Real Performance?

Jan Vrabec & David Harley
ESET

About the Authors

Jan Vrabec is a Security Technology Analyst at ESET. He specializes in performance testing of ESET products. He has attained a Master's Degree from the Faculty of Electrical Engineering and Information Technology of the Slovak University of Technology, Slovakia, where he is currently working on his Ph.D dissertation and concurrently fills the role of thesis consultant. He has worked for a number of companies in research and development positions and authored chapters in industry publications, as well as research articles and conference papers.

Contact Details: ESET, spol. s r.o., Aupark Tower, 16th Floor, Einsteinova 24, 851 01 Bratislava, Slovak Republic, Europe, phone +421 (2) 32244218, e-mail vrabec@eset.sk

David Harley is Research Fellow and Director of Malware Intelligence at ESET, a member of the Board of Directors of AMTSO (Anti-Malware Testing Standards Organization), Chief Operations Officer for AVIEN (Anti-Virus Information Exchange Network) and an independent security author, blogger and consultant. In his copious free time he maintains the Mac Virus and Small Blue-Green World web sites, including blogs on Mac security, hoaxes and other security and non-security issues. He also blogs for Securiteam, (ISC)2, AMTSO and AVIEN. He has authored or co-authored over a dozen books on security, including "Viruses Revealed" and the "AVIEN Malware Defense Guide for the Enterprise" as well as many articles and conference papers.

Contact Details: c/o ESET, 610 West Ash Street, Suite 1900, San Diego, CA 92101, USA, phone +1-619-876-5458, e-mail dharley@eset.com

Keywords

Performance Testing, Evaluation, Testing Scenarios, Comparative Testing, Methodology, Benchmarking, AMTSO, Scanning Speed, Memory Usage, False Positives, Detection, Performance, Usability

Real Performance?

Abstract

The methodology and categories used in performance testing of Anti-malware products and their impact on the computer remains a contentious area. While there's plenty of information, some of it actually useful, on detection testing, there is very little on performance testing. Yet, while the issues are different, sound performance testing is at least as challenging, in its own way, as detection testing. Performance testing based on assumptions that 'one size [or methodology] fits all', or that reflects an incomplete understanding of the technicalities of performance evaluation, can be as misleading as a badly-implemented detection test. There are now several sources of guidelines on how to test detection, but no authoritative information on how to test performance in the context of anti-malware evaluation. Independent bodies are working on these right now but the current absence of such standards often results in the publication of inaccurate comparative test results. This is because they do not accurately reflect the real needs of the end-user and dwell on irrelevant indicators, resulting in potentially skewed product rankings and conclusions. Thus, the "winner" of these tests is not always the best choice for the user. For example a testing scenario created to evaluate performance of a consumer product, should not be used for benchmarking of server products.

There are, of course, examples of questionable results that have been published where the testing body or tester seem to be unduly influenced by the functionality of a particular vendor. However, there is also scope, as with other forms of testing, to introduce inadvertent bias into a product performance test. There are several benchmarking tools that are intended to evaluate performance of hardware but for testing software as complex as antivirus solutions and their impact on the usability of a system, these simply aren't precise enough. This is especially likely to cause problems when a single benchmark is used in isolation, and looks at aspects of performance that may cause unfair advantage or disadvantage to specific products.

This paper aims to objectively evaluate the most common performance testing models used in anti-malware testing, such as scanning speed, memory consumption and boot speed, and to help highlight the main potential pitfalls of these testing procedures. We present recommendations on how to test objectively and how to spot a potential bias. In addition, we propose some "best-fit" testing scenarios for determining the most suitable anti-malware product according to the specific type of end user and target audience.

Introduction

Clearly, evaluation and testing are not the same thing. While testing of a product's capabilities is sometimes an important part of the evaluation process, especially for a corporate customer, the time, resources and in-house expertise available to all but the largest customers are generally too limited to allow accurate and exhaustive hands-on testing of all aspects of a product's performance. Thus most potential customers base buying decisions on third-party tests, either commissioned from a presumed expert source or harvested from sources such as consumer or business magazines.

Detection is one of the primary functions of a malware-specific product or service, but only *one* of those primary functions, even though it can entail many facets such as raw detection of specific malware, proactive prevention of infection or compromise by malware not specifically identified by signature, and post-execution remediation in the event of a compromise.

Detection performance isn’t enough in itself (Lee & Harley, 2007). In fact, we will follow common industry practice here by distinguishing between detection and other aspects of performance by using the term “performance” to refer to characteristics such as memory usage, resource footprint and throughput speed *as opposed* to raw detection capability. This is because even though detection is critical, a product also needs to meet the needs of the customer in other ways, especially given the difficulties of realistic comparative evaluation of detection capability. (Vrabec, 2010; Harley, 2009a), so considerations such as those shown in Table 1 become critical.

• Usability, ergonomics and configurability	To suit the needs of both the system administrator and the end-user or home user.
Functional adaptation.	For instance, response to drastic change in the threat landscape such as a significant new threat vector: examples might include the dramatic rise of macro viruses in the 1990s, the surge in malicious email attachments in the first few years of the 21st Century, or the slower but even further-reaching shift from self-replicating malware to Trojans in past years.
Responsiveness to the needs of and changes in the organizational environment or infrastructure.	Examples might include modifications to the network, hardware and software upgrades and patches, realignment to changes in policy or strategy framework.
Responsiveness or adaptability to business needs	For instance, the impact of security software on host hardware and other applications, and therefore on day-to-day business processes.

Table 1: Primary Functionality of Anti-Malware Programs (Harley, 2009a)

There is, however, little guidance currently available on formal objective testing that addresses these issues in the specific area of performance testing (Harley, 2009b). Consequently, reviewers and their audiences tend to fall back on detection testing as the main criterion for comparative evaluation. “It is, after all, a core function, and offers a deceptively simple, apparently objective metric.” (Harley, 2009a).

There is a noticeable trend among mainstream reviewers (AV Comparatives, 2009) towards addressing some of these factors more formally. Generalist consumer and business magazines have, on occasion, attempted to evaluate such issues in parallel with detection testing (an approach that can stumble upon a number of potential pitfalls that we will attempt to address in the next section). Larger corporate organizations are often aware of and even focused on the need for procurement processes that take into account business and operational needs as well as more technical aspects of product evaluation: indeed, raw detection data may rank quite low in the priority list, given the common (and not entirely unjustified) perception that detection rates among mainstream products are roughly comparable.

Detection Testing Versus Whole Product Testing

Self-evidently, testing detection rates are not the same as whole product testing, and should not seen as such. We are not just referring to detection versus system impact, usability and so on. What we used to call ”anti-virus” now does much more than detect viruses or even the entire gamut of malware, of course. At least, mainstream commercial products do. But it also embraces a range of protective technologies that go far beyond simple blacklisting of known malicious code, even in products that are essentially marketed for their capabilities as regards protection from malware.

Other products are marketed as suites rather than anti-malware and include an even wider range of protective functionality. However, the more such functions a product has, the more necessary it is to take into account the impact of those additional functionalities on performance. And, unfortunately, the more difficult it becomes to keep the playing field level. As products become more complex, more technical understanding of the interaction between multiple functionalities and their impact upon performance is demanded of the conscientious tester. Or, at least it should be.

In practice, it's extraordinarily rare for a corporate evaluator to find comparative reviews that are not too subjective to be useful (or, like most consumer-oriented reviews, fixated on a subjective, one-size-fits-all perception of "good practice" that is expected to all individuals and types and sizes of enterprise. (Harley, 2009b; AMTSO 2010a)

If these interactions are not taken into account, it becomes practically impossible to establish a level playing field: an apples-to-oranges test (one that doesn't compare like to like) is of no real comparative value. Otherwise, it ceases to be a comparison of functionality, and instead becomes a comparison of design philosophies. It is widely assumed that the "fairest" test of a product is to use "out of the box" settings because these are the settings that will be used most. They *may* be the most commonly used settings (especially by home users): however, the use of default settings doesn't constitute a "level playing field." Even in detection testing, it means that products that discriminate between "possibly unwanted" applications (and other forms of "greyware") and out-and-out malware may be penalized when tested against programs that adopt a more aggressive approach to greyware, the possibly legitimate use of run-time packers, and so on. It can certainly be argued in performance testing that while default settings may be the most suitable in many contexts, that "in more complex solutions or more tailored tests testers may wish to discuss the required settings with the solution developers or the test clients." (AMTSO, 2010b)

Best Antivirus Solution

Vendor marketing departments are notorious for claiming that they have the best protection for everyone, but what is the "best" antivirus solution?

Leaving aside the fact that different stakeholders – home users, corporate end-users, the media, system administrators security researchers, vendors – may have very different perceptions of what is "best", it is reasonable to envisage an "ideal" solution which should offer the highest degree of protection to its user: the user should not notice any degradation of performance and when he needs to interact with the software it should be "user-friendly".

Due to the fact that typically only one line of product performance is assessed by one test, when all products parameters should be taken into consideration, we propose a triangle depicting this combination of parameters, see Figure1. The Detection, Performance and Usability of a given product should be at a maximum, but well-balanced. This means that no one of these indicators should override the other. *Very* important is a low count of false positives, in other words the "Type I" representation of the product's detection error rate. A false positive (incorrect classification of innocent code as malicious) is in a very real sense the obverse of a false negative, but Type I testing requires a different approach to Type II testing, and the two test types are best kept separate as far as possible. For example, speed testing for detecting known malware samples does not belong in the same test iteration as speed testing for scanning known clean files: if they are mixed, it becomes impossible to disentangle detection performance from speed performance, and detection of true positives may distort false positive reporting.

The basic methodology for measuring detection capabilities is very well documented in several guidelines and therefore will not be considered further in this paper except where detection issues impact upon system performance issues. Similarly, the evaluation of attributes, such as product's user friendliness and effectiveness of GUI is very subjective and does not lend itself well to making a methodology or guideline on this topic.

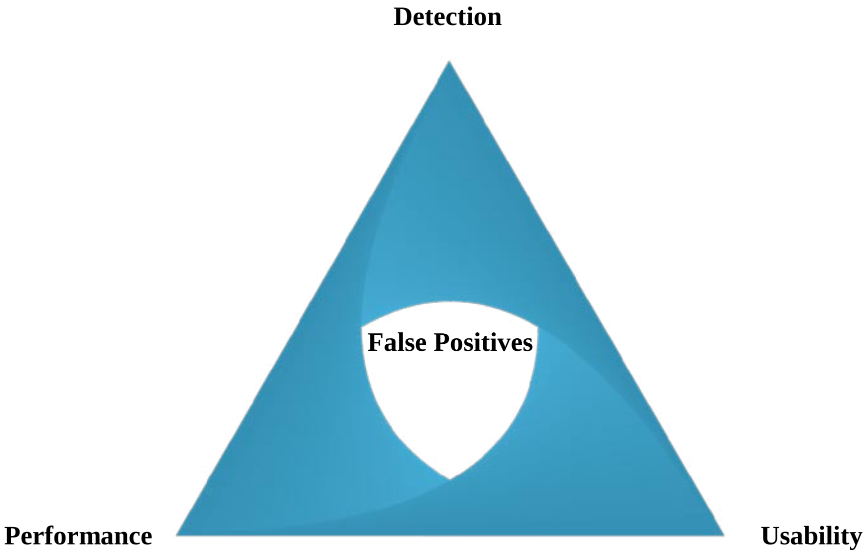

Fig. 1 Well-balanced protection

The measurement of performance and a product's impact on the system can at first look seem as a very easy task that requires only measuring time or disk space usage. But appearances can be misleading. If we peer deeper into the problem, we start to see that the methodology is very important to arriving at consistent testing results.

Scanning Throughput

The easiest way to benchmark the performance of an AV solution is to measure the scanning speed of a static sample set – containing only clean files. From the reader's point of view, it is a relatively easy test, but there can be major pitfalls in its correct implementation. One such pitfall is the selection of suitable sample sets. Sample sets used for scanning should be as representative of the real world as possible in terms of all types and sizes of files, and should only include clean files. Finding a suitable sample set that is really clean for all tester suites often proves to be a very difficult task. Moreover, if the sample set contains files contaminated with malware, it can extend the time needed for scanning, which may introduce bias into the testing. Sample sets may be split according to file types to provide separate measurements for different types of data. If a sample set is used a whole content of a primary hard drive, the tester should ensure that the sample size is not artificially increased by the antivirus product itself.

Taking multiple measures ensures the validity of testing results.

In the era of using caching, logging or other techniques that speed up scanning, it is very useful to measure the scanning time of new files, which were not hashed separately from files that have been altered in such a way. Measuring the time of first scan for multiple times can be very difficult: because the tester needs to use a new machine each time, we normally disable the hashing feature where practical. In any case, multiple iterations of a test are advised in order to compare "first run" performance to subsequent performance to accommodate caching, whitelisting and so, and/or to establish a longitudinal baseline that is normally a more useful metric in terms of real-life performance than a one-time scan.

Some solutions employ various techniques to skip over files, which may increase the scanning speed, but can also introduce the risk of not scanning those files that are infected. The testers should make sure that they check the log for those files that were accessed last to see whether the solution actually scanned certain file types.

Memory Usage

There are several ways – deceptively easy in concept – to measure memory usage of an AV solution. For example, one would think that all that is needed is to take the value in Windows Task Manager of the AV solution process. But this can lead to inaccurate or misleading results. Another, more accurate method, and the one most commonly used one by experienced testers, is to perform a baseline measurement of total memory consumption by an idle system without any security software installed, and then take the same measurement with the solution in place, again with the system idle. The tester should ensure consistency of using the same techniques for each test.

To maintain accuracy, the preferred approach is to take memory usage readings periodically after the computer is booted and take an average of the readings. The difference between the commit charge of a system with installed solution and the commit charge of a clean system should, in theory, represent the total memory consumed by the solution. However, this approach may not be fully accurate either, as the product under test could have reserved some memory space for other purposes, and may access more of this memory when performing activities such as scanning or updating. Throughout the testing procedure, the tester should make sure that all the suites are in the same state and the tests are repeated several times to ensure a high level of accuracy.

System Boot Speed

This test is the most controversial test of all. Security solutions need to be active on a system as early as possible in the boot process, and most local anti-malware solutions will have some impact on the system start up time. Some vendors have attempted to make their solution load after the computer was started, but this practice proved dangerous as the system was not protected during this vulnerable period.

Among the most significant issues the tester must face is to define exactly when the system is fully started, as many operating environments may continue to perform start-up activities for some time after the system appears responsive to the user. This issue can be resolved by waiting until the computer is in idle state and determining when the protection provided by the security solution is fully deployed. It should also be noted that if a USB drive or network is used, this can also have an effect on the boot-up speed. Most importantly, the tester should ensure that the configuration is the same for all tested products.

Irrelevant Testing

It makes sense to test and compare the above-mentioned performance aspects of AV products because they have a direct effect on the user interaction with a PC, but some performance indicators used in some tests are completely irrelevant because they cannot affect the performance in the slightest. Where such metrics are in use, testing such attributes as Registry Key Count, Process Count and others, giving significant weight to those attributes may give the tester the means of establishing more *differentiation* between products, but does so in an arbitrary fashion that doesn't really reflect superiority on the part of a higher-scoring product. .

Black Box Testing Suites

Some testers are trying to enrich their testing procedures by introducing new tests, which they claim to be a better reflection of actual user behaviour in several programs or games. The testing software emulates the mouse, keyboard and interacts with real programs on the machine. Such complete testing suites are readily available on the market and include programs, such as World Bench (http://www.pcworld.com/misc/worldbench/index.htm) or Passmark Performance test (http://www.passmark.com/products/pt.htm).

From the viewpoint of the tester, performing these tests is a very easy task that only requires hitting the start button: after few hours, the results are ready to be read out. The issue with these instant testing products is that the results of such testing are highly questionable. These suites are intended for use primarily to test the impact of hardware on the performance and usability of the PC. Although the testing suite may indeed have a justified reputation in the area of hardware testing, the testing of antimalware products is a complex, very delicate task.

While an anti-malware solution sometimes has a measurable short-term impact on performance on very specific operations that pose particular risks, it will also often have a negligible long-term impact to register in tests like this, and from the point of the user may be unobserved and quite irrelevant. (Do I really care if scanner A takes two seconds more than scanner B to check a large attachment or file download?) In fact, the statistical error of these measurements is often bigger than the differences among several competitors' products.

Commercial test solutions nevertheless assign a final mark or number as a result of such bad-fit testing, but interpreting such a number is difficult and not necessarily an accurate reflection of the product's capabilities. The best answer to this kind of "black box" testing tool is to develop one's own testing application, so that the tester knows exactly what is under the hood. We understand that this approach can prove to be a very difficult and laborious task: however, defending the methodology behind a black box test suite can be even more difficult, if not impossible. The bottom line is that a tester who doesn't know the nuts and bolts of his/her test can be very easily discredited. For a tester with the depth of knowledge that such testing really demands, a hands-on engineering approach may be easier to understand and customize to suit the specialist context of anti-malware testing, as well as easier to verify.

Malware Performance Testing by User Type

Each user is different, uses different applications, has different file types and uses his PC for different purpose. Can we make a default test scenario for each one? Definitely not. Should we attempt to create one testing scenario that fits all users? Again, the answer is "NO." The best way is to create models of PC users. Knowing full well that we cannot cover all users with our models, we want to at least give advice on how to create user-specific testing scenarios. At first, we can divide the testing scenarios into two categories; at times two or more models can fit one user type:

Now, we will try to describe the models and the respective tests summarized in Table 2. There are several types of tests that apply for all consumers, i.e. the types of users that don't particularly care about the antivirus they are using; they just want to be protected and get high performance out of the solution. Then, we can sub-divide the consumers into more detailed groups The "Surfer" sub-group and the "Gamer" sub-group. The former encompasses users who often visit websites, download files or watch video streams, and so on. The members of the gamer sub-group are mainly concerned with gaming, require a high FPS, and often encounter a problem when an antivirus product degrades their system's performance while gaming. Of course, for this particular user profile, pop-ups or scheduled scanning events running in the background are entirely inappropriate. Therefore, it is often the case that gamers disable their protection in favour of added FPS. The result is that once they do that, they are no longer protected and become exposed to web-borne threats. Any antivirus protection that aims to fit the gamer profile should be very light, with all tasks running in the background. Moreover, when playing online games, the antimalware system's latency on the network represents a very important metric. Similarly, for the home user and the average consumer, it is important that there are no slowdowns when sending and receiving e-mails, starting email client and opening documents, such as spreadsheets. Also, these users can engage in activities, such as editing video and audio files, converting files from one format to another, as well as running specific applications. Therefore, any relevant testing should take into account a whole range of factors and user actions.

Segment	**User**	**Proposed Tests**
Consumer	All	Boot time Memory consumption Installing common software applications Copying files to the system or to and from a local network resource
	Surfer	Browsing of web pages from proxy server Browser start-up time Viewing video files streamed from a Web server
	Gamer	Latency on the network Degradation of frame per seconds
	Worker	Downloading emails from server Email clients start-up Time of opening, closing, saving and copying documents Editing video and audio files Converting from one format to another Start-up times of specific applications
Corporate	Users	Simulation of work with common business software Time taken to open, process and close single or multiple documents and applications Network performance Accessing email or messaging services Web browsing Designing internal applications, procedures and implementations in-house.
	Administrators	Performance on File and mail servers, gateways

Table 2: Malware Performance Testing by User Type

The “Corporate User” is the second segment of the user group that can be sub-divided into two smaller sub-groups. Firstly, end-users working with business software and documents with focus on factors, such as file-handling performance and resource usage (the time it takes to open, process and close single or multiple documents and applications, network performance, data backup and moving files over a network). Other important activities to consider within this segment include Internet-related tasks such as accessing e-mail/messaging services and the Web, designing internal applications, and a host of in-house procedures and implementations. The second user sub-group is made up of the support staff and administrators who primarily deal with file and mail servers, gateways, and others – in short, delivering support and services to the “Corporate” segment.

Conclusion

With this paper, we intended to demonstrate on specific examples a simple fact - that even though measuring the impact of antimalware software can be viewed as an easy task, it is fraught with several pitfalls. Testers should always decide carefully which tests are relevant and if their measurement techniques are valid and objective. We believe that the activities of independent bodies within the testing and security communities, such as EICAR and AMTSO, result in the release of better information and general testing guidelines. These can help raise awareness across the board and help advice testers on how to employ sound techniques when measuring the performance impact of antimalware solutions. This may entail more work in some respects for testers and publishers, but ultimately it increases their credibility and value to prospective customers. What’s more, it also increases their value to the vendor community in that more accurate independent testing will give them an invaluable extra insight into the ways in which products can be improved to meet the needs of their customers.

References

Harley, D. (2009a). Making Sense of Anti-Malware Comparative Testing. Information Security Technical Report. Retrieved 10th March, 2010 from http://dx.doi.org/10.1016/j.istr.2009.03.002, Elsevier.

Harley, D. (2009b). Execution Context in Anti-Malware Testing. Conference Proceedings for 18th EICAR Annual Conference. Retrieved 10th March 2010 from http://smallbluegreenblog.wordpress.com/2009/05/15/execution-context-in-anti-malware-testing/

Lee, A.J. & Harley, D. (2007). Antimalware Evaluation and Testing. In D. Harley (Ed.) AVIEN Malware Defense Guide for the Enterprise (pp. 441-498): Syngress

AMTSO (2010a). AMTSO Whole Product Testing Guidelines (in preparation)

AMTSO (2010b). AMTSO Performance Testing Guidelines (in preparation)

ESET Research (2010). Retrieved 10th March 2010 from http://www.eset.com/blog/2010/01/25/generalist-anti-malware-product-testing

AV Comparatives (2009) Retrieved 10th March 2010 from http://av-comparatives.org/images/stories/test/performance/performance_dec09.pdf

Vrabec, J. (2010). Generalist Anti-Malware Testing (In preparation)