore, arguably, perpetuate such assumptions by publishing 'top ten' information, we have gone to some trouble to encourage differentiation between malware-specific and generic detections, focusing on useful trend information rather than raw statistics.

Sometimes we can quote a somewhat equivalent detection: for example, many vendors have generic signatures for Storm-related malware under different names (Zhelatin, Nuwar, Peacomm etc. [2]). However, the correlation may be illusory: it's often unrealistic to compare generic detections because they catch whole families of known malware (and sometimes unknown family members), and each vendor uses slightly different (sometimes *very* different) criteria to define and identify a family, let alone individual family members and variants. Under these circumstances it's not at all easy:

- to compare detection rates of specific family members, let alone variants and sub-variants
- to establish whether a (presumed specific) malicious program is detected by a generic driver or heuristic without reference to a specific sample.

Looking back over our own product's detections over a period of over a year, we find that 50% or more of our 'top ten' detections are, characteristically, generic and/or heuristic, and that in some areas (email-borne replicative malware, for example) generic and heuristic detections far exceed malware-specific detections [13]. Of course, this kind of statistic will vary widely across products, not only according to the extent to which individual vendors make use of generic technologies, but also according to the effectiveness of those technologies against the current totality of active threats. This effectiveness may vary dramatically over lengthy periods. As has been pointed out many times, nearly all detection is, to some extent, heuristic, so percentages like this are, to some extent, 'magic numbers'.

An informal survey of the 'top ten' detections by several of the 'big names' in anti-virus demonstrates that a very high proportion (in some cases all) top ten entries are, nowadays, generic. Even if such detections were not characterized by the .gen suffix which is conventionally used to denote generic signatures, figures characteristically given on vendor websites are aggregated under a generic identifier: for example, W32/Mytob rather than a specific Mytob variant identified by a suffix consisting of a string such as .A, .FG, or .CAC. Even where a highly specific variant is included in a list, this will generally not take into account such variations as packed/repacked malware, where compression and obfuscation are used to increase resistance to detection by near-exact identification and generic signature. To take a vendor-neutral example, in the *Virus Bulletin* Prevalence Table for March 2008 [14], the top ten detections consisted entirely of generic detection names such as Pushdo, Netsky, OnlineGames and Agent.

At *ESET*, we currently see consistently high scores from a heuristic applied to a very wide range of malware with the shared characteristic of using autorun.inf for malicious purposes. (This facility allows a program on removable media to run more-or-less automatically when mounted: this is convenient for legitimate installation programs, for instance, but has obvious advantages for malcode.) When a program displays characteristics that suggest malicious intent *and* uses the Autorun facility, it is likely to be flagged by the INF/Autorun identifier.

However, a specific instance of malcode may be detected with a completely different heuristic rule and therefore another identifier, depending on other characteristics, on where it is in its life cycle, and on infection vector. Some samples are identified generically as a bot or agent: for example, programs using packing and obfuscation techniques characteristic of a class of malware.

Figure 1 shows the number of detection strings generated when scanning a set of samples detected as 'Autorun' by *ESET Antivirus*. *ESET Antivirus* has 296 different strings that include the substring 'Autorun'. Running a number of scanners against the same sample set, we get different names for the samples, as we'd expect, but also a huge diversity in the number of labels used by each vendor: in one case, 675 different labels, and in another, less than 100. (These data do not represent any kind of measurement of effectiveness: they simply represent the extent to which identification processes vary between labs.)

NAMING VERSUS IDENTIFICATION

If your product of choice detects malware, does it matter what identifier it uses? The sheer volume of malware variants nowadays means that it's more efficient to use more generic detection techniques such as static and dynamic analysis and

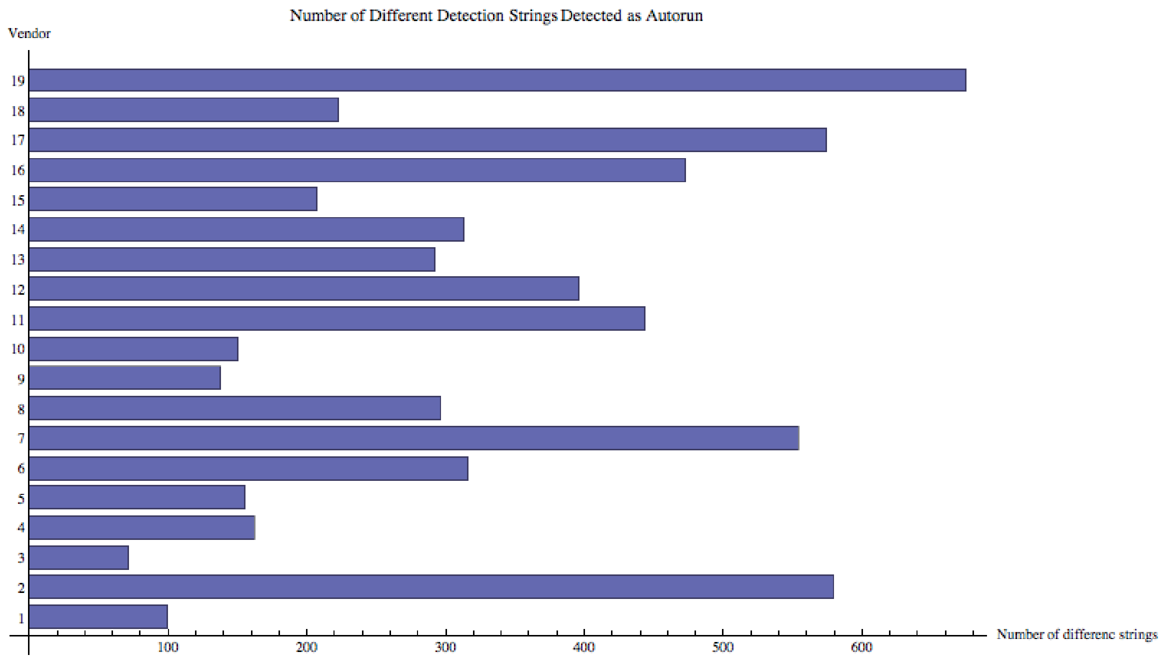

Figure 1: Data extracted from the very comprehensive xref collection released by AV-Test [15] in April 2008. Vendor names are replaced by numeric identifiers to avoid inappropriate interpretations.

generic signatures wherever possible. This has been represented as a detection failure [16], but it's unlikely that botnets and other threats would have a significantly smaller impact if all companies used the same identifier. It could be seen (and no doubt is) as an illustration of the problem this industry has with naming. The time and resources needed to cross-match all the samples seen, so that each vendor can use the same name for each variant or sub-variant, is simply not available in a time of glut, when it's sometimes estimated that the number of new samples for analysis runs into several thousand per hour [16]. In real life, the actual name used by any product might vary widely according to which variant it might have picked up, not to mention when and where the detection was triggered.

Generic signatures offer a considerable advantage by identifying, for instance, software that appears to attempt to exploit a software vulnerability such as the CVE-2007-0038 animated cursor issue [17], lessening the need for further analysis. On the other hand, if detection names are disseminated across other databases (including those of other vendors) without a clear understanding of what characteristics are flagged using that name, confusion ensues. This is *especially* so where the classification is so broad, that it is only possible to map name to sample using a proprietary classification scheme that isn't exported with the name. But anti-malware companies continue to pretend that generic or heuristic detections refer to a single specific malicious program, when it's exactly that proactive detection that keeps the anti-malware industry from wasting resources on unnecessarily exact identification.

It doesn't matter how exact a detection is, unless it generates problems like a false positive, or a disinfection more damaging than the malware itself. (Unfortunately, malware-specific detection is often far easier than generic disinfection, especially for non-viral malware. Thus it is particularly important that malicious programs are detected as early as possible in their evolution, ideally [but not always feasibly] as soon as they hit a proactive scanner's radar.)

The user community's ambivalence around the issues of near-exact identification versus generic detection derives from the blind belief that exact ID should somehow deliver the proactive benefits of generic detection without its uncertainties and risks of false positives. This in turn derives from the belief that security software should offer the 100% security that will absolve the customer from the need to take any responsibility for their own safety [18]. As long as anti-malware marketeers remain reluctant to get away from the TOAST (see glossary) mentality and pander to the perceived 'need' for spurious and unprovable statistics, end-users will continue to expect 100% detection irrespective of their own actions and fume when they don't get it.

The answer is, yet again, education and expectation management: not in terms of turning the entire population into malware experts (we are not that naïve…) but in terms of persuading them to adjust their expectations. In the first instance, we can do this by overhauling our naming and marketing practices. In particular, by making clearer distinctions between generic/heuristic identifiers and more-or-less specific identifiers, and by providing better information on how scanning technologies actually work – not by revealing protected trade secrets, but with clear and easily accessible information on naming practice. Also, by making it clear that:

- There are no 100% solutions.
- They cannot rely on automated detection to free them from the need to implement sound, across-the-board security practice.

- More generic approaches improve detection rather than hamper it.
- The website-hosted malware descriptions our customers flock to are a limited approximation to reality, not precise, authoritative identifications of every known malicious program and, in the absence of a huge real-time reference collection, they cannot be. How and even whether such a collection can ever be realistically maintained is not an issue we can address here.

Conveying these messages to customers, the media, and even the broader security community is no easy task, and there's no guarantee that customers won't reach the conclusion that 'malware detection is broken, but (insert your own panacea du jour here) will give us 100% protection.' However, if we don't acknowledge these difficulties and capitalize on our real strengths, the industry will contract dramatically, and our customers will have as much to lose as we do.

Here's a quotation from science fiction/fantasy that curiously echoes the popular overestimation of the value of standard identifiers, based on the assumption that they always reflect a high degree of precision: '… the name is the thing…and the truename [*sic*] is the true thing. To speak the name is to control the thing' [19]. In real life, malware naming is closer to a very different conceptualization, also from (an earlier era of) science fiction.

> '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.' [20]

However, in the context of malware nomenclature, the driver is not a failure of memory, but a failure of prescience. The industry is surprisingly successful at partially overcoming sample glut using various (and overlapping) forms of proactive detection based on automated analysis of characteristics and behaviour (passive and active heuristics, emulation and so on). But it cannot always ascribe meaningful and precise classification to malicious objects purely on the basis of automated analysis.

To return to Juliet's question [3], a name is an abstraction we use to describe an object, especially its characteristics and its history (what it was before, and so on).

In the malware context, it is possible for two malicious programs to have the exact same (defining) characteristics at one point in time but they might still be named differently because they have completely different histories and might change in opposite directions with time.

Are there alternatives to malware naming that might give more information to users and administrators, while steering them away from inappropriate assumptions? To help users to understand the threats that may be detected on their system, security solutions can provide information detailing the reasons that make an object suspicious. For example, a report might allude to the changes to the system that were attempted by a program, or the infection vector used to install a potentially malicious file on the system. It is possible to have such information in a form that is generic enough not to explain to the bad guys how to bypass the product, but specific enough to inform the user. In fact, this is already happening to an extent (as with our INF/AUTORUN example), but is only useful where the user is in a position to understand what a heuristic or generic identifier really denotes.

In the same vein, the ever-popular 'top ten' reports might be replaced or augmented with reports on the underlying trends in malware. Users might learn more from a report telling them that online game password stealers are on the rise than from a report that a specific family (let alone variant) was responsible for 5% of total detections over a period. Unfortunately, this doesn't answer the question 'Am I protected from…' where often the only accurate answer is 'We can't say for sure without reference to a specific (set of) sample(s).' Translating that into a form of words which is both accurate and acceptable to marketing departments is left as an exercise for the reader. But the user community would benefit from the realization that it's the wrong question to ask.

So what is the 'right' question? That depends on context, but might include all sorts of issues around performance testing and product evaluation that we need not cover again here. In terms of specific threats, though, information useful to a reasonably informed individual might include:

1) What is it? What are its characteristics and how has it evolved? (Clearly, that's a best case scenario: there will be many instances where we don't have the luxury of that precision.)
2) Why is it malicious? In case of generic detection, what makes it potentially harmful?
3) Why was it flagged as (potentially or actually) harmful? What else do I need to know about its effects? Where can I learn more?
4) What is, realistically, the likelihood that my systems will be exposed to it?
5) Is there action I can or should take apart from keeping my anti-virus product updated?
6) Can I expect complete disinfection from the scanner, or will it involve some manual disinfection? If so, how do I do it, and why can't the product do it automatically?
7) What should I tell my users/customers?
8) What alternative performance metrics are available to me?
9) Last but not least, how specific is the information available to me? Am I looking at a broad class of malware, a specific family, or a specific variant?

CONCLUSION

It is important to try to keep customer expectations realistic. The glut problem can't be fixed by throwing more and more resources at analysis throughput focused on near-exact identification. Proactive detection (behaviour analysis) is a Good Thing (though not the 100% solution), *not* a shortcut for lazy programmers, and we need to get this message over to people who are hung up on the idea of 100% detection, perfect signature detection and infallible heuristics, and give precise information about how much detail is available about a given threat.

Modern malware is not always susceptible to automated removal: some families are notorious for digging themselves into a system without any regard for the effect of a botched removal. Precise information about a short-lived variant is a lower priority than detection and blocking of malware families, and precise identification is a poor performance metric without a firm correlation between names and samples [21].

ACKNOWLEDGEMENTS

We don't have space to thank everyone who contributed directly or indirectly to our research, but wish to acknowledge the following people in particular, for their willingness to discuss and share information pertinent to this paper: members of AVIEN, including Conny Jäverdal and Mark Ackermans; Julio Canto of *Hispasec*; Andreas Marx and his colleagues at *AV-Test*; our colleagues at *ESET LLC*, especially Andrew Lee.

REFERENCES

[1] http:// http://www.virusbtn.com/resources/vgrep/; http://vgrep.viruspool.net/.

[2] http://cme.mitre.org/.

[3] Shakespeare, W. Romeo and Juliet: Act II Scene ii.

[4] Bontchev, V. Current status of the CARO malware naming scheme. http://www.people.frisk-software.com/~bontchev/papers/naming.html.

[5] Solomon, A.; Gryaznov, D.O. Dr. Solomon's Virus Encyclopaedia (2nd edition). Dr Solomon's, 1995.

[6] Wells, J. How scientific naming works. http://www.wildlist.org/naming.htm.

[7] Beck, D. The CME process: scope, identifiers, and guidelines for deconfliction. http://cme.mitre.org/cme/process.html.

[8] Harley, D. Macs and Malware. Computer Weekly (in press).

[9] http://www.avien.net/.

[10] Lee, A. Personal communication.

[11] Bustamente, P. Fenomen(al) false positives. http://research.pandasecurity.com/archive/Fenomen_2800_al_2900_-False-Positives.aspx. 2008.

[12] http://www.channel4.com/4homes/ontv/location/index.html.

[13] Global Threat Report 2007, ESET Research team. http://www.eset.com/threat-center/case_study/GlobalThreatReport(Jan2008).pdf.

[14] http://www.virusbtn.com/resources/malwareDirectory/prevalence/index.xml?200803.

[15] http://www.av-test.org/.

[16] Krebs, B. Anti-virus firms scrambling to keep up. http://www.washingtonpost.com/wp-dyn/content/article/2008/03/19/AR2008031901439.html?hpid=sec-tech.

[17] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0038.

[18] Harley, D. I'm OK, you're not OK. Virus Bulletin. November 2006. http://www.virusbtn.com/virusbulletin/archive/2006/11/vb200611.

[19] Le Guin, U. K. The rule of names. Ziff-Davis 1964.

[20] 'Doodad', Ray Bradbury, in 'Astounding', Abner Stein Ltd. 1943.

[21] Harley, D.; Lee, A. Who will test the testers? Proceedings of the 18th Virus Bulletin International Conference 2008.

GLOSSARY

Active heuristic	Analysis of the code present in an object by executing it in some form of virtual environment (roughly equivalent to the forensic term 'dynamic analysis': this is more time-consuming than passive analysis, but can offer more and better information in some circumstances).
Almost exact identification	Recognition of a virus 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. Not every section of the non-modifiable parts of the virus body is uniquely identified.
Exact identification	Recognition of a virus when every section of the non-modifiable parts of the virus body is uniquely identified.
False positive	Describes the scenario where an anti-malware scanner incorrectly detects malware where there is none. Antonymous to 'false negative' where a scanner fails to detect malware.
Generic	Describes security programs that don't recognize specific threats, but defend using a method that blocks a whole class (or classes) of threats. A generic signature is a special case of this; a whole set of known and sometimes unknown variants are detected and processed by a single signature rather than by individual signatures for each variant. Antonym of 'malware-specific.'
Heuristic detection/ scanning	Recognition of an object that has enough viral or malicious characteristics to suggest that it is probably a virus or other malware. Normally assigns a score to each characteristic: a score above a pre-defined threshold triggers a detection identifier.
Known virus scanning, virus-specific scanning	Scanning for known viruses resulting in the identification by name of a virus found in the scanned environment. Malware-specific scanning (known malware scanning) extends the concept to non-replicative malware as well as viruses, worms and so on.

Location, Location, Location	UK property-search television programme: see [12].
Negative heuristic	A rule or criterion, which, if met, lessens the likelihood that the object being analysed is not viral or malicious.
Passive heuristic	Analysis of the code present in an object by passive scanning: the code is examined, but not executed in some form of virtual environment.
Positive heuristic	A rule or criterion, which, if met, increases the likelihood that the program being analysed is viral or malicious.
Scan string, search string	A sequence of bytes found in a known virus that shouldn't be found in a legitimate program. The term is not restricted to static search strings, and may include wildcards and regular expressions, or the use of another virus-specific detection algorithm. Also sometimes known as 'scan signature' or just 'signature'.
Signature	Synonym for 'scan string'. May accurately be applied to a static search string, but often misleads people into thinking there is a single byte sequence used by all virus scanners to recognize each virus or variant.
Sub-variant	A loose descriptor for a variation on a known and defined variant where one detection addresses, or tries to address, a wide range of discrete sample types.
TOAST	'The Only Anti-virus Software That…' (tip of the hat to Padgett Peterson): used here as shorthand for the marketing approach 'Trust us and install our software and we will protect you from everything…'
Variant	Member of a malware family that has its own individual identifier. However, there is little standardization of practice in defining and distinguishing variants across vendor boundaries. Furthermore, the use of obfuscators, runtime packers and so on to hide a known variant from signature scanning is not usually considered to result in a new variant, so a single variant detection may actually address a broad range of individual samples, not all of which may be detected at any one time.
Virus-specific detection	Detection of known viruses using search strings specific to those viruses or variants.
VX	Acronym probably coined by Sarah Gordon for 'Virus eXchange': i.e. relating to authors, distributors and collectors of viruses: the antonym of AV (Anti-Virus). Capitalization is optional and sometimes contentious.
WildCore	Sample collection compiled and maintained by the WildList Organization International, and widely used as a validated test set for In-the-Wild (ItW) testing.