CALL OF THE WILDLIST: LAST ORDERS FOR WILDCORE-BASED TESTING?

David Harley
ESET LLC, 610 West Ash Street, Suite 1900, San Diego, CA 92101, USA

Email David.Harley@eset.com

Andrew Lee
K7 Computing Private Ltd, 6th Floor, Rayala Techno Park, 144/7 Old Mahabalipuram Road, Kottivakkam, Chennai, India

Email alee@k7computing.com

ABSTRACT

The well-documented problems with WildList testing derive from difficulties in adjusting to the 21st Century threat landscape. The (obviously overstretched) WildList Organization's [1] focus on self-replicating malware, which nowadays comprises just a small percentage of the whole range of malware types; the lengthy testing and validation process between the appearance and the inclusion of a specific malicious program on the list, and the availability of the underpinning test set to WildList participants are all cited as objections to the validity of WildList testing, and some vendors and testing organizations have heavily criticized it – some vendors even withdrawing from tests that rely heavily on it.

In line with AMTSO's preference for dynamic over static testing, most mainstream testers have supplemented or replaced WildList testing with some form of dynamic methodology, which, done correctly, is assumed to be a better reflection of today's user experience. So does WildList testing still have a place in testing and certification? Is it still a meaningful differentiator? If it isn't, does that mean that sample validation is no longer considered a practical objective for testers, or is that a misreading of the AMTSO guidelines on dynamic testing?

This paper summarizes the static/dynamic debate, examining the contemporary relevance of the WildList and WildCore.

INTRODUCTION

So, to adapt Sarah Gordon's question [2], *now* what is wild? The WildList Organization's own website uses a definition that hasn't changed much over all the years of the Organization's existence. It still requires a virus to be reported by two or more reporters, and it still quotes a definition written by Paul Ducklin [3]:

> 'For a virus to be considered In the Wild, it must be spreading as a result of normal day-to-day operations on and between the computers of unsuspecting users.'

The note goes on to explain that viruses which 'merely exist but are not spreading are not considered "In the Wild"'. Interestingly, in the current climate where self-replicating malware has largely given way to trojans which have to use external resources for dissemination, the definition has been broadened to include non-replicating malware: '...for a trojan to be considered "In the Wild", it must be found on the computers of unsuspecting users, in the course of normal day-to-day operations.' In fact, though the WildList is seen as 'only viruses' it has included non-replicating malware from time to time for many years.

At the same *Virus Bulletin* conference at which Ducklin presented his paper, Vesselin Bontchev was already questioning the usefulness of the WildList [4], at any rate in the form in which it existed then. In addition to methodological problems such as cross-reporting, the data gathering process, and classification and naming issues, he also pointed to a number of testing problems that are worth summarizing:

- 100% detection of WildList malware is not necessarily a reliable indicator of overall detection performance.
- The WildList doesn't represent all the malware that meets Ducklin's definition: only that which is actually listed.
- What a tester understands to be WildList malware is not necessarily identical to similarly named malware in the current WildCore sample set.

Of course, this wasn't the last we were to see of any of these objections: in fact, one of the authors flagged [5] a 2007 test where a number of vendors were lambasted for missing 'in the wild' malware, apparently on the assumption that a similarity in detection name indicated a sample equivalent to a 'known' sample from WildCore (the set of actual samples generated by the samples listed in the WildList – a set that remains accessible only to a restricted group of trusted recipients). In fact, lax use of the term has resulted in its taking on more or less any meaning the tester wants it to, such as (paraphrase) 'presumed to be an In-the-Wild virus since there's a copy in my mailbox and my favourite scanner says it's malicious' [6]. However, this issue has become less problematic as WildList naming has moved [7] to a convention that does not easily map to vendor samples names, at any rate for anyone not in receipt of the WildCore sample set that reflects the items on the list.

```
W32/Sdbot!ITW#2693.
W32/Sdbot!ITW#2704.
W32/Slenping!ITW#4.
W32/Slugin!ITW#1...
W32/Sohanad!ITW#141
W32/Sohanad!ITW#146
W32/Sohanad!ITW#147
W32/Sohanad!ITW#148
W32/Sohanad!ITW#76.
W32/Spybot!ITW#294.
W32/Spybot!ITW#295.
W32/Taterf!ITW#100.
W32/Taterf!ITW#102.
W32/Taterf!ITW#103.
```

Figure 1: Extract from April 2010 WildList showing naming convention.

Do the other objections still apply? Clearly, the WildList is not a list of all the malware (self-replicating or not) that is in some sense in the wild at any one time. Even if it were or could be that comprehensive, by the time the lengthy validation and classification process is completed, the (more-or-less) monthly list would be even more hopelessly out of date by the time it was published.

While Bontchev commented, quite rightly, that the WildList's 'increased authority' in anti-virus testing entailed increased responsibility, that authority is greatly reduced in 2010. Has responsibility diminished along with that authority? Not, perhaps for the WildList Organization, which continues to represent a (maybe unrealizable) ideal, but many current sources of samples have never lived in a world where such an ideal seemed attainable. Indeed, one of the constant battles for AV vendors and testers alike is to somehow provide more than the most cursory validation of the viability (replicability) of the samples under test, let alone full validation – a goal for which the WildList (laudably) still strives.

HISTORY AND EVOLUTION

What does the history of the WildList tell us about the evolution of the threatscape? The WildList was conceived [1] in a more leisurely time, when viruses ruled the malware roost, definitions updates were monthly or even quarterly [3], most threats remained static in format (and even complex polymorphic malware didn't pose an insuperable problem), and so a collection of everything known to be a current viral threat was a viable objective.

Sarah Gordon's 1997 paper 'What is Wild?' [2] summarizes the development of the WildList in its early years (from 1993), and addresses the issue of what 'in the wild' (capitalization and hyphenation largely optional) means in that context. A simpler summary (including definitions) is still available at http://www.wildlist.org/faq.html, though it seems a little quaint in the current threatscape. In its first incarnation, it was aimed much more at the public, being made freely available 'in hopes to offset some of the "numbers games" being played by some anti-virus product developers'. In 1995, WildCore, the sample set that 'represents the threat to computer users' posed by those malicious programs included in the current WildList, was made available to *ICSA* and *Virus Bulletin* as a tool for regularized testing.

Do we care if malware replicates? Well, we do if it's supposed to replicate: if it fails to do so, it is not a virus (or worm or any other form of replicating sample).

It's important to note here that many viruses are dependent on the host environment. This is particularly true of worms, which may use a particular feature or vulnerability of the host operating system or application in order to replicate, but is also true of file infectors, boot sector viruses and other types of replicating malware. In other words, viruses may only be considered 'viral' within the environment in which they replicate. This is perhaps a rather fine point (and may well be disputed by some) but it is important in terms of testing. Indeed, there are many viruses which may only operate under very specific sets of circumstances, including requiring specific hardware. A simple example might be a virus that will not replicate under any version of *Windows* but will replicate under *RedHat Linux 4.1*. More commonly, viruses created for DOS executables would not infect *Windows* PE format executables correctly, if at all, and so on. There are also differences between the platforms – just because something will replicate on *Windows 9x*, it will not necessarily do so on *Windows NT* or its successors.

Therefore, for any virus to be considered in the wild in accordance with the WildList's own definition, it must be spreading as a result of normal day-to-day operations on and between the computers of unsuspecting users, whatever those computers may be running as an operating system. This leaves us quite a broad enough spectrum to include any and all replicating malware.

Importantly, this also makes a distinction between viruses that are actively spreading, and those which merely exist but are not spreading. Many viruses never make it into the 'wild'. In the early days, many virus writers would simply send their creations directly to anti-virus companies, as the act of creation was more to do with personal kudos than an active desire to cause destruction – these viruses were not considered to be in the wild. Nor are proof-of-concept malware, 'intended' viruses and so on, which have similarly restricted distribution and/or impaired functionality.

Of course, as we have already pointed out, viruses (we use the word here as a collective term for all the subsets of replicating malware) do not, today, make up the majority of the malware that we see. This has diminished the relevance of the WildList as a definitive list of spreading malware. There are circumstances in which trojans (non-replicating malware) have been added to the WildList, and similarly, for a trojan to be considered in the wild, 'it must be found on the computers of unsuspecting users, in the course of normal day-to-day operations', caveat the same discussion on environment as above.

So, yes, it still matters if viral malware replicates. If it fails to do so, it may or may not still be valid, but that state of being is no longer a purely technical issue of detection, but may also be a matter of design philosophy, where previously it was usually considered enough (by AV vendors, at any rate) to establish whether it was capable of replicating. Now, however, while self-replication is still considered a malicious characteristic, it's not the only malicious characteristic, and isn't even particularly common: some highly visible malware is spreading extremely aggressively and has become widely prevalent by means other than self-replication.

WHERE THE WILD THINGS ARE

In fact, WildList testing is rarely encountered in the context of the sort of one-off comparative test that we associate with magazines, but rather with organizations that offer some kind of certification service: notable examples are *Virus Bulletin*'s VB100 [8], *ICSA Labs*' Antivirus Product Certification [9], and *West Coast Labs*' Checkmark [10].

While it's sometimes claimed and often assumed that these organizations base their testing purely on WildCore, this is not actually the case.

While the VB100 is awarded on the basis of detection of 100% of in-the-wild malware (both on demand and on access) with no false positives, other detection tests are also incorporated into *Virus Bulletin* reviews, notably RAP (Reactive And Proactive) [11] testing, which measures product performance over time against 'fresher' malware [12].

ICSA Labs offers a range of certification services. While the anti-virus module [13] focuses on 'in-the-wild and other computer viruses' and doesn't include requirements regarding the handling of other forms of (non-replicative) malware, other modules may be used to assess a product's other security capabilities, such as anti-spyware and disinfection.

West Coast Labs has been redirecting its energies for some time now towards real-time testing, where samples are collected and tested continuously [14], rather than relying exclusively on WildCore samples, although they also offer a range of more static-based certifications. These often use WildCore as a base, but include their own supplementary sets of validated samples.

TUNING OUT THE STATIC

Static testing reflects a single point in time [15]. Just as you cannot step twice into the same river [16], a test set can only represent a snapshot of the threat landscape that may have some validity at that moment in time. A result based on that snapshot will probably not stay entirely valid for the next few hours, let alone for days or months to come.

A product's detection rates can vary [17], even against a single malicious program, in a way similar to that in which the value of stock-market-listed shares can rise and fall according to the economic environment (and do so on a frequent basis). Therefore, current success in a product's detection against a given list of viruses is not necessarily an indication of future performance (nor even of performance against an alternative list). There are two reasons for this variability:

- Companies often change or revise their detections, either to take account of false positives that might have arisen from the detection, or to bolster or improve previously incomplete detection e.g. to add to the number of close variants detected in a generic detection.
- It's not unknown for a scanner to drop detection for a given threat over time, though there may be a number of reasons for that: configurational changes, threat reclassification, or, of course, a processing glitch. The upshot, though, is that detection of a given sample may be successful or unsuccessful depending on timing.

Clearly, a static test against pre-collected, pre-validated samples using 'passive' scanning is less reflective of the real user's experience of the current threat landscape than a good dynamic test, though frankly we prefer to see a good static test than a dynamic test where the methodology is clearly faulty or where such issues as validation, classification and sample selection criteria are kept as a deep, dark secret by the tester.

While little work has been done in the area of examining detection/prevention over the course of an attack, the recent work of Igor Muttik is instructive in demonstrating such a methodology [15], and it is to be hoped that, in future, dynamic or real-time testing will be able to take account of such variations in detection.

VALIDATION, VALIDATION, VALIDATION

High volumes of samples are the enemy of careful validation, and it is worth noting that the rules that WildCore recipients are required to obey state that the recipient must always replicate the samples before any subsequent use is made of them. This has on occasion highlighted problems with an initial sample, and is one way to ensure the ongoing quality (in terms of validity) of the set. If nothing else, it ensures that a test of a product against a WildCore-based sample set will at least be a test against valid and viable replicating malware. We'll discuss validation more thoroughly later.

As soon as it becomes impossible to validate the samples within a test set individually (whether manually or by automated means), a larger margin of error is introduced and the method of collection becomes a major source of potential bias. Indeed, at this point it's JUNC (JUst a Nameless Collection). This is, of course, the strength (and weakness) of the WildCore set: it is fully validated, and this is why it takes so long to compile it – leading inevitably to accusations of its being dated when delivered. In this sense, the WildList's concentration on viruses makes things simpler (it's far, far harder to validate a trojan on any day of the week), but this is also a strength – we know (fairly) unequivocally that missed detection of a WildCore sample is truly an error of omission.

STATUS QUO, QUO VADIS?

Well, Status Quo will probably be continuing to deploy more or less the same riffs as they have since the 1970s. WildList testing, however, has to move beyond the tried and tested (no pun intended). Happily, this isn't news to the testers who still use WildCore as a major component of their methodology.

At this time such methodologies are largely associated with the more established certification specialists (*West Coast Labs*, *Virus Bulletin*, *ICSA Labs*), whose core functionality is based not on comparative testing (ascertaining which is 'the best' product) but on confirming that tested products reach an acceptable standard. However, in at least the above-mentioned cases, these testers have adapted, updated and modified their tests to include more modern and relevant testing methodologies. The fact that they continue to test based on WildCore is for the very good reason that many products continue to fail against that very benchmark.

So, why has this approach fallen out of favour with the anti-malware industry in general? Well, in fairness, this is a complex issue, and many products employ multiple methods, complementary to file scanning, that will, in an ideal situation, prevent the infection of a system. But where now for the WildList, and for the AV and product testing industries?

It is important, and almost certainly inevitable, that testing will evolve to a point where testers employ mathematical principles in calculating test scores rather than simply compiling lists of samples against which tested products either 'pass' or 'fail' detection. Knowing the margin of error in their sample sets (that

is, how much junk/clean/corrupt stuff is in the set) is also an important factor, as is the idea of assigning an importance to a sample under test. Perhaps some weighting could be based on knowing when and how (or if) the sample was validated: for example *fully validated*, *partly validated*, or *unvalidated*. Furthermore, working on tests that will produce a detection score over time, rather than at a static point, will inevitably show up where companies are failing to truly protect the users [15].

As things stand, the WildList reflects a static approach to detection and to testing in more than one sense, whereas modern malware frequently morphs, and as we have pointed out, is less frequently replicative in the traditional sense.

Let's be clear, though: when we discuss static testing, we don't mean that this excludes the possibility of testing that includes execution of the sample under test – indeed, 'on-access' tests were an attempt to address this. We mean rather that the malware itself is not collected during the test, or as part of the test the systems are not exposed to 'live' or 'real-world' scenarios that much more accurately reflect the way that users would encounter malware on their systems.

PERHAPS, PERHAPS, PERHAPS

Perhaps there is a way to update the protocols of the WildList to reflect the modern threatscape and to include a more representative portion of families along with some sort of prevalence-based data – indeed, internal discussions within the WildList Organization have mooted the inclusion of such capabilities for some time.

Perhaps the previously attempted idea of the 'real-time' WildList can be revived and reliably implemented (this is no easy task, and it is far from a criticism to mention that it was not particularly successful in its earliest incarnation) so that the time between updates of the WildList (currently monthly) can be significantly reduced. This, of course, raises the ugly question of further financing and staffing of what is essentially a public service to testers and AV vendors.

Perhaps when more companies have adopted and are publishing IEEE standard [18] malware metadata for sample sharing and tracking, these feeds can also be applied to creating a 'wildlist' for use in testing. It may also be that such a move could make regionalized reports more relevant – something that the WildList as it stands has largely lost the ability to do. Indeed, performance on regional threat detection is something that has gone largely unmeasured in testing, even though regional factors can have a major (but mostly ignored) impact on detection testing.

STILL CRAZY AFTER ALL THESE YEARS

So, is it wholly useless to base any test on WildCore? In our opinion it is premature to dismiss such testing entirely when there are still so many products which simply fail to achieve full detection in WildList-based tests, on a regular basis. Leaving aside the thorny question of false positives (which form a significant part of *Virus Bulletin*'s WildList-based VB100 certification), there are still very many failures, and many problems showing up in the test. In April 2010 John Leyden wrote a report [19] that highlighted the problems that arose for many products during testing against the *Virus Bulletin* set – where 20 of the 60 products failed to achieve the VB100 award (the actual VB100 award only depends on full detection of the WildList and generating no false positives, not [as is sometimes wrongly assumed] detection of all the malware in *Virus Bulletin*'s test set [or even all the malware in the world!]). As a basic benchmark of competence, surely such a test has a value, if only to back up the view of Vesselin Bontchev that:

> 'An AV product which detects 100% of the WildList viruses is not necessarily acceptably good. It is only not unacceptably bad.' [4]

CONCLUSION

'The problems that exist in the world today cannot be solved by the level of thinking that created them.' [20]

There are a number of conceptual problems with detection testing. For one, in most cases, it doesn't take into account an unascertainable margin of error. For another, it's largely based on the assumption that testers are better at collecting and selecting samples than AV vendors, and that assumption doesn't really stand up to scrutiny. There is a difference, though, between detection testing as it's carried out for certification purposes or to affirm that a product meets a better than minimum standard, and testing that's intended to ascertain which of a range of products is 'the best'. The fact that WildCore consists of shared samples is sometimes presented as weakening its validity, and even as somehow conspiratorial. But if WildList testing is so easy to 'game', why doesn't every vendor manage to achieve regular VB100 awards? Should it really be the tester's role to 'trick' vendors into false negatives with esoteric sample sets in the hope of separating out clear winners and losers, in a field where the margin for error of so many tests is already so high? When sample sharing between vendors (and indeed with mainstream testing organizations) forms a critical part of the industrial ecosystem that supports and (partially) protects the customer, it seems odd that this particular shared resource attracts so much criticism [21]. (Perhaps less so when it comes from competitors in the testing industry.)

Not so long ago we asked the question 'who will test the testers?' [22]: much of the general population seems to take it for granted that almost anyone who performs a test of some sort is qualified and equipped to perform accurate testing, and yet the ways in which the skill and professionalism of the tester can be tested remain limited. A lab's conformance with quality standards such as ISO/IEC 17025 gives some indication of its professionalism, though it doesn't directly assess the quality of the lab's samples and methodology. The Anti-Malware Testing Standards Organization (AMTSO) has a mechanism for assessing a test's conformance with the 'fundamental principles of testing' defined by the organization retroactively, but has shied away to date from offering any form of accreditation for testers (individuals or organizations).

It may be that until we see methodologically sound 'testing of testers' we will have to accept that there is still a place for 'best endeavours' testing based on sample sets that lack freshness but are known to have been validated. Clearly, though, certification

based entirely on detection of WildCore samples can no longer be regarded as a sufficient guarantee of a product's effectiveness. It's fortunate, then, that the reputable organizations making use of WildList testing are already using or moving towards methodologies that hold onto what is good about the WildList while adding functionality and value by bolstering it with fresher samples and more dynamic technologies.

DISCLAIMER

Although one of the authors is currently a reporter for the WildList, and both have a long association with the WildList Organization, neither they nor their employers have any financial interest or influential position in the WildList Organization or its parent organizations. The WildList is wholly operated by *ICSA Labs*, an independent division of *Verizon Business*. None of the opinions expressed herein necessarily reflect the views of those organizations, nor any other organization mentioned here. They are simply the personal opinions of the authors, based on their understanding of the current situation in WildList-based anti-malware testing.

REFERENCES

[1] http://www.wildlist.org/.

[2] Gordon, S. What is Wild? 1997. http://csrc.nist.gov/nissc/1997/proceedings/177.pdf.

[3] http://www.wildlist.org/aboutus.htm; http://www.wildlist.org/faq.htm; Ducklin, P. Counting Viruses. Proceedings of the 9th Virus Bulletin International Conference, 1999.

[4] Bontchev, V. The WildList – Still Useful? Proceedings of the 9th Virus Bulletin International Conference, 1999.

[5] Harley, D. Untangling the Wheat from the Chaff in Comparative Anti-Virus Reviews. http://www.eset.com/resources/white-papers/AV_comparative_guide.pdf.

[6] AntiVirus Fightclub Results! http://www.untangle.com/blog/?p=96.

[7] Harley, D.; Bureau P-M. A Dose by Any Other Name. Proceedings of the 18th Virus Bulletin International Conference, 2008.

[8] VB100. http://www.virusbtn.com/vb100/index.

[9] Anti-Virus Criteria. https://www.icsalabs.com/technology-program/anti-virus/criteria .

[10] Checkmark Certification and Platinum Product Awards. http://www.westcoastlabs.com/checkmark/.

[11] VB RAP Test Results. http://www.virusbtn.com/vb100/rap-index.xml.

[12] VB RAP Testing. http://www.virusbtn.com/vb100/vb200902-RAP-tests.

[13] Anti-Virus Overview. https://www.icsalabs.com/technology-program/anti-virus.

[14] Garrad M.; Jones P.; Myers L.; Parsons M. Paradigm Shift – From Static To Realtime, A Progress Report. Proceedings of the EICAR 2010 Conference.

[15] Muttik, I. A Single Metric for Evaluating Security Products. Proceedings of the EICAR 2010 Conference.

[16] Heraclitus: Fragment 41; Quoted by Plato in Cratylus. http://en.wikiquote.org/wiki/Heraclitus.

[17] Lee, A. Towards A Methodology For Providing Prevalence And Persistence Based Weighting In Anti-Malware Testing. Submitted. University of Liverpool.

[18] Weafer V.; Muttik, I. ICSG – Driving Security Standards, Practices and Industry Co-operation. http://standards.ieee.org/prod-serv/indconn/icsg/ICSG_pres.pdf.

[19] Leyden J. Third of XP security suites flunk tests: 'Crashes, freezes, hangs and errors' blight VB run-through. http://www.theregister.co.uk/2010/04/13/winxp_anti_malware_tests/.

[20] Attributed to Albert Einstein.

[21] Townsend, K. Product Testing: Valuable or Meaningless? http://kevtownsend.wordpress.com/2010/02/16/product-testing-valuable-or-meaningless/.

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