# AMTSO Performance Testing Guidelines

amtso™

Anti-Malware Testing Standards Organization

## 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.

# AMTSO Performance Testing Guidelines

## Introduction

This document provides guidelines for testing the performance, in terms of speed and resource usage, of anti-malware solutions. Its aim is to give an overview of the issues involved in the accurate testing of such aspects of security 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.

Throughout this document the terms 'performance' is used to refer generally to the performance of a product or solution purely in terms of speed and resource usage. Unless otherwise defined herein, all other 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* and other information available on www.amtso.org.

### The Purpose of Performance Testing

When measuring the quality of a security solution, whether on its own or in comparison to other solutions, many aspects beyond the level of security provided may be of interest. Of these, perhaps the most universal and commonly analyzed is the performance of the solution in terms of speed and resource usage. In any environment, memory, CPU cycles, network bandwidth and storage space are finite resources; products which use more than can be spared in a particular environment may be unsuitable for that purpose regardless of their ability to provide protection.

This makes the measurement of such factors highly significant to the end-users of many tests, with different types of user having different interests and requirements depending on how their systems, networks and security solutions are used. Such testing should be approached methodically with the aim of producing balanced, relevant and accurate data.

## General Guidelines for All Types of Performance Testing

There are a number of different types of performance testing, measuring different aspects of performance in different ways. For all of these however there are some basic steps which should always be taken to ensure relevant and accurate measurements.

### Balance

As in all testing, performance measurements should be balanced, unbiased and fair.

- Test Coverage

The selection of metrics should be as broad as possible to provide the best overview of performance, and to avoid biasing the results in favor of a solution which excels in a specific aspect of performance. When producing rankings of compared products based on performance and resource usage, the selection of metrics used to produce rankings should be weighted appropriately. The selection of metrics used, and the importance given them in the ranking system, should reflect the significance of those metrics in the environment of the target audience of the test.

It is also important that speed issues are not given exclusive attention at the expense of other aspects. The purpose of all security solutions is to provide security, and this should be the primary criterion in evaluating such solutions for any purpose. Data on speed and resource usage should not be provided on its own, but should where possible accompany or be in some way linked to corresponding data on the protective abilities of the solutions analyzed.

- Test Equivalence

In comparative testing, all solutions should be subjected to an equivalent set of tests in the same or equivalent testing environment. If multiple test systems are used, the hardware and components used should be as identical as possible. Operating systems and installed software should also be the same – cloning or imaging an original master system to all test machines is a good way of ensuring equivalence.

Automatic updating of systems should be disabled in most cases, as the updating process itself may impact testing and the resulting changes to the system may lead to significant differences from one test run to the next. Other factors affecting the system performance should also be taken into consideration, such as bootup optimization – in most cases it is best to ensure test systems are used for some time to ensure all such optimizations are complete before commencing testing.

It is important to ensure that solutions are comparable in terms of release timeframes as well as in functionality. Solution developers may use different methods to label their products, for example in some cases applying a new year label several months before the start of that year, and in others keeping the previous year label several months into the following year. Testers should be aware that such labels should not be used as the only criteria for product selection, and that more detailed information from developers may be needed to properly select products with concurrent release and use periods.

## Relevance

Performance measures should be relevant to the requirements of the intended audience of the test, and test scenarios should mirror real-world situations as closely as possible. For most types of test, it is important to first define the target audience, and to consider what aspects of performance will most affect that audience.

- ### Appropriate Use Cases

In general, all performance measurements should be taken while handling 'clean' samples or test scenarios, as this should be the main experience of most users of security solutions. Testing scanning speed over infected samples may be of interest in some highly specific cases, but such cases are extremely rare and for the vast majority of users handling of clean samples and scenarios will provide a more useful measure of performance. The quantities, types and sizes of files used should be tuned to reflect the real-world situations of the intended audience; measuring the impact of home-user solutions on the operation of business software may not be very enlightening.

Selection of test scenarios and sample sets for performance measurements should focus on accurately reproducing real-world scenarios. They should include a similar balance of behaviors, data types and file sizes as seen in the typical environment of the intended users of the test results. It may be useful to define user 'personas', with specific needs and usage habits, and to base the selection of test metrics on the activities carried out most often by these characters. For example, a home user running desktop software may be interested in how the security solution impacts their ability to play games or watch movies, but wouldn't worry so much about on-demand scanning of large numbers of files which would most likely only be run when the system is not in use; an enterprise admin with large file servers to protect may find data on scanning speed and related resource usage very useful, but would not need information on how solutions affect music players  or photo management software.

Settings should reflect the most common configuration in use in the real world. In most cases the default settings for a given solution will be the most appropriate, but in more complex solutions or more tailored tests testers may wish to discuss the required settings with the solution developers or the test clients.

- ### Appropriate Test Environment

The typical environment of the intended users of the results should also be reflected in the test setup. In general real 'bare metal' systems will provide more accurate and relevant measurements than virtual environments, which may affect performance in a number of ways.

Tests specifically designed to measure performance in virtual environments should make this very clear to their readers.

For tests not focused on a specific user group, the most typical hardware in common use should be used for the test. Performance measures taken on extremely high- or low specification systems will have very limited value. In general, a medium-range system will provide the most broadly applicable results.

When selecting hardware, testers should be aware that systems with pre-installed operating systems and other software may introduce bias into tests, as some solutions may be optimized for certain setups, particularly those on which trial solutions come pre-installed by the OEM provider. To avoid this risk, testers may find it preferable to remove pre-installed operating systems and install their own environment from scratch.

When preparing test platforms, it is usually necessary to ensure that solutions are only tested in environments which for which they provide full and official support. While in some circumstances there may be some value in testing edge cases, to see how products perform in non-standard situations, for most consumers of test data such information is of limited value.

Full details of all hardware, operating systems and additional software used in tests should be made available with test reports. As in all testing, full details of solutions tested should also be provided.

## Accuracy

All tests should strive to provide the most accurate results possible, and in comparative testing it is vital to ensure all solutions are tested equally.

- ### Baselines

In many kinds of performance testing it is important to take baseline measurements against which the performance of a protected system can be compared. Such measurements should be taken in the same environment, including hardware and software setups and test tools, as that in which solutions will be tested.

- ### Avoiding Anomalies

Most forms of performance testing should be run multiple times. In a single run results may be skewed by inconsistent and anomalous activity; an average for multiple runs will minimize the impact of such anomalies and provide more accurate results. Testing that requires networking activity is particularly vulnerable to the impact of external forces, and tests across a network should, where possible, be performed with exclusive access to that network to minimize such issues. One simple technique for removing anomalous results is to run each test multiple times and removing the highest and lowest results, averaging the remaining figures.

It is also important to take into account optimizations carried out by both operating systems and security solutions. Many products will avoid re-scanning known-clean files, so may demonstrate large decreases in scanning times once a system or file-set has become familiar. Operating systems may also change the way data is accessed or processed when the same activities are repeated, and the effect of all such optimization techniques should be considered when designing tests and presenting results.

- Appropriate Configuration

For most types of test, the tester may choose to use default, out-of-the-box settings, 'best-possible' settings designed to measure the maximum detection levels of a solution, or the most appropriate configuration for a given environment, often specified by the solution developer. Default settings are the most common and generally most appropriate for measuring performance. However, testers should be aware of certain differences between performance and other types of testing in this regard.

For many other types of testing of anti-malware solutions, such as measuring protection levels, it is appropriate for the purposes of the test to adjust some settings which will not affect the data being recorded. Most notably, many tests will require logging levels to be adjusted to ensure adequate information is gathered.

For performance testing, any such adjustments to logging or other features may impact the performance of the solution under test, and testers should take such possibilities into consideration when designing tests and selecting configuration adjustment procedures.

Updating of products should also be considered; for many solutions, access to external resources will be necessary to fully test the real-world performance or performance impact of the solution, and most solutions will require updating of some kind. It is normally desirable to ensure products are fully updated prior to commencing performance tests, as some solutions may initially install with minimal content which will allow them to operate more quickly than when fully updated and operational. Testers should also be aware of the potential impact of scheduled update attempts on performance tests; in some circumstances it may be appropriate to disable scheduled updates to avoid having the additional activity influence the performance measures, while in other types of tests it will be necessary to allow updates to operate normally in order to measure their contribution to the data recorded.

- Reproducibility and Auditing

Details of test design and setup should be fully documented, and all logging from solutions under test and test tools should be kept to allow tests results to be checked and reconfirmed at a later date.

## Some Types of Performance Test

There is a wide range of aspects of performance which may be of interest to the end-users of test results, and testers should try to cover as many as possible to ensure their results are as comprehensive and relevant as possible. Not all metrics will be relevant to all types of solution. The following test types represent the most common and useful aspects to test. Along with some advice on how best to approach them are some of the pitfalls to watch out for when designing a test.

### Solution-Specific Factors

Some factors are specific to the solutions themselves, rather than their environment. Many aspects of solutions' composition and behavior can be measured and compared, but some of them have limited value. Measures of the time taken to install solutions, the size of installers and installed packages, counts of new registry keys created and so on are of some technical interest to specialists, but present little value to the general reader. Similarly, aspects such as user interface launch speed, and general reaction time of the interface, which may seem significant to testers who interact with products in an unusual way, are of minor interest to

most users who will spend little time interacting with their security solutions. Such aspects may be more significant for some specialized audiences, for example when testing corporate components such as large-scale management and deployment solutions, but provide minimal useful data for most.

### Scanning Throughput

The simplest and most commonly performed type of performance testing, measuring of file scanning speed for a static anti-malware scanner presents few major pitfalls. Sample sets used for scanning should be as reflective of the real world as possible, in terms of the types and sizes of files included, and should only include clean files. Sample sets may be split into file types to provide separate measures for different types of data. Taking multiple measures will avoid anomalous results.

As many solutions include techniques to maximize throughput by caching or logging of previously-scanned files, it may be useful to produce separate measures for 'cold' scanning over new files and 'warm' handling of samples already checked by the solution. The 'cold' scan will be considerably more time-consuming to run multiple times, but some solutions may offer the ability to disable caching or to flush caches, which may assist the tester here.

Such caching systems, while providing increased scanning throughput, may also introduce protection problems. If not designed properly, scanners which ‘skip’ previously scanned files could potentially allow infected files to go unscanned. It may be valuable to test for this, ensuring that files added to caches or other systems are fully scanned after each update.

Some throughput speed tests may make use of an entire disk drive or partition as a sample set; for example, using the standard system partition should provide a simple and easy sample set, containing a representative set of files and file types. However, testers should be sure to consider differences in sample sets caused by files introduced or altered by the product under test itself.

## System-Wide Factors of General Interest

Some factors will be of interest to most users of security solutions, but testers should remember to consider the specific needs of their target audience and measure these factors in a way which reflects the usage patterns and interests of the readers.

## File Access Time

At the simplest level, the impact of an anti-malware solution on the time taken to access files can be measured either on read (simple file access) or on write, depending on the specific requirements of the test, and a test scenario involving copying sample sets from one area of the protected system to another will exercise both forms of on-access protection. The overhead imposed by security solutions can be measured by comparing the time taken to perform an identical activity, such as reading, writing or copying files from one location or device to another, or running a dedicated test tool, on protected and unprotected systems. Sample sets should include only clean files, and tests should be run multiple times to ensure accuracy.

Such tests will only give a limited picture of the on-access performance, and tests more closely reflecting real-world activities will provide more useful information. Various more complex measurements of on-access protection are possible, such as measuring the time taken to install an application, or any other common activity regularly performed by a typical user. This can be performed in a similar way by comparing two sets of times, but results may also be affected by other aspects of the solution’s performance, including its memory and CPU usage. If installing software requires Internet access to download or register components, fluctuations in the speed of the Internet connection may also have an effect on speed of the installation, and as with most types of speed tests multiple runs may be the best way to smooth out any such anomalous results.

### Memory Usage

There are a number of systems in use for measuring memory usage, a complex issue with modern computer design as memory allocations come in a variety of types and may fluctuate considerably depending on demand. Many measures may thus produce inaccurate or misleading results.

One of the simplest and most commonly used measures of memory usage is to take a baseline measure of total memory usage by an idle system with no security software installed, and then take the same measurement (using the same measuring tool) with the solution in place, again with the system idle. The difference between the two should be the total used by the solution.

Measuring total memory use is far simpler and less prone to error than attempting to measure memory use of the specific product under test separately; many products run multiple processes, which may not be active at all times, while different methods of measuring memory use may report different types of memory use differently. Processes can reserve and use memory in different ways, with the total private use, shared and shareable covered separately by some measurement tools and combined by others. Built-in tools provided with operating systems may even change the way they report different types of use from version to version. Testers should ensure they use accurate measurement tools, focus on appropriate rubrics for memory usage and apply their measurement procedures consistently.

For more accurate measurements, multiple or continuous measures of memory can be taken over a period of time and an average produced. This approach can also be taken with solutions in an active state, for example while new files are being written to the system or when running on-demand scans. In cases such as on-demand scans where an activity cannot be performed without the solution installed for comparison, a comparison between a range of products can give a useful indication of relative memory usage.

### CPU Usage

This can be measured in a similar way to memory usage, with the same provisos and potential pitfalls. Again attention should be paid to the methods and tools used to measure and record usage, to ensure a useful and accurate measurement is obtained.

As with memory usage measurements, recording the decreased availability when the system is idle, with and without a solution installed, provides a simple measure of resources used by a security solution. Similar measures can also be taken during some strenuous activity designed to exercise the solution, as long as the activity can be accurately reproduced; multiple runs of all such tests, including those recorded on bare test environments as control measurements, will help ensure greater accuracy.

### Network Overhead

Security solutions which filter network traffic, whether for filtering emails, monitoring web traffic entering a network or system, or watching for malicious files passing between two nodes of a LAN, will often have high levels of traffic and will thus be required to provide fast throughput with minimal overheads.

Measuring these overheads can be performed in a similar manner to measuring on-access speeds at a local level, by comparing the time taken to transfer sample data from point A to point B with no filter in place with the time taken when the filter is interposed between A and B. Taking the average of multiple measurements, both for the benchmark and the solution overhead, will provide the most accurate results. Care should be taken to ensure no other network activity can influence the results; ideally an isolated network should be used for such tests.

Most security solutions need to be updated frequently, pulling down new data from the internet or local network. The amount of data downloaded and the frequency of update checks can vary greatly between solutions, and in some situations this additional network overhead may be a significant factor in selecting a solution.

Measuring the bandwidth used for updates can be performed by monitoring the network connection between the protected system of network and the external source of data and recording details of each connection. Care should be taken to differentiate updating activity from other network activity, and to reduce noise during the test it is advisable to keep other activity on the network to a minimum.

The same approach can also be applied to solutions which access internet-based databases to check for records on files or URLs – again, traffic should be carefully parsed to accurately measure what traffic represents what activity.

## Factors of Interest to Specific User Groups

Some factors may be of interest to certain groups of users, or appropriate only when testing certain types of solutions.

### System Boot Speed

Security solutions need to be active on a system at an early stage, and most local anti-malware solutions will have some impact on the time taken for the system to start up. Measuring boot times accurately presents the tester with many difficulties however. The most significant issue the tester must face is to define exactly when the system is fully started, as many operating environments may continue to perform startup activities for some time after the system appears responsive to the user. One reasonably scientific way to measure this is to monitor CPU usage and disk activity and judge that the boot is complete when both are idle for an adequate period of time, and various tools are available which can be used to take such measurements.

It is also important to consider when the protection provided by the security solution being tested is fully active, as this could be a useful measure of boot completion as far as the security solution is concerned. Some products may also wait for the system to become idle before they start their activities, which would cause problems for measurements discussed in the previous section; however, such an approach may leave systems unprotected for a period, and testers may find it interesting to check for such weak spots.

If a USB drive or network is attached to the test system, this can also influence the boot speed. Therefore, the tester should ensure that the configuration is the same for all products, including the availability or otherwise of removable or networked resources. This type of measure will be primarily of interest in tests of desktop solutions

### System Shutdown, Restart, Hibernate and Recover Speed

The impact of security solutions on shutdown, reboot and hibernate times may also be of interest to some target audiences. As with the boot speed test, the tester should be sure to set a firm and clearly-measurable definition of when the machine has completed its shutdown or hibernation process, and apply it consistently across all solutions tested. When measuring startup or recover times, again it is vital to set a specific and measurable definition of completion.

In general, these factors, as with boot time, are primarily of interest when testing desktop products, as servers and appliances tend to be restarted much less frequently.

### Battery Drain and Power Consumption

High levels of activity on mobile devices can increase the speed at which the battery drains. Testers may wish to measure the solution’s ability to operate efficiently under limited resources. This can be measured by running as series of predefined, ideally automated tasks on the test system and recording how long the battery lasts. The same set of activities should be performed on an unprotected reference system to show the difference in drain time imposed by the protection solution.

The device battery should be charged to the maximum before the test is performed. Like most performance-related measurements, repeating the test multiple times will produce more accurate and reliable results. However, the tester should take into consideration the possibility of the battery's capacity decreasing after large numbers of charge-drain cycles. The impact of this degradation in battery performance can be countered by running the test once for each product being tested, then continuing to put each product in turn through a single run of the test. It may help to rearrange the order in which products are run for each iteration of the test.

Again, this measure will only affect solutions intended for use on mobile devices. However, the influence of security solutions on power consumption in general can also be measured, using the standard pattern of comparing the power usage of an unprotected system with one running the solution being tested. A range of power usage meters are available at reasonable cost. Note that in devices which also have batteries, it may be advisable to remove the battery when performing mains power tests, as periodic battery charging activities may have a significant effect on the amount of power used.

## Impact On Specific Activities

One of the most useful ways to measure performance is also one of the most demanding – measuring the impact of security solutions on everyday tasks. The selection of tasks should be made carefully to reflect the normal activities of the intended audience of the test. Single tasks on their own will provide limited information, but a well-balanced collection of tasks can give a useful indication of the impact of security solutions on everyday system use.

Most of the following activities would be appropriate for both home-user and business desktop solutions:

- Opening document files in Microsoft Office or other popular office suites

The time taken for the viewer or editor application to open and a document to be displayed can be measured, with a comparison drawn between the opening time with and without security software in place

- Opening a PDF file in a popular PDF viewing or editing application

As with office document files, PDF files are common in both business and home-user situations. Again the time to open the application and display the document can be measured and compared with that on an unprotected system.

- Browsing popular Web sites

Ideally using an automated system, a fixed selection of browsing activities can be performed with and without the solution in place and the difference in page load times compared. The influence of uncontrollable fluctuations in internet speed can be countered either by repeating measures in large quantities at different times to achieve statistical validity, or by directing traffic to a caching proxy controlled by the tester to ensure network traffic remains the same for all tests. When taking this approach, testers should ensure proxies are properly configured to allow solutions access to any external resources which may impact their performance.

Tools are available to measure the time taken for pages to load; some types of web page may continue to load content without ever having a true 'complete' time, and testers may want to exclude this type of page from tests for logistical reasons.

- Downloading files or emails from the internet

Documents, archives and other files are commonly downloaded from the internet or received as email attachments, both at home and in business. The added time taken to download to the local system can be measured. The same process can be used for uploading files or sending emails.

- Startup time of browsers and email clients

Web browsers and email clients are commonly used in all situations. The impact of security solutions on their startup time can be measured compared with that on unprotected systems. Testers should be sure to define a specific and measurable point at which the software is adjudged to be fully operational. In the case of web browsers, a blank page might not provide a proper measure of real-world user experience, and a real page may be preferable; to ensure that this is loaded at the same speed for each solution tested and each test iteration, the proxying steps described above may be useful.

- Copying files to removable or network drives

USB storage devices and network shares are commonly used in most environments. The time taken to write or upload batches of sample files can be measured.

- Creating or unpacking archive files

Archive formats are commonly used for file storage, and the impact of security solutions on the time taken to create new archives or to decompress files from existing archives is a legitimate point of interest for most users.

The following activities may be appropriate for specific groups of users:

- Editing video and audio files and converting from one format to another

Images, video and sound are commonly stored and manipulated on home systems and in some types of business, and editing and converting such files can require a large proportion of system resources, so measuring the slowdown imposed by security solutions can be of great interest to many home users and to those corporate users whose businesses are required to perform such activities.

- Viewing video files streamed from a Web server

Viewing streaming video, and other forms of streamed media, may be affected by some security solutions such as web filters. Measure the impact of filters will mainly be of interest to home users, as few businesses require their employees to stream live media regularly.

- Installation and removal of software

Third-party applications provided either as download or on media such as CDs and DVDs provides a thorough workout of some aspects of anti-malware solutions, as many application installers come in the form of large MSI of self-extracting executables. Measuring the time taken to complete an installation can provide a useful measure of overheads imposed by security solutions, which may be most applicable to home users, as business systems are less likely to have new software installed frequently.

- Opening and operating business software

Solutions such as CAD applications and other design software, database viewing and accessing software and so on are more commonly used in business than in a home context, and are therefore more appropriate for tests of solutions designed for corporate use. Measures can be taken of the time taken for applications to fully load, and in many cases of performing specific activities, which may involve network activity. Some applications may have very specific and targeted user base, while others are more generic and details of how these are affected may be of interest to a wider group of readers. Most such applications have little relevance to the home-user market.

## Benchmarking

Some testers may wish to combine a selection of the above test types into a single benchmark measurement reflecting a typical user's interaction with a protected system. A number of dedicated benchmarking solutions are available, but most are designed to measure hardware rather than software performance, and few will produce data in a format which can be readily tied to real-world values. Testers can instead design and automate their own suite of tests, which should be designed to include factors of significance to the intended audience. A number of tools are available to help take these measurements and to automate the process of taking them.

For home users for example, the selection may include web browsing and downloading files, accessing emails, copying files around the system or to and from a local network resource, installing software common applications and encoding media files. A test of products for a corporate market may want to focus more on business activities, running common business software such as spreadsheet and document tools, design applications and so on, alongside more standard activities such as accessing email and backing up files over a network.

These suites of tests can then be run with and without security solutions in place and the times compared to find the overall system impact of the solution. In all cases, running repeatable tests multiple times, removing anomalous results and average the remaining figures will provide the most accurate picture of true performance.

______________________________________________________________________________

This document was adopted by AMTSO on May 25, 2010