# Best Practices for Testing in-the-Cloud Security Products

## Notice and Disclaimer of Liability Concerning the Use of AMTSO Documents

This document is published with the understanding that AMTSO members are supplying this information for general educational purposes only. No professional engineering or any other professional services or advice is being offered hereby. Therefore, you must use your own skill and judgment when reviewing this document and not solely rely on the information provided herein.

AMTSO believes that the information in this document is accurate as of the date of publication although it has not verified its accuracy or determined if there are any errors. Further, such information is subject to change without notice and AMTSO is under no obligation to provide any updates or corrections.

You understand and agree that this document is provided to you exclusively on an as-is basis without any representations or warranties of any kind whether express, implied or statutory. Without limiting the foregoing, AMTSO expressly disclaims all warranties of merchantability, non-infringement, continuous operation, completeness, quality, accuracy and fitness for a particular purpose.

In no event shall AMTSO be liable for any damages or losses of any kind (including, without limitation, any lost profits, lost data or business interruption) arising directly or indirectly out of any use of this document including, without limitation, any direct, indirect, special, incidental, consequential, exemplary and punitive damages regardless of whether any person or entity was advised of the possibility of such damages.

This document is protected by AMTSO's intellectual property rights and may be additionally protected by the intellectual property rights of others.

# Best Practices for Testing in-the-Cloud Security Products

## Introduction

This document provides guidelines for testing anti-malware solutions which make use of 'in-the-cloud' technology. Its aim is to give an overview of the issues involved in the accurate testing of such technologies, and how tests may be designed so as to produce valid and useful test results. These guidelines are not a comprehensive listing of all such issues. Unless otherwise defined herein, all terms included in this document are used with their common meaning. The following document should be read in conjunction with AMTSO's *Fundamental Principles of Testing*, *Best Practices for Dynamic Testing*, and other information available on www.amtso.org.

## In-the-Cloud Technologies in Anti-Malware Solutions

Throughout this document, the terms 'cloud' and 'in-the-cloud' refer, respectively, to the internet (or other resources external to a protected system), and to resources and technologies run or served from there - online detection databases, reputation systems, black- and whitelists, managed services and so on.

Such technologies present significant difficulties to testers. Tests of standalone products have traditionally been run in a sealed environment – this gives testers absolute control over all aspects of their tests, allowing them to be completely repeatable and reproducible by anyone with access to the full range of resources used in the original test. However, as solutions make more and more use of cloud-based technologies, testing such solutions fully and fairly requires that they be given access to these external resources, forcing the tester to cede control of the test environment. Essentially, running such tests requires allowing external parties and mechanisms to influence the test environment, and tests can thus no longer be accurately or reliably reproduced. When running comprehensive tests of multi-layered solutions, tests can include a wide range of scenarios, referred to throughout this document as 'test cases', and in many situations some aspects of the test case itself may not be under the tester's control.

Beyond the simple loss of control over the test environment, a number of other factors come in to play when running tests connected to the live internet, many of which will impact the design and implementation of appropriate tests. Here some of these issues are addressed, along with some suggested techniques for overcoming them, or at least minimizing their impact on the accuracy of test findings.

## Test Environments Are No Longer Controlled and Reproducible

With the implementation of in-the-cloud technologies, it is no longer possible for testers to prepare a standard test environment and to run each product under test in exactly the same conditions, one after another, with the option of rerunning a test in those same conditions at any time. Solutions utilizing web-based resources cannot be 'frozen', as these remote resources operate beyond the tester's control.

To compare such solutions in a fair and balanced way, the traditional methodology of direct comparison of detection rates in static on-demand scans becomes difficult to implement, and a different approach based on statistical measurement of performance against non-identical sets of test cases may be more appropriate.

### Direct Comparatives

In the direct comparison model, all products under test are exposed to the same selection of threats, usually large sets of malicious samples for static testing and smaller sets for dynamic testing. In both of these test types, when applied to solutions with resources hosted online, test cases should be synchronized as closely as possible, in order to avoid biasing the test. Such a bias may be introduced where products are tested sequentially rather than in parallel, since products tested later in the sequence have more opportunity to discover, analyze and add protection against a given threat.

Scanning large sample sets may introduce a further bias even when scans are started simultaneously, as slower products again get more time to update their protection. It would be preferable to run multiple smaller scans, or ideally to test each sample as a separate scan or test case. This will of course be more time-consuming, but will produce more accurate results with less risk of bias.

### Statistical Comparatives

Even when tests are synchronized as closely as possible, and use the most up-to-the-minute samples, there will inevitably be differences in the response time of solutions and the behavior of test cases themselves, especially when testing dynamically against active malware with its own remote resources and influences. For example, when a test includes a malicious sample served from a given URL, the nature of that sample at any given time is subject to changes outside the control of the tester. The impact of these anomalies on test results can be reduced by running large numbers of test cases over an extended period, and using statistical data to provide an accurate reflection of each solution's relative performance.

When such statistical data is collected in large enough quantities, the need for direct comparison is reduced. By measuring a solution's performance against significant quantities of threats over a longer period of time, performance quality can be accurately compared even when solutions have been tested against different sets of threats at different times, as the effect of anomalous test cases should be leveled out. This approach avoids many of the synchronicity issues raised by testing against larger numbers of test cases in a shorter time frame, and also enables the tester to perform more realistic dynamic testing. Such testing can be run continuously, with periodic reporting of results, providing an in depth picture of performance over time.

### Virtualization

Testing multiple products simultaneously and over long periods is inevitably more resource-intensive than running multiple tests in series on a small number of test systems in a short period. In some cases virtualization may be an appropriate means of reducing the costs of such testing, but it may also introduce further problems as not all solutions are fully supported in virtualized environments. Running multiple solutions in parallel in virtual environments on a single host also presents issues for speed measurement, as VM prioritization may mean that not all guest systems are run at the same speed. This

approach may also impact tests in other ways, as competition for network and other resources may affect detection levels. It may be more appropriate to take a statistical approach here also, running multiple speed tests at different times and recording average speeds for each solution under test.

In the case of dynamic testing, malware samples may also exhibit anomalous or uncharacteristic behaviors in virtualized environments, as discussed in the 'AMTSO Best Practices for Dynamic Testing'.

## Test Environments Cannot Be Sealed Off for Security

One of the prime tenets of anti-malware testing, 'first do no harm', is often best maintained by keeping test environments sealed off from external networks. When testing solutions with components hosted in the cloud, it is no longer possible to maintain complete separation from the internet, and so alternative methods must be implemented to minimize the risk of unintended leakage of malicious activity.

### Connection Filtering

The simplest way to allow solutions access to remote resources is to filter traffic at the gateway, allowing complete connectivity only to the specific URLs, domains, ports and protocols required by solutions. Other traffic, including malicious test cases, can then be blocked, redirected, throttled or otherwise filtered according to the specific malware-handling policies of the individual tester and the requirements of the test.

When designing such filtering, it is important to consider the properties of the protection technologies being tested, and the possible impact of any variations from real-world behavior imposed by the test environment on the protection provided. Protection technologies may analyze a range of behavioral information including URLs, domains and IP addresses communicated with, communication protocols, formats and patterns, and much else besides, and any manipulation of test case behavior, such as emulation of services or spoofing of online resources, may impact the performance of the solution if not implemented with complete transparency and accuracy. At the very least however, test environments should not allow test cases to perform actions which could present a danger to the public, such as spamming, denial-of-service attacks or worm propagation.

Solution developers are encouraged to share with testers any necessary information about online resources and communication protocols used by their solutions, and the types of behavioral data analyzed.

### Data Leakage

Another effect of allowing solutions to transfer data outside of testing labs is a potential loss of privacy. It may not be appropriate for specific data held in test labs to be accessible externally, including details of sample collections and other potentially confidential information, depending on the policies of the individual testing body.

To minimize the problems posed by such potential data leakage, testers may wish to obtain further information on the operation of solutions under test, such as what kind of data is transmitted externally, and in what manner; it may also be necessary to decrypt or otherwise analyze traffic between solutions

and remote resources, to ensure that no private data is leaked. Again, developers may wish to provide testers with details of the internal workings of solutions, their remote resources and the communication between the two.

Testers may find it useful to monitor all traffic between their test environments and remote locations, to gather their own independent verification of requests made and responses provided. Such data on the traffic produced by the communications of solutions under test may provide a useful measure of the comparative bandwidth load imposed by different technologies.

## Other Issues

### The Internet as Part of a Test Environment

Solutions which make use of online resources must access those resources via the internet, which is outside the control of the tester. Variations in connectivity may have an impact on both the protection provided and the speed of solutions. Location of the test network in relation to online resources may affect response times, so it may be useful for testers to publish the location of their test facilities. Web connectivity service providers may employ measures which filter or throttle connections between test labs and online resources, and appropriate service agreements should be made ensuring providers are aware of the probability of unusual traffic patterns from test labs.

Changing providers and test locations during testing, while perhaps desirable, may be impractical, and the use of in-the-cloud hosting services to run tests may be a way of avoiding location and connectivity issues, but such an approach brings its own issues. Testing on virtualized systems may affect the behavior of both solutions and test samples, as discussed above, and the policies of hosting providers regarding the introduction of malware into their networks should always be consulted and carefully observed, particularly in dynamic testing.

### Reliability of Protection

As well as speed implications, network connectivity can impose problems of resilience on solutions under test. Some solutions may offer dual modes of operation, with local detection technologies supplemented by additional remote resources. Some test scenarios might usefully test resilience to network problems, and compare the levels of protection offered when access to remote resources is removed, as may occur due to intermittent networking or intentional blocking of connection to a system for security purposes.

Testing for full coverage of all malware types is also important when testing solutions using online resources. Online databases of known bad files allow a much broader range of files to be identified specifically, but such technologies may not provide full protection against some malware, such as polymorphic viruses. When such samples are used in tests, they should be replicated in significant numbers by testers, to properly test the capability of solutions to handle these kinds of malware effectively.

## Sample Selection

Sample selection is a significant issue in all test methodologies, and to fully test the responsiveness of real-time systems samples should be as 'fresh' as possible. In most tests, maximum freshness can be achieved by testing solutions against all available samples and performing sample validation later, with only success or failure against proven valid samples taken into consideration when reporting results.

## Version Information

Test reports should, where possible, provide detailed information on the specific products and solutions included in tests, including version details, so that their audience can tell exactly what is being tested. When running solutions over long periods, with some portions of them managed remotely, such version information may not be available, or may change frequently as various components are updated. In such cases, testers may wish to provide version information only from the beginning and/or end of the test, along with details of the timeframe of the test. A clear policy on how this will be handled should be defined as part of test methodology.

## Trust Relationships with Developers

It may be useful for testers and solution providers to reach a level of trust. As discussed above, details of the technology and resources used in a given solution will help testers measure its performance more accurately, while safely controlling malicious samples. In some cases, solutions may respond anomalously to the unusual activities of a test environment, particularly during the running of largescale static tests. Developers may be able to correct for this if provided with details of a test, to ensure their solutions behave as they would in a normal environment.

It may also be useful for testers to have access to logging data gathered at the server side of solutions, and indeed some test formats such as auditing rely heavily on such data. To enable such exchanges of sensitive information, trust relationships, possibly including formal non-disclosure agreements, may be required. Trust between testers and solution providers is best fostered by open communication. One of the core aims of AMTSO is to encourage and facilitate such openness.

## An Example Methodology for Testing in-the-Cloud Solutions

A sensible comparative test methodology for host-based anti-malware solutions utilizing in-the-cloud technology may include the following steps:

- The test environment is configured to allow all traffic between test systems and online resources provided by their developers, using the specific domain/port/protocol requirements of the product. This traffic is monitored and logged at the test environment's gateway, and if sufficient information is available to testers, may be decrypted and parsed.
- Solutions selected for testing need not necessarily all make use of in-the-cloud technologies, but those which do not should be given equal access to external networks for their more traditional updating etc. All solutions under test are installed on identical test systems, using a consistent policy of setup and configuration. A matching control system is used as a reference for threat and system behavior.

- The traffic produced by test cases is controlled and filtered to properly manage malware risks, in accordance with the policies of the individual test body and the requirements of the specific test type. This filtering is designed to minimize impact on threat behavior and solution data exchange, including such details as IP addresses and URLs used by malicious test cases.
- Arrangements are made with providers of internet connectivity to ensure testing is not interfered with, and ISPs may be changed periodically during the test. Where necessary to ensure accurate testing, developers are made aware of the design, methodology and/or timing of the test, so they can ensure that full logs are kept and that anomaly filtering does not bias the performance of products.
- Each system is then exposed to a set of test cases comparable in quantity, type, severity, and significance. These may be identical and synchronized as closely as possible, but may alternatively achieve statistical equivalency through the running of large numbers of test cases over an extended period, possibly even continuously.
- Both malicious and false positive test cases are performed. The most recent malicious samples are used, for best measurement of responsiveness and handling of new threats not previously seen by solution developers, and where necessary post-test validation of samples is performed to ensure only data on valid threats is included in test reports.
- Data on test results is gathered from test systems and test environment gateways, and possibly also provided by solution developers. After appropriate parsing and interpretation, data is stored for traceability and, where appropriate, provided to solution developers for their own analysis. Test results are published, making clear the location of the test environment and other factors affecting networked solutions.

---

This document was adopted by AMTSO on May 7, 2009