ANTI-MALWARE TESTING UNDERCOVER

Luis Corrons
Panda Security, Spain

Righard Zwienenberg
ESET, The Netherlands

Email luis.corrons@pandasecurity.com;
righard.zwienenberg@eset.com

ABSTRACT

Anti-malware testing has been always controversial, and it seems likely to stay that way. There are many different parties involved, each with their own agenda and (usually) a lot of money invested. Contrary to what most people believe, anti-malware testing is highly complex, and it becomes more and more challenging as new technologies are adopted by the industry to protect users. Rather than focusing on the technical challenges that testers face nowadays – a topic that has been discussed already a number of times at *Virus Bulletin* Conferences – we will focus on some other aspects that have not been presented before.

HOW DO TESTS INFLUENCE ANTI-MALWARE DEVELOPMENT?

It is normal, even desirable, for testing to influence the development of security solutions in such a way that they can improve in terms of protection, performance, usability and so on. However, this influence is not always beneficial, and we'll describe various real scenarios where vendors have been 'forced' to develop certain characteristics in their software in order to please testers, or at least to obtain good results in tests, even though those developments did not improve anything from the point of view of the end-users.

On-demand tests

Believe it or not, on-demand tests (scanning millions of inactive malware files on a hard drive) are still performed, even by reputable testers. This is an unrealistic, artificial scenario, and it does not really show how well a security solution protects users, as nowadays security solutions are complete suites with different configurations and levels of protection, where each product deploys different techniques to detect and remediate malware. Static (on-demand) testing is, of course, by far the easiest and cheapest way to test – it's a no-brainer, and that is why it is still carried out. Without going into too much detail as to why this kind of testing is no longer suitable, over the last three decades its continued use has forced anti-malware companies to dedicate resources to adding 'dumb' signatures or simple hash detections for huge chunks of inactive malware, just so their products perform better in these types of tests.

Performance

Previously, the on-access or real-time scanning component of anti-malware products would scan almost every file that was accessed on a computer in real time. That is, they would monitor any file operation (read, write, and so on) whether on the same computer or coming from the network. With good reason, anti-malware solutions were notorious for the way they caused computers to slow down. Performance tests have forced all vendors to work harder and improve in this respect by scanning files only when necessary from a security perspective (that is, when there is a possible risk of infection). However, as testers do test performance, rate performance for certification points, and publish comparative results (and a low ranking causes users to complain, and reduces sales), there is a growing trend for vendors to improve their results in speed tests by reducing the circumstances in which on-access scanning is triggered by default – which, of course, can also reduce detection and thus security. Some vendors actually see themselves as being forced by the testers to follow the trend and reduce safety levels in order to achieve better results in the performance test. In the end, how this situation is deal with is each vendor's decision. It would be logical not to measure performance alone (the impact of the security product on the machine performing a specific task), but to complement it with an efficacy test (does the product protect you when performing the same task with relevant examples of malicious code?). Only then can a valid comparison be made.

Real world

The tests that better reflect security solutions' ability to protect users are those that mimic what users face in their day-to-day computing. That's why the most reputable testers started some years ago to improve their tests and added 'real-world testing'. If vendors want to obtain a good score, they have to focus and spend resources on protecting their users – a win-win situation. However, we have seen many cases where this 'real world' is only seen in a tester's lab environment – in which case it is not real. But unless vendors are able to convince testers that a test is unrealistic, they need to take the scenario into account anyway, in order to obtain a good test score, even though the scenario is useless from the perspective of protecting the end-user.

Default configuration

Most testers use the default (out-of-the-box) configuration for the security solutions they test. This is valid for many reasons, including:

- In the home-user and SMB scenario in particular, the default setting is what the vendor considers to be the best for the user's safety.
- The tester can compare out-of-the-box settings to avoid running into certain configuration scenarios where vendors could get the tester to change settings to the vendor's advantage in the test.

As usual, there is a conflict of interests: better real-world protection versus better test scores. A number of well-known vendors have products where the default settings are not considered the best real-world protection for their users (the out-of-the-box settings are calibrated for better test results), and the vendor gives the end-user tips on how to change the default settings. Some even display notification messages

recommending/suggesting that configuration be changed for better protection.

Exploit detection

This kind of test is difficult to perform, and obtaining real-world exploits requires a lot of work. There are well known tools, such as Metasploit, that provide a framework to create some basic tests. However, when testers use these tools to measure how well security solutions protect against exploits, the tests fall at the first hurdle. It is very easy to create dumb detections for default Metasploit attacks, making it possible for a product to obtain excellent scores in exploit protection tests, while in the real world it would fail miserably. Yet, even Metasploit is used by some low-hanging cybercriminals, so of course security products should detect Metasploit attacks. If the security product is genuine and has real exploit detection capabilities, detection of Metasploit attacks should be taken for granted. Testing only with Metasploit attacks does not say anything about the broader exploit detection quality of a product. Sadly, the majority of testers stick to this kind of testing, rewarding products that detect Metasploit attacks regardless of whether they have real exploit detection.

HOW DO TESTS INFLUENCE ANTI-MALWARE DEVELOPMENT?

Not all side effects of anti-malware tests are bad. They can also help to bring about genuine improvement.

Finding bugs

As anti-malware tests can – due to the wide variety of malware used (almost simultaneously) – be somewhat heavy, testers sometimes find stability problems and bugs in the security solutions tested that have not been detected before either by the vendors' QA teams or by their users. Even though a large test includes more malware than a user will ever see in the lifetime of one system, this definitely helps to improve the quality of security solutions, as, with bugs, you never know when they will appear.

Malformed Email Project

A few years ago, a security researcher became aware of a security issue involving malformed emails that could prevent malware detection in certain circumstances. He and a tester co-operated on a research study, warning all the vendors about the issue and the tests they were going to perform, and giving them time to resolve any problems their products may have had with the issue. Even though it was not a real threat at that moment, if cybercriminals had become aware of the problem they could have exploited it before the industry had resolved it. More information about this project can be found at [1, 2].

We have cited several good examples of how anti-malware development is influenced by tests and testers. Is this a good or a bad thing? It can be said that as long as it helps improve security, it is good. But that's not completely true. At the end of the day, tests have a big influence on vendors' sales, so they can influence vendors' priorities in development cycles. Even in cases such as the aforementioned Malformed Email Project – which effected a clear security improvement for our industry – vendors were obliged to spend resources on addressing it on top of any other scheduled development. We can conclude that tests not only influence the development of security solutions, but also set a vendor's priorities – even though this is an involuntary and unintended side effect.

HOW DO ANTI-MALWARE VENDORS INFLUENCE TESTS?

What have vendors historically done and what do they currently do in order to influence testers? What is behind this lobbying pressure? In this section we will illustrate real cases showing what testers have to face in order to maintain their objectivity.

As well as the influence (both good and bad) that testers have on how security products are built, of course the opposite happens too. Vendors may lobby testers to do specific tests intended to create specific results favouring their products. Or they may simply ask the tester for another interesting 'incentive' to help them.

It is common for vendors to visit testers on a regular basis. There is nothing to hide, there are many valid reasons to do so, such as informing the testers about new products or forthcoming technologies, giving the testers time to update their testing frameworks so as to stay compatible, or helping testers to update their testing methodology when new technologies take over from old technologies. But of course those visits are also misused.

The following paragraphs have been composed with information from testers.

Certification

A vendor asks a tester to certify their product, the tester tests the product, the product performs badly and thus (logically) the tester does not grant the certification. The vendor is then willing to pay a large sum of money in order to get the certification anyway. The tester still (rightfully) declines, the vendor moves to another tester and suddenly gains a certification there.

Collections

Testers have been asked to sell their collections to vendors pre-test so the vendors in question will perform well. There are some creative variants to this one, as the following cases where one tester received messages from a couple of vendors:

- 'Can you run a test in which we provide all the samples and you see how our product compares against the competition?'
- 'Can you test us privately, but if we do well, will you please publish the results?'

Sometimes it is not about making suggestions, but direct threats. The following quotes come from a renowned tester we talked to while we were writing this paper, who had to face these threats:

- 'If you test us, or publish the results you have already obtained, we will sue you.'

- 'If you do not change the results to be more favourable, we will sue you.'

CHEATING

We'll cover anti-malware test cheating since the beginning of time, showing all kinds of different cheating by vendors, testers and publishers.

Vendors

People may assume that cheating is obvious, but there is a very fine line and sometimes it is hard to say whether cheating is actually going on or not. Some cheats are very obvious and undisputed, such as:

- Detecting everything as malware after a number of malware files are detected in a folder or on a hard disk.
- Creating specific versions of a product for testing when the test protocol demands exactly the same versions as are given to customers.
- Some testers organize their malware testbeds by naming the files by their hashes. There have been cases where products were not able to detect malware in a file with a 'normal' filename, while it was detected when the name of the file matched the file's hash.
- Identifying that a performance test is being carried out and disabling normal detection to increase speed.

We can also find some grey areas that might or might not be considered cheating in performance tests. For example, in the past vendors used to scan the computer before installation to remove any malware that could interfere with the installation process. That made installations long and tedious, and many vendors decided to skip this step, as they could scan the computer more effectively after installation (with all their technologies in place) and if needed they offered standalone tools to perform computer clean-ups without having the anti-malware installed. Recently, we have seen some vendors going back to the old days and running a scan during the installation process. Why? Because during this scan they create a cache of scanned files so that when the performance test is run they can run much faster than their competitors.

As mentioned earlier, vendors have changed the way their solutions work in order to improve their test performance. However, some behaviours are inherently more suspicious and more likely to be detected. It is a vendor's decision whether or not to scan files that are copied from the network. The attendant risks can be discussed, but in no case can it be considered cheating. However, when a security solution is unable to detect any malware copied from the network, yet it is able to detect the EICAR standard anti-virus test file… why is that? If it is the only thing the product can detect, no real malware is being detected, as it is not actually scanning anything coming from the network. Is this cheating? At the very least it is a deception, as it may make users believe that the protective program is actually scanning, when it is not.

Other borderline cases include those where vendors stop scanning when copying files on the same hard drive, on the same computer, and so on.

Tester/vendor side

This is a specific case that involves both testers and vendors. When measuring performance one thinks that tests will focus on the day-to-day things users do on their computers. However, we see that some tests focus on things such as the installation of productivity suites like *Libre Office* and *Microsoft Office* (for example), extracting and copying tens or even hundreds of thousands of files from one location to another – things that do, at some point, happen on a normal computer, but not on any kind of regular basis. How many times a week does the average user install *Office*? Is this cheating? Or is this done just because it is easy to do?

It is also very interesting to see the reaction of some vendors to this: sometimes when a product detects that an installer is being executed, it stops scanning at all. This entails a small risk for customers, and is intended not to improve the user's experience but to get better performance results in tests that measure performance during the installation of applications. Is this cheating?

Testers

Sample selection is one of the challenges every tester has to face. It can be argued that a bias can be introduced by factors such as prevalence and geographical distribution. Even though this by itself can hardly be considered cheating, what happens when there are malware samples used systematically by a tester that are never seen anywhere other than in their tests? Is that cheating from the tester's side? Could a third party be poisoning their testbed?

How do testers decide what samples are malicious? If they decide by pre-testing using anti-malware engines, even if it's not cheating, it's a flawed sample selection strategy that inevitably and inexcusably favours those vendors. When one or more of these vendors discover this, they can exploit it by polluting the sample set they send to the tester. Non-malicious files (that they create) are added to the sample set and 'detected', resulting in a miss for competing products. The tester and vendor supplying the samples to the tester are then, to all intents and purposes, partners in cheating. Another way vendors can influence the testers is by supplying them with clean software that they know is falsely detected by competitors.

There are different business models for testing labs, which in general are fine, unless the tester provide samples beforehand to 'premium' vendors (those who pay). They won't give the premium vendors the exact sample set that will be used in the test, but a larger one from which the premium vendors know the final sample selection will be taken. Doing this can guarantee premium vendors a good position in the results.

All products now use a cloud service of some kind for additional protection. There are lots of potential issues for testers when testing products with cloud services. If we start with the geographical location, a tester can experience slow-to-no responses from the cloud at the testing facility. The fact is that the product may only be sold in a specific region and cloud connectivity there may be extremely fast. Another side effect is the cloud response in different geographical regions. This may

be by design and implemented for good reasons, but it makes fair comparison rather difficult.

Another problem, however, is the licensing of scanner engines with cloud services. If Vendor Z is licensing its scanner engine with cloud services to Vendor A, and Vendor A's product is tested before Vendor Z's, Vendor Z has an advantage as details of all samples missed by Vendor A are now known to the cloud service, and can either quickly be analysed and detection created by Vendor Z, or the cloud is simply told that if the same file is scanned by a tester, to mark it as infected.

Publishers

The person responsible for writing magazine articles about anti-malware solutions has a lot of power. They can decide the weight given to each section of a particular test, which means they can almost build a custom result deciding who will be in the first or last position. This might not be common, and even if it were it would be almost impossible to prove. However, we have seen cases such as one where two magazines from two different countries used the results of the same test performed by a professional testing lab. In one of the magazines a particular vendor achieved first place. The other magazine listed that vendor in last place, and the vendor that had been in last place in the first magazine was this time in first place. Suspicious at least, and even more so when the vendor that suddenly got the first place in the second magazine was found to be spending lots on advertising in that particular magazine.

Cloud

Cloud technologies have been a game changer for most vendors, amplifying their protection capabilities and improving their response times. Testers have faced challenges in order to properly test anti-malware solutions with cloud capabilities. However, we will consider a different issue: cloud adoption has blurred the line of what is and isn't cheating, and testers may not be aware of all the information that vendors can have about their tests in real time. Now, not cheating is becoming something that depends on the good faith of each vendor. We'll cover the main risk scenarios where cloud technology is involved.

Security programs, by the nature of the tasks they're expected to achieve, try to monitor everything that's happening on a protected PC. While it's not the job (or the intention) of the vendor to retrieve and inspect all this information in individual cases, there is scope for misuse.

What computers are being used? What is being tested? When is it being tested? How is it being tested? Vendors have the power to obtain all this information, and more. You can see the content of the drives (files, folders, etc.), what URLs are being visited, and more.

What are the risks? Well, they are pretty obvious: vendors could change the configuration for that specific product in that specific machine (or licence key, or IP address range, or ASN even). They could add detections on demand (Vendor A knows that a certain part of a test is being carried out, and decides to detect something that it would not detect on any other computer in the world). Vendors can decide to withhold certain detections while false positive testing is being performed, and so on.

It is important to mention that big testing labs are run by professionals and they follow their internal procedures every time they run a test to be sure that everything is done properly and following their own quality standards. What's the problem with that? Well, if labs run periodic testing (e.g. monthly, quarterly), and each testing lab has its own favoured order in which to perform the different sections of their test (real world, false positives, performance, and so on), it becomes even easier to track what they are doing.

CONCLUSIONS

A number of security professionals are already aware of the technological challenges that surround anti-malware testing, but as we have shown there is much more happening under the testing umbrella than it might seem.

The main goal for all of us should always be the same: protecting the end-user. But it is a fact that the testing of security solutions influences their development, and that won't change. Vendors also try hard to influence testing in pursuance of their own interests, and as we have seen, some even threaten testing labs in an attempt to force them to accede to their demands.

One of the tools that can be useful to ensure that everyone behaves properly is the Anti-Malware Testing Standards Organization (AMTSO, http://www.amtso.org). Since its inception in 2008, this non-profit organization has developed various standards, addressing the global need for improvement in the objectivity, quality and relevance of anti-malware testing methodologies.

It is a point of contact for the industry, where vendors, testers and academia work together, fulfilling the goals of the organization:

- Providing a forum for discussions related to the testing of anti-malware and related products.
- Developing and publicizing objective standards and best practices for the testing of anti-malware and related products.
- Promoting education and awareness of issues related to the testing of anti-malware and related products.
- Providing tools and resources to aid standards-based testing methodologies.

REFERENCES

[1] Marx, A. Malformed Email Project. https://www.av-test.org/fileadmin/pdf/publications/vb_2002-11_avtest_paper_malformed_email_project.pdf.

[2] Marx, A. Malformed Email Project – Part 2. https://www.av-test.org/fileadmin/pdf/publications/vb_2003-02_avtest_paper_malformed_email_project-part_2.pdf.

[3] Jon Friedman, J.; Goretsky, A.; Zwienenberg , R. Truth or Spin in AV Testing. https://www.brighttalk.com/webcast/6213/46565/truth-or-spin-in-av-testing.