

**WEBROOT**<sup>®</sup>

# Webroot SecureAnywhere<sup>®</sup> AntiVirus vs. Six Competitor Antivirus Products

(March 2016)

Windows 10

Antivirus Performance Benchmark

**Document:** Webroot Secure Anywhere AntiVirus vs. Six Competitor Antivirus Products (March 2016)

**Authors:** M. Baquiran, D. Wren

**Company:** PassMark Software

**Date:** 9 March 2016

**File:** Webroot\_SecureAnywhere\_vs\_antivirus\_competitors\_2016\_Edition\_1.docx

**Edition:** 1

# Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>2</b>
<b>REVISION HISTORY</b> .....	<b>3</b>
<b>REFERENCES</b> .....	<b>3</b>
<b>EXECUTIVE SUMMARY</b> .....	<b>4</b>
<b>OVERALL SCORE</b> .....	<b>5</b>
<b>PRODUCTS AND VERSIONS</b> .....	<b>6</b>
<b>PERFORMANCE METRICS SUMMARY</b> .....	<b>7</b>
<b>TEST RESULTS</b> .....	<b>10</b>
BENCHMARK 1 – SCAN TIME.....	10
BENCHMARK 2 – SCHEDULED SCAN TIME.....	10
BENCHMARK 3 – INSTALLATION SIZE.....	11
BENCHMARK 4 – INSTALLATION TIME.....	11
BENCHMARK 5 – REGISTRY KEYS ADDED.....	12
BENCHMARK 6 – BOOT TIME.....	12
BENCHMARK 7 – USER INTERFACE LAUNCH TIME.....	13
BENCHMARK 8 – INTERNET EXPLORER LAUNCH TIME.....	13
BENCHMARK 9 – MEMORY USAGE DURING SYSTEM IDLE.....	14
BENCHMARK 10 – MEMORY USAGE DURING INITIAL SCAN.....	14
BENCHMARK 11 – BROWSE TIME.....	15
BENCHMARK 12 – FILE COPY, MOVE AND DELETE.....	15
BENCHMARK 13 – NETWORK THROUGHPUT.....	16
BENCHMARK 14 – FILE COMPRESSION AND DECOMPRESSION.....	16
BENCHMARK 15 – FILE WRITE, OPEN AND CLOSE.....	17
<b>CONTACT DETAILS</b> .....	<b>18</b>
<b>APPENDIX 1 – TEST ENVIRONMENT</b> .....	<b>19</b>
<b>APPENDIX 2 – METHODOLOGY DESCRIPTION</b> .....	<b>19</b>

## Revision History

Rev	Revision History	Date
Edition 1	Initial version of this report, includes new results for six (6) antivirus products.	9 March 2016

## References

Ref #	Document	Author	Date
1	<a href="#">What Really Slows Windows Down (URL)</a>	O. Warner, The PC Spy	2001-2016

# Executive Summary

PassMark Software® conducted objective performance testing on seven (7) antivirus software products, on Windows 10 (64-bit) during February and March 2016. This report presents our results and findings as a result of performance benchmark testing conducted for these consumer antivirus products.

The aim of this benchmark was to compare the performance impact of Webroot's SecureAnywhere AntiVirus product with six (6) competitor antivirus products. Testing was performed on all products using fifteen (15) performance metrics. These performance metrics are as follows:

- Scan Time;
- Scheduled Scan Time;
- Installation Size;
- Installation Time;
- Registry Keys Added;
- Boot Time;
- User Interface Launch Time;
- Internet Explorer Launch Time;
- Memory Usage during System Idle;
- Memory Usage during Initial Scan;
- Browse Time;
- File Copy, Move and Delete;
- Network Throughput;
- File Compression and Decompression; and
- File Write, Open and Close.

## Overall Score

PassMark Software assigned every product a score depending on its ranking in each metric compared to other products in the same category. In the following table the highest possible score attainable has been normalized to 100. This would be the score given if a product attained first place in all fifteen (15) metrics. Antivirus products have been ranked by their overall scores:

Product Name	Overall Score
Webroot SecureAnywhere AntiVirus	74
Norton Security	60
McAfee AntiVirus Plus	58
Windows Defender	57
Kaspersky Anti-Virus	48
MalwareBytes Anti-Malware	41
Trend Micro Antivirus	42

## Products and Versions

For all products, we have tested the full retail release of the most current and publicly available version of each antivirus product. The names and versions of products are given below:

Manufacturer	Product Name	Release Year	Product Version	Date Tested
Trend Micro Inc.	Trend Micro Titanium Antivirus + Security 10	2016	10.0.1186	March 2016
Webroot Inc.	Webroot SecureAnywhere AntiVirus	2016	9.0.8.72	March 2016
Symantec Corp	Norton Security	2016	22.5.5.15	February 2016
Microsoft Corp	Windows Defender	2016	1.215.678.0	February 2016
McAfee, Inc.	McAfee AntiVirus Plus	2016	14.0.6136	February 2016
Kaspersky Lab	Kaspersky Anti-Virus	2016	16.0.1.455(a)	February 2016
Malwarebytes Corp	Malwarebytes Anti-Malware	2016	2.2.0.1024	March 2016

# Performance Metrics Summary

We have selected a set of objective metrics which provide a comprehensive and realistic indication of the areas in which an antivirus may impact system performance for end users. Our metrics test the impact of the antivirus software on common tasks that end-users would perform on a daily basis.

All of PassMark Software's test methods can be replicated by third parties using the same environment to obtain similar benchmark results. Detailed descriptions of the methodologies used in our tests are available as "[Appendix 2 – Methodology Description](#)" of this report.

## Benchmark 1 – Scan Time

All antivirus solutions have functionality designed to detect viruses and various other forms of malware by scanning files on the system. This metric measured the amount of time required to scan a set of clean files. Our sample file set comprised a total file size of 982 MB and was made up of files that would typically be found on end-user machines, such as media files, system files and Microsoft Office documents.

## Benchmark 2 – Scheduled Scan Time

This metric measured the amount of time required to run a scheduled scan on the system. The scan is set for a particular time via the client user interface.

## Benchmark 3 – Installation Size

In offering new features and functionality to users, antivirus software products tend to increase in size with each new release. Although new technologies push the size limits of hard drives each year, the growing disk space requirements of common applications and the increasing popularity of large media files (such as movies, photos and music) ensure that a product's installation size will remain of interest to home users.

This metric aims to measure a product's total installation size. This metric is defined as the total disk space consumed by all new files added during a product's installation.

## Benchmark 4 – Installation Time

The speed and ease of the installation process will strongly influence the user's first impression of the antivirus software. This test measures the minimum installation time required by the antivirus software to be fully functional and ready for use by the end user. Lower installation times represent antivirus products which are quicker for a user to install.

## Benchmark 5 – Registry Keys Added

A large registry increases a machine's use of resources. This may negatively impact system performance, especially on much older machines. This test measures the amount of keys and values added to registry, after rebooting the test machines, following a successful product installation. Lower numbers mean that a product has added fewer keys during installation and had less impact on the registry.

## Benchmark 6 – Boot Time

This metric measures the amount of time taken for the machine to boot into the operating system. Security software is generally launched at Windows startup, adding an additional amount of time and delaying the startup of the operating system. Shorter boot times indicate that the application has had less impact on the normal operation of the machine.

## Benchmark 7 – User Interface Launch Time

This metric provides an objective indication as to how responsive a security product appears to the user, by measuring the amount of time it takes for the user interface of the antivirus software to launch from Windows. To allow for caching effects by the operating system, both the initial launch time and the subsequent launch times were measured. Our final result is an average of these two measurements.

## Benchmark 8 – Internet Explorer Launch Time

This metric is one of many methods to objectively measure how much a security product impacts on the responsiveness of the system. This metric measures the amount of time it takes to launch the user interface of Internet Explorer 11. To allow for caching effects by the operating system, both the initial launch time and the subsequent launch times were measured. Our final result is an average of these two measurements.

## Benchmark 9 – Memory Usage during System Idle

This metric measures the amount of memory (RAM) used by the product while the machine and antivirus software are in an idle state. The total memory usage was calculated by identifying all antivirus software processes and the amount of memory used by each process.

The amount of memory used while the machine is idle provides a good indication of the amount of system resources being consumed by the antivirus software on a permanent basis. Better performing products occupy less memory while the machine is idle.

## Benchmark 10 – Memory Usage – Scan

This metric measures the amount of memory (RAM) used by the product during an antivirus scan. The total memory usage was calculated by identifying all endpoint protection software processes and the amount of memory used by each process during an antivirus scan.

## Benchmark 11 – Browse Time

It is common behavior for security products to scan data for malware as it is downloaded from the internet or intranet. This behavior may negatively impact browsing speed as products scan web content for malware. This metric measures the time taken to browse a set of popular internet sites to consecutively load from a local server in a user's browser window.

## Benchmark 12 – File Copy, Move and Delete

This metric measures the amount of time taken to move, copy and delete a sample set of files. The sample file set contains several types of file formats that a Windows user would encounter in daily use. These formats include documents (e.g. Microsoft Office documents, Adobe PDF, Zip files, etc), media formats (e.g. images, movies and music) and system files (e.g. executables, libraries, etc).

### Benchmark 13 – Network Throughput

The metric measures the amount of time taken to download a variety of files from a local server using the HyperText Transfer Protocol (HTTP), which is the main protocol used on the web for browsing, linking and data transfer. Files used in this test include file formats that users would typically download from the web, such as images, archives, music files and movie files.

### Benchmark 14 – File Compression and Decompression

This metric measures the amount of time taken to compress and decompress different types of files. Files formats used in this test included documents, movies and images.

### Benchmark 15 – File Write, Open and Close

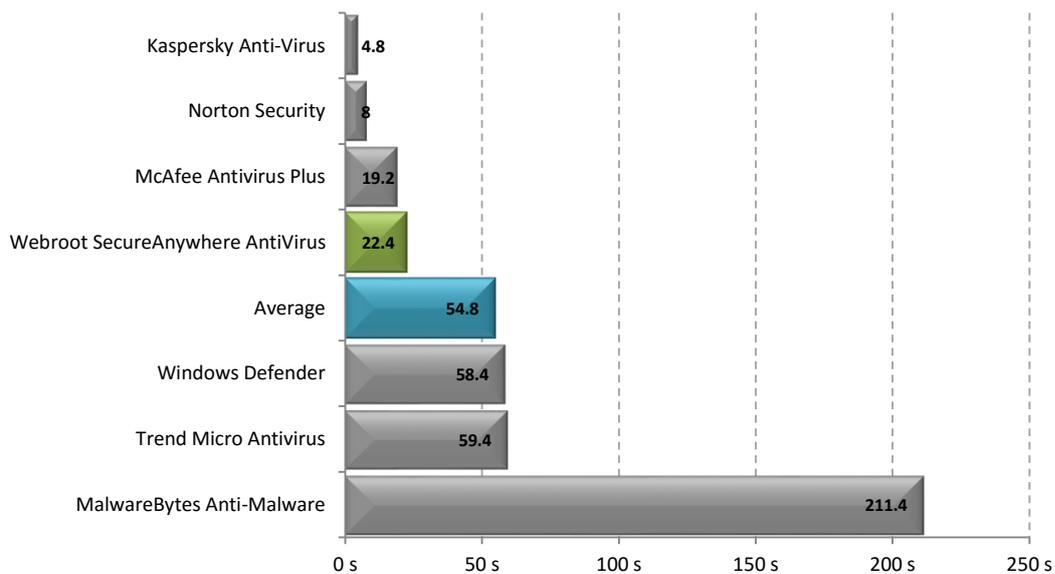
This benchmark was derived from Oli Warner's File I/O test at <http://www.thepcspy.com> (please see *Reference #1: What Really Slows Windows Down*). This metric measures the amount of time taken to write a file, then open and close that file.

# Test Results

In the following charts, we have highlighted the results we obtained for Webroot SecureAnywhere AntiVirus in green. The competitor average has also been highlighted in blue for ease of comparison.

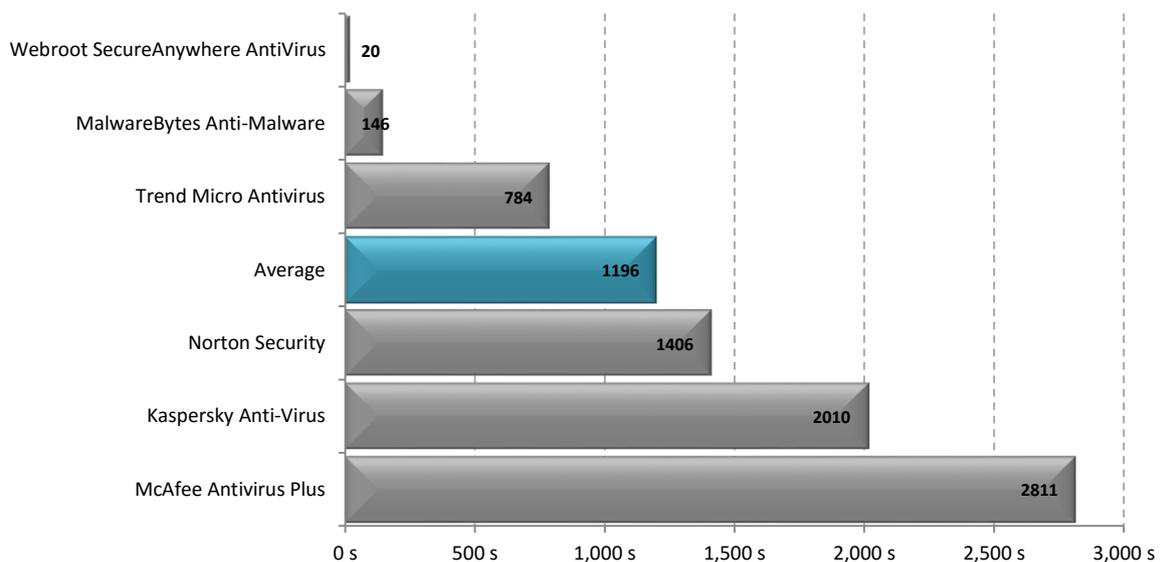
## Benchmark 1 – Scan Time

The following chart compares the average time taken to scan a set of 6159 files (totaling 982 MB) for each Antivirus product tested.



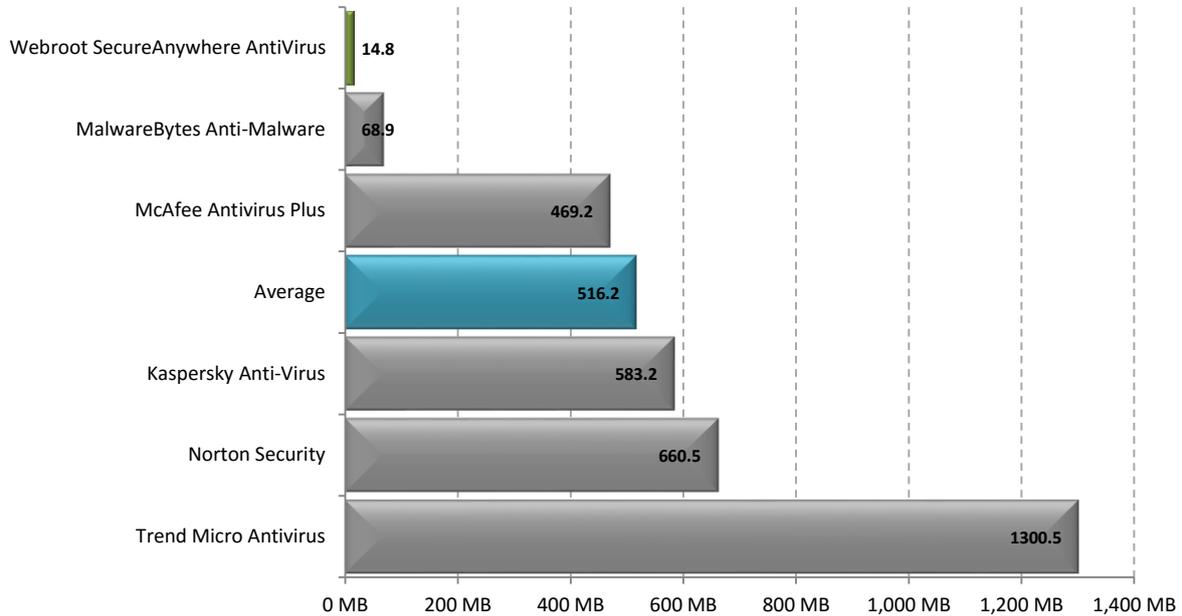
## Benchmark 2 – Scheduled Scan Time

The following chart compares the average time taken to run a system scan at a specified time. Products with lower scan times are considered better performing products in this category.



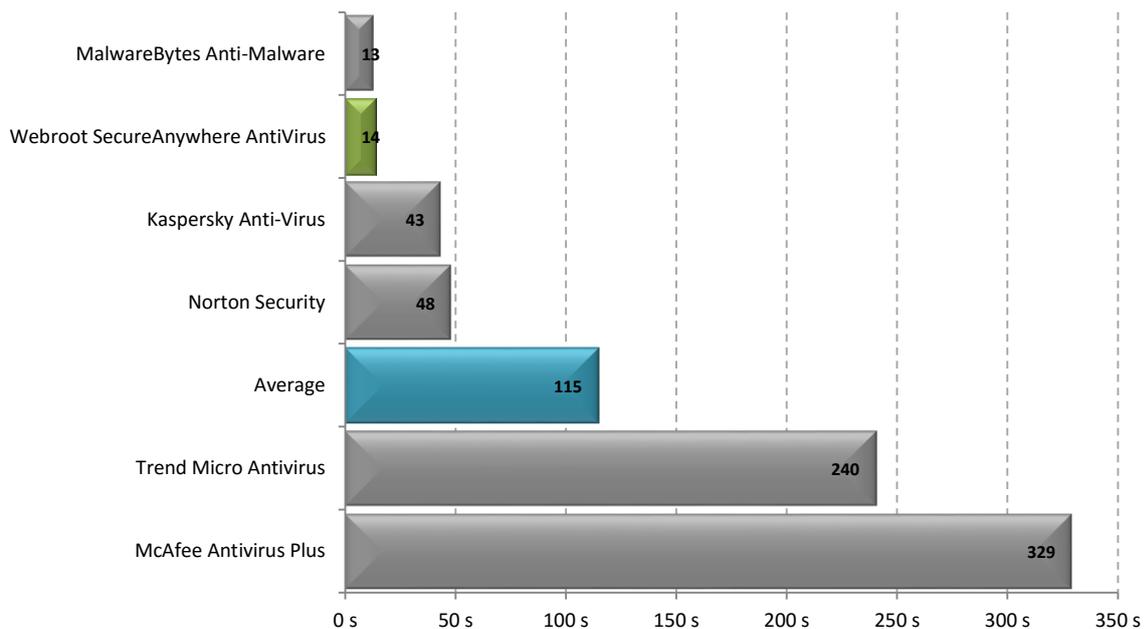
### Benchmark 3 – Installation Size

The following chart compares the total size of files added during the installation of Antivirus products. Products with lower installation sizes are considered better performing products in this category.



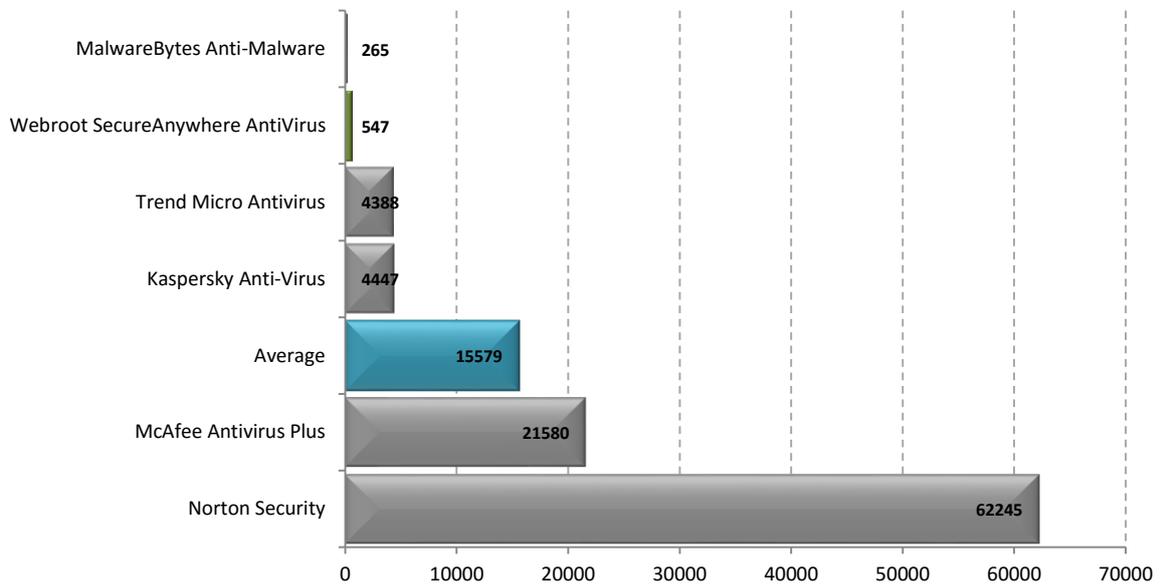
### Benchmark 4 – Installation Time

The following chart compares the minimum installation time it takes for Antivirus products to be fully functional and ready for use by the end user. Products with lower installation times are considered better performing products in this category.



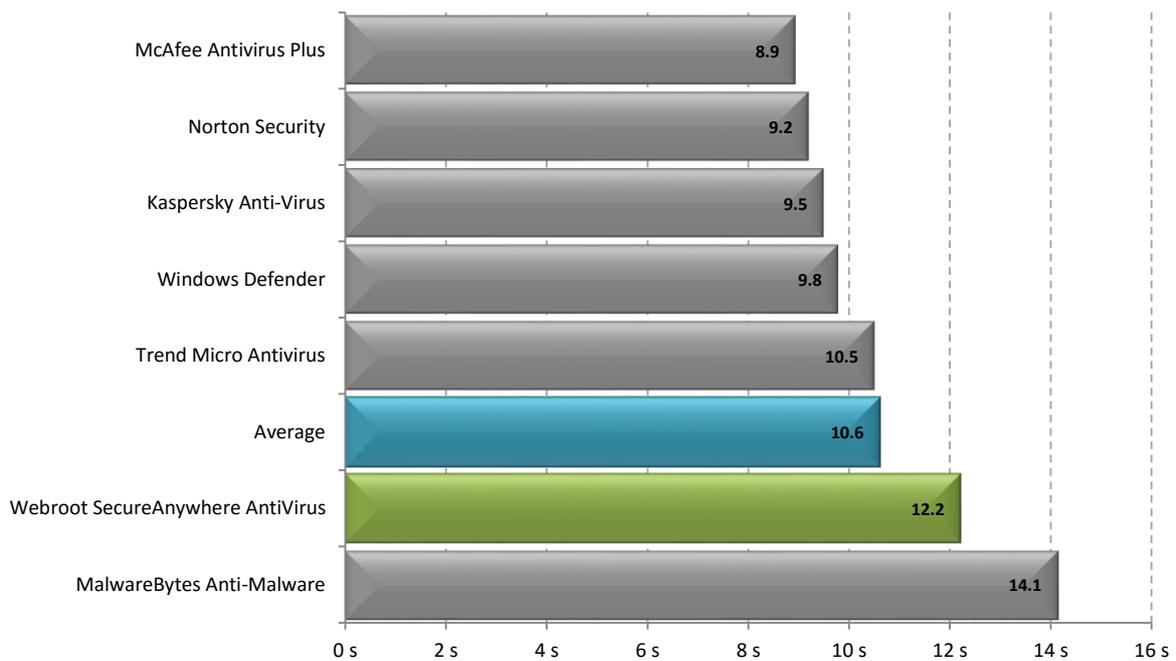
### Benchmark 5 – Registry Keys Added

The following chart compares the amount of Registry Keys created during product installation for each Antivirus product tested. Products with lower key counts are considered better performing products in this category.



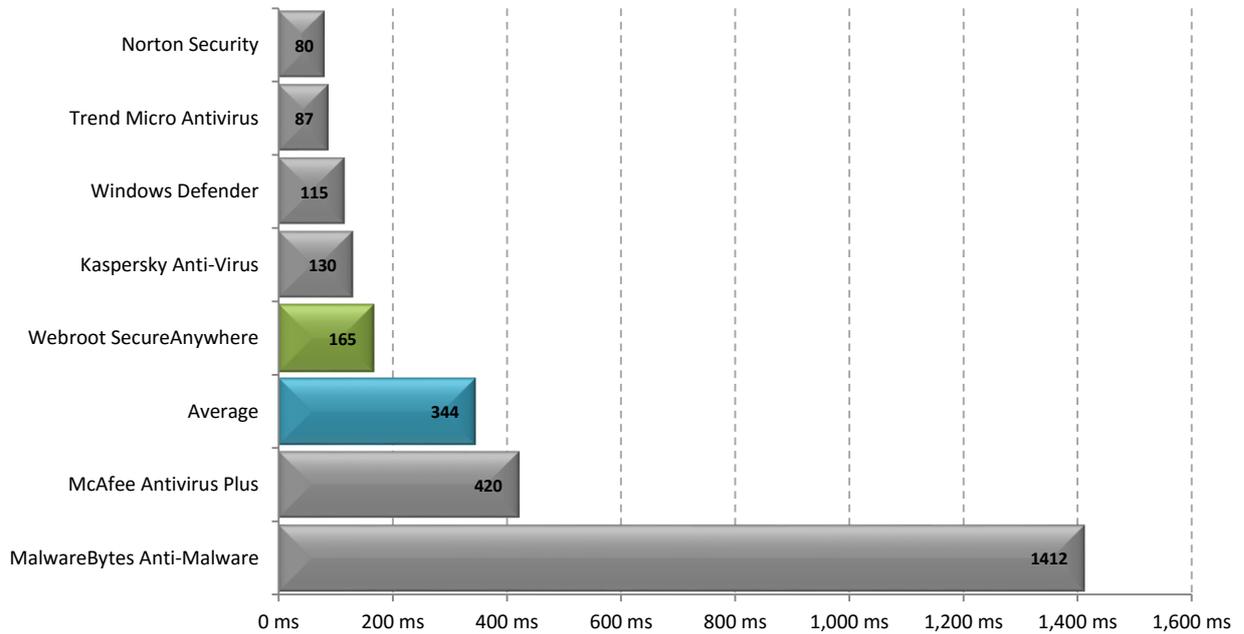
### Benchmark 6 – Boot Time

The following chart compares the average time taken for the system to boot (from a sample of five boots) for each Antivirus product tested. Products with lower boot times are considered better performing products in this category.



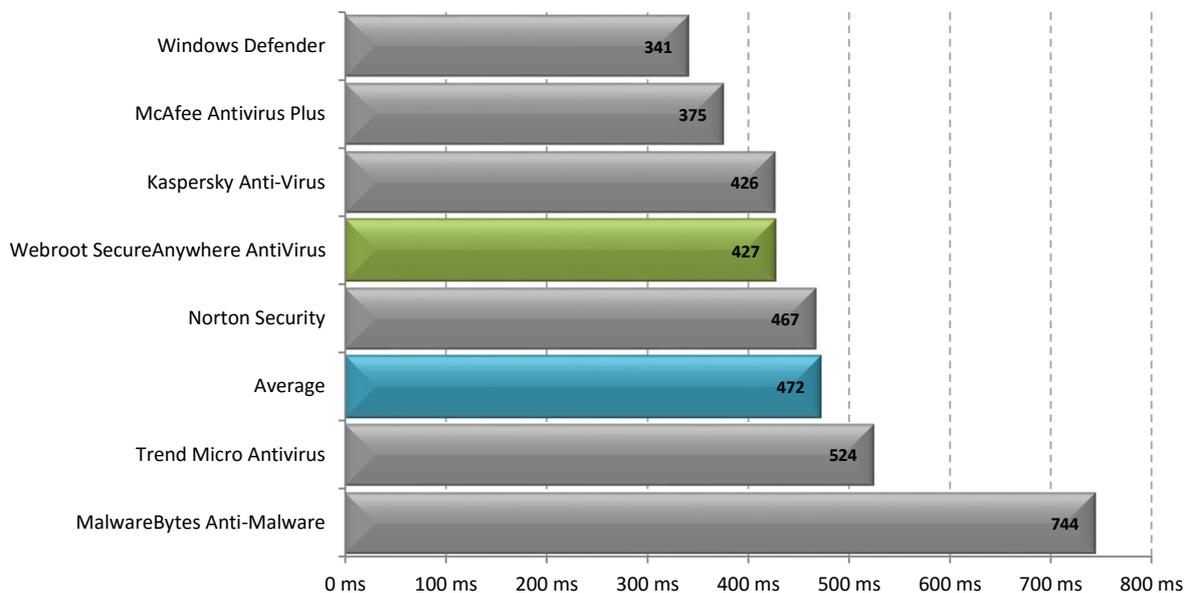
### Benchmark 7 – User Interface Launch Time

The following chart compares the average time taken to launch a product’s user interface. Products with lower launch times are considered better performing products in this category.



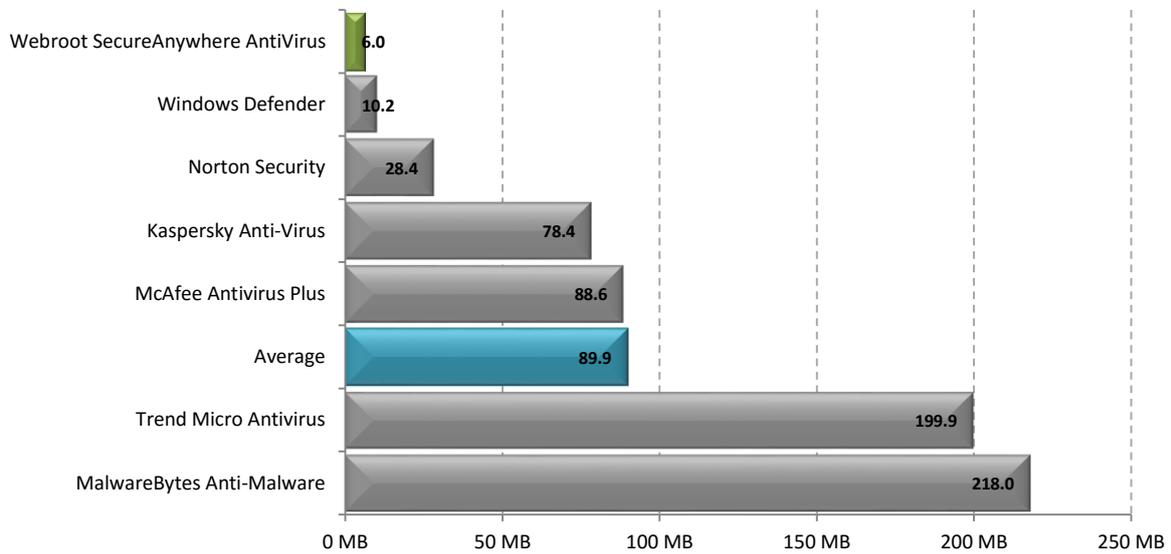
### Benchmark 8 – Internet Explorer Launch Time

The following chart compares the average launch times of Internet Explorer after rebooting the machine for each Antivirus product we tested. Products with lower launch times are considered better performing products in this category.



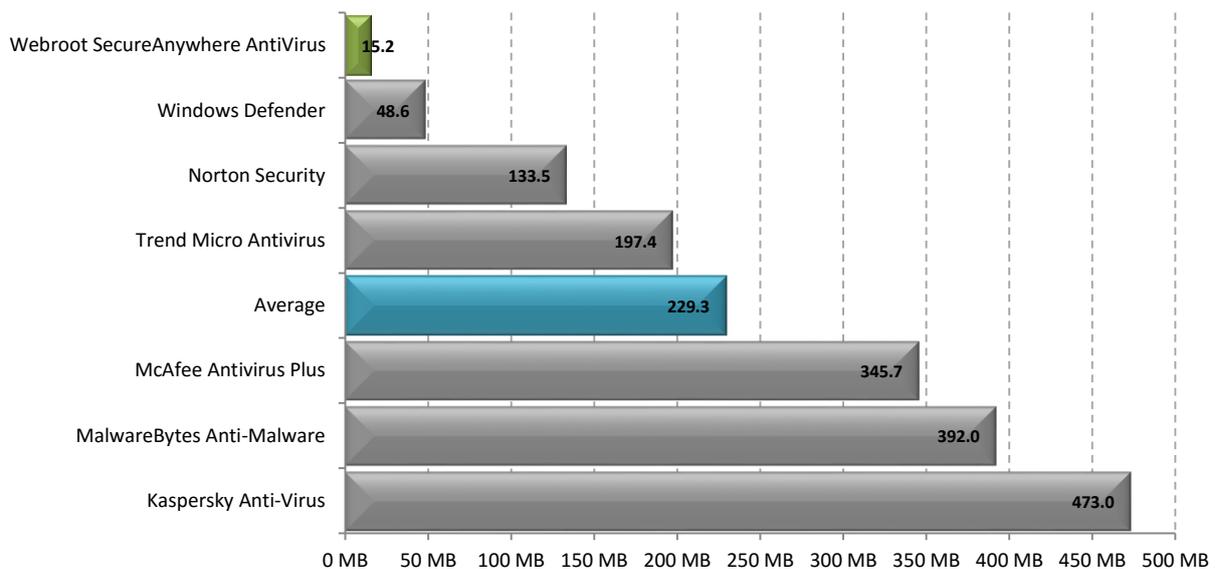
### Benchmark 9 – Memory Usage during System Idle

The following chart compares the average amount of RAM in use by an Antivirus product during a period of system idle. This average is taken from a sample of ten memory snapshots taken at roughly 60 seconds apart after reboot. Products with lower idle RAM usage are considered better performing products in this category.



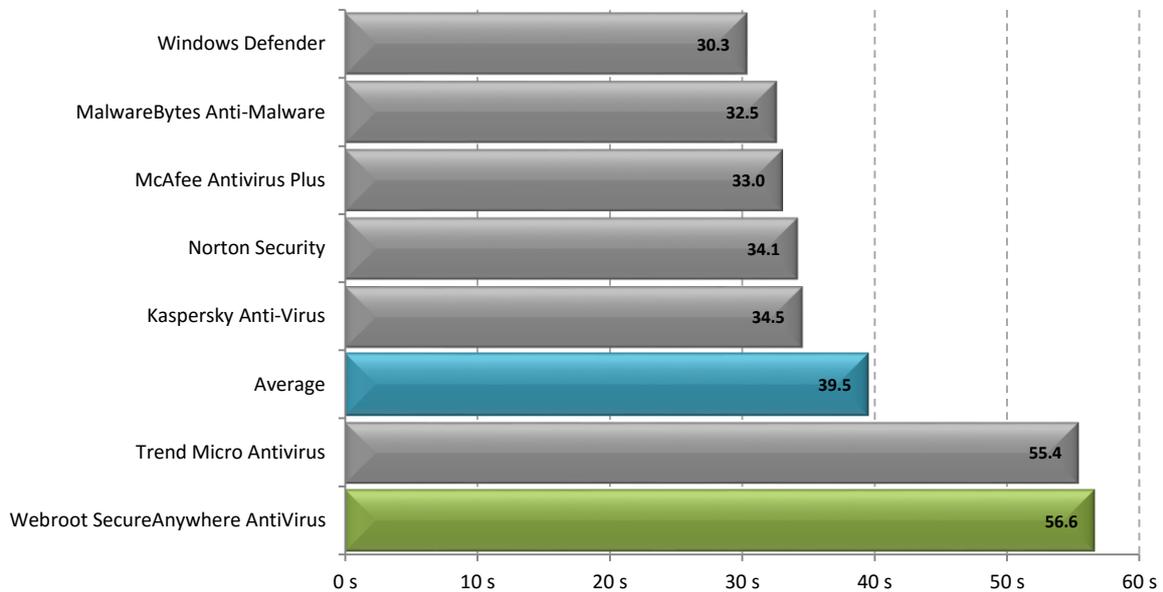
### Benchmark 10 – Memory Usage during Initial Scan

The following chart compares the average amount of RAM in use by an Antivirus product during an initial scan on the main drive. This average is taken from a sample of ten memory snapshots taken at five second intervals during a scan of sample files which have not been previously scanned by the software. Products that use less memory during a scan are considered better performing products in this category.



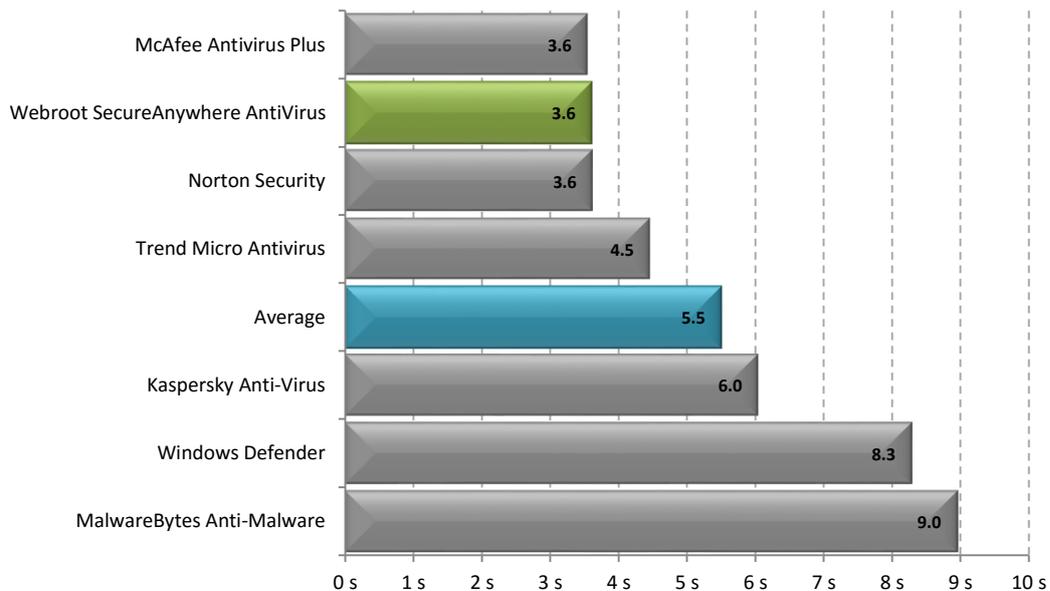
## Benchmark 11 – Browse Time

The following chart compares the average time taken for Internet Explorer to successively load a set of popular websites through the local area network from a local server machine. Products with lower browse times are considered better performing products in this category.



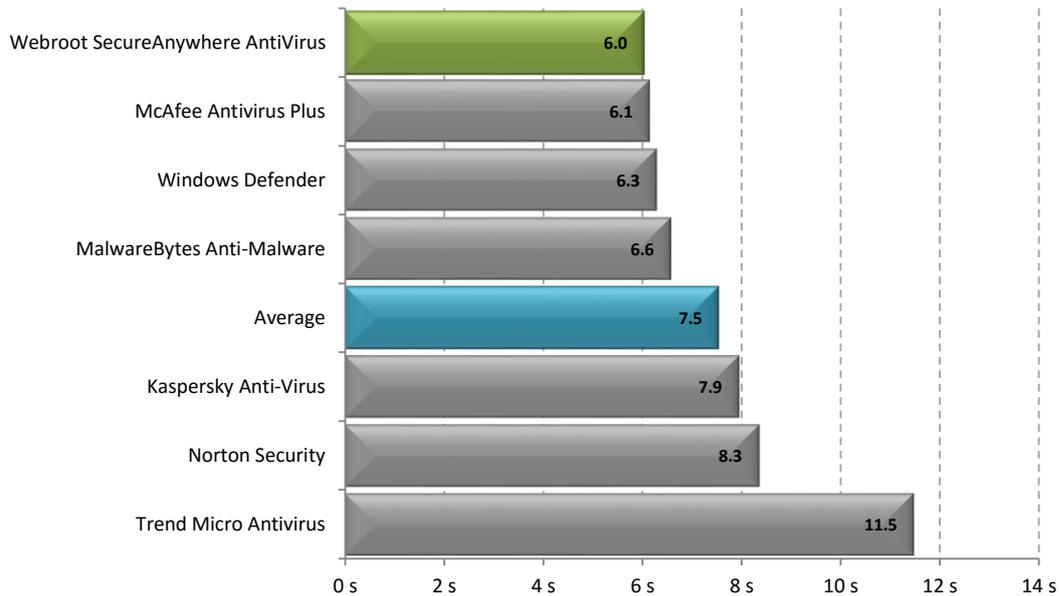
## Benchmark 12 – File Copy, Move and Delete

The following chart compares the average time taken to copy, move and delete several sets of sample files for each Antivirus product tested. Products with lower times are considered better performing products in this category.



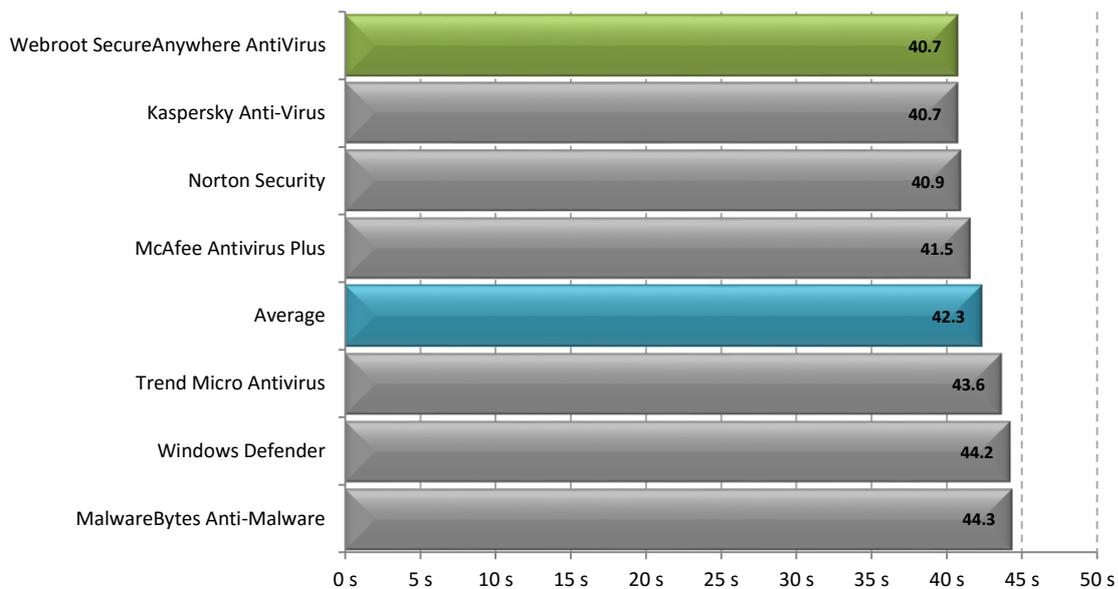
## Benchmark 13 – Network Throughput

The following chart compares the average time to download a sample set of common file types for each Antivirus product tested. Products with lower times are considered better performing products in this category.



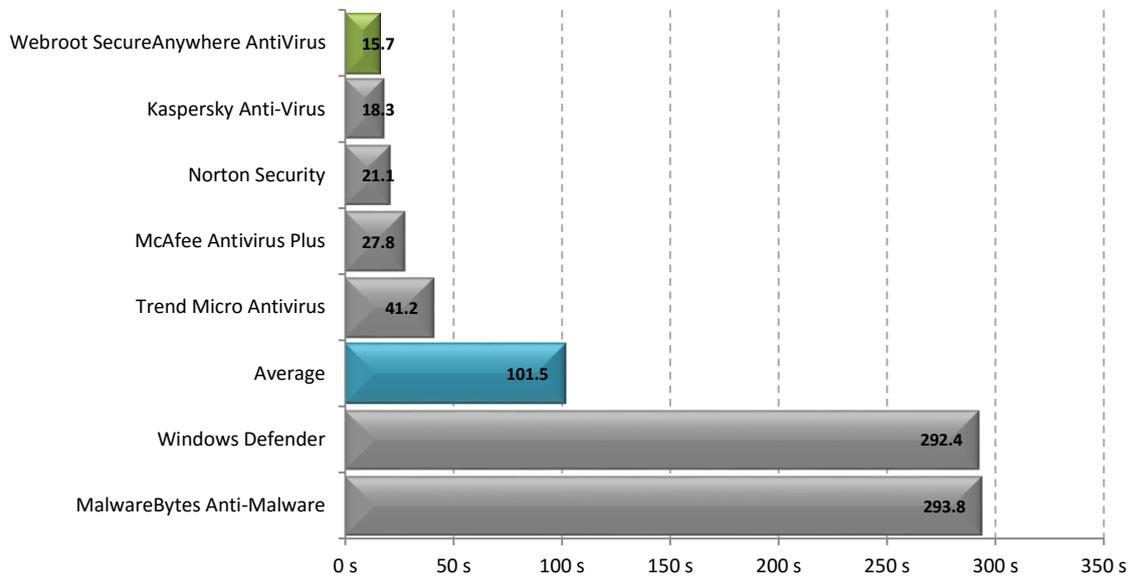
## Benchmark 14 – File Compression and Decompression

The following chart compares the average time it takes for sample files to be compressed and decompressed for each Antivirus product tested. Products with lower times are considered better performing products in this category.



## Benchmark 15 – File Write, Open and Close

The following chart compares the average time it takes for a file to be written to the hard drive then opened and closed 180,000 times, for each Antivirus product tested. Products with lower times are considered better performing products in this category.



## Disclaimer and Disclosure

This report only covers versions of products that were available at the time of testing. The tested versions are as noted in the “Products and Versions” section of this report. The products we have tested are not an exhaustive list of all products available in these very competitive product categories.

## Disclaimer of Liability

While every effort has been made to ensure that the information presented in this report is accurate, PassMark Software Pty Ltd assumes no responsibility for errors, omissions, or out-of-date information and shall not be liable in any manner whatsoever for direct, indirect, incidental, consequential, or punitive damages resulting from the availability of, use of, access of, or inability to use this information.

## Disclosure

Webroot Software Inc. funded the production of this report. The list of products tested and the metrics included in the report were selected by Webroot.

## Trademarks

All trademarks are the property of their respective owners.

# Contact Details

### PassMark Software Pty Ltd

Level 5

63 Foveaux St.

Surry Hills, 2010

Sydney, Australia

**Phone** + 61 (2) 9690 0444

**Fax** + 61 (2) 9690 0445

**Web** [www.passmark.com](http://www.passmark.com)

# Appendix 1 – Test Environment

For our testing, PassMark Software used a test environment running Windows 10 (64-bit) with the following hardware specifications:

## Windows 10 Home (64-bit) System

<b>Model:</b>	Lenovo H50W-50 i5
<b>CPU:</b>	Intel Core i5-4460 CPU @ 3.20GHz 3.20 GHz
<b>Video Card:</b>	NVIDIA GeForce GT 705
<b>RAM:</b>	8GB DDR3 RAM
<b>SSD (Main Boot Drive):</b>	Intel SSD 730 Series 240GB
<b>Network:</b>	Gigabit (1GB/s) switch
<b>O/S:</b>	Windows 10 Home 10.0 (Build 10240)

## Appendix 2 – Methodology Description

### Windows 10 Image Creation

*OSForensics* was used to create a “clean” baseline image prior to testing. Our aim was to create a baseline image with the smallest possible footprint and reduce the possibility of variation caused by external operating system factors.

The baseline image was restored prior to testing of each different product. This process ensures that we install and test all products on the same, “clean” machine.

The steps taken to create the base Windows 10 image are as follows:

1. Installation and activation of **Windows 10**.
2. Disabled Automatic Updates.
3. Changed User Account Control settings to “Never Notify”.
4. Disable Windows Defender automatic scans to avoid unexpected background activity.
5. Disable the Windows firewall to avoid interference with security software.
6. Disabled *Superfetch* to ensure consistent results.
7. Installed *HTTP Watch* for Browse Time testing.
8. Installed *Windows 10 ADK* for Boot Time testing.
9. Installed Active Perl for interpretation of some test scripts.
10. Install OSForensics for testing (Installation Size and Registry Key Count tests) purposes.
11. Disabled updates, accelerators and compatibility view updates in Internet Explorer 11.
12. Run a Windows Update.
13. Created a baseline image using OSForensics.

### Benchmark 1 – Scan Time

This benchmark measured the time it took for each product to scan a set of sample files. The sample used was identical in all cases and contained a mixture of system files and Office files. In total there were 6159 files whose combined size was 982 MB. Most of these files come from the Windows system folders. As the file types can influence scanning speed, the breakdown of the main file types, file numbers and total sizes of the files in the sample set is given here:

File Extension	Number of Files	File Size
.dll	2589	490MB
.exe	695	102MB
.sys	332	23MB
.gif	302	1MB
.doc	281	64MB
.wmf	185	2MB
.png	149	2MB
.html	126	1MB
.nls	80	6MB
.jpg	70	1MB
.ini	59	2MB
.ico	58	<1MB
.mof	43	6MB

.ax	39	4MB
.xls	38	3MB
.ime	35	5MB
.drv	31	1MB
.txt	31	1MB
.chm	30	6MB
.cpl	29	4MB
.mfl	29	3MB
.inf	26	2MB
.hlp	22	3MB
.imd	20	18MB
.py	20	<1MB
.msc	18	1MB
.vbs	18	1MB
.xml	18	1MB
.rtf	16	62MB
.ocx	16	4MB
.tsp	14	1MB
.com	14	<1MB
.xsl	14	<1MB
.h	13	<1MB
.vsd	12	2MB
.scr	12	2MB
.aw	12	2MB
.js	12	1MB
.zip	11	25MB
.lex	9	10MB
.ppt	9	4MB
.acm	9	1MB
.wav	7	5MB
Total	6159	982

This scan was run by right-clicking the test folder and choosing the “Scan Now” option. To record the scan time, we have used product’s built-in scan timer or reporting system. Where this was not possible, scan times were taken manually with a stopwatch.

For each product, five samples were taken with the machine rebooted before each sample to clear any caching effects by the operating systems. In the past, many products have shown a substantial difference between the initial scan time (first scan) and subsequent scan times (scans 2 to 5). We believe this behavior is due to products themselves caching recently scanned files. As a result of this mechanism, we have averaged the four subsequent scan times to obtain an average subsequent scan time. Our final result for this test is an average of the subsequent scan average and the initial scan time.

## Benchmark 2 – Scheduled Scan Time

This test measures the average time required by the software to complete a scheduled scan on the system. The default scheduled scan settings are kept (except for the start time) and the scan is scheduled to run at the next convenient time. The scan is run three times with a reboot between each run to remove potential caching effects. The result is calculated as a weighted average with a 50% weighting on the initial scan and a 50% weighting on the subsequent scans. Where this option is not available or the scheduling appears to be unreliable, the product is omitted from the metric, and given the lowest score for this metric.

### Benchmark 3 – Installation Size

A product's Installation Size was previously defined as the difference between the initial snapshot of the Disk Space (C: drive) before installation and the subsequent snapshot taken after the product is installed on the system. Although this is a widely used methodology, we noticed that the results it yielded were not always reproducible in Vista due to random OS operations that may take place between the two snapshots. We improved the Installation Size methodology by removing as many Operating System and disk space variables as possible.

Using PassMark's **OSForensics** we created initial and post-installation disk signatures for each product. These disk signatures recorded the amount of files and directories, and complete details of all files on that drive (including file name, file size, checksum, etc) at the time the signature was taken.

The initial disk signature was taken immediately prior to installation of the product. A subsequent disk signature was taken immediately following a system reboot after product installation. Using **OSForensics**, we compared the two signatures and calculated the total disk space consumed by files that were new, modified, and deleted during product installation. Our result for this metric reflects the total size of all newly added files during installation.

The scope of this metric includes only an 'out of the box' installation size for each product. Our result does not cover the size of files downloaded by the product after its installation (such as engine or signature updates), or any files created by system restore points, pre-fetch files and other temporary files.

### Benchmark 4 – Registry Key Count

This test measures the amount of keys and values added to registry, after rebooting the test machine following a successful product installation. The test was conducted using PassMark's **OSForensics**, to count the number of keys, errors and values that were added under HKEY\_LOCAL\_MACHINE and HKEY\_USERS. The Create Signature feature is used to take a before and after signature of these folders, and the signatures are then compared so that the new keys can be identified.

### Benchmark 5 – Boot Time

PassMark Software uses tools available from the *Windows Performance Toolkit* (as part of the Microsoft Windows 10 ADK obtainable from the [Microsoft Website](#)).

The Boot Performance (fast startup) test is ran as an individual assessment via the Windows Assessment Console. The network connection is disabled and the login password is removed to avoid interruption to the test. The final result is taken as the total boot duration excluding BIOS load time.

### Benchmark 6 – User Interface Launch Time

The launch time of a product's user interface was taken using *AppTimer (v1.0.1006)*. For each product tested, we obtained a total of fifteen samples from five sets of three UI launches, with a reboot before each set to clear caching effects by the operating system. When compiling the results the first of each set was separated out so that there was a set of values for the initial launch after reboot and a set for subsequent launches.

We have averaged the subsequent launch times to obtain an average subsequent launch time. Our final result for this test is an average of the subsequent launch average and the initial launch time.

In some cases, *AppTimer* did not correctly record the time taken for UI launch. For instance, some applications would open their window and look like they were ready, but then continued to be unresponsive. Where the measurement from *AppTimer* appeared inaccurate, we have taken the time manually with a stop watch.

*AppTimer* is publically available from the [PassMark Website](#).

## Benchmark 7 – Internet Explorer Launch Time

The average launch time of Internet Explorer interface was taken using *AppTimer*. This test was practically identical to the User Interface launch time test. For each product tested, we obtained a total of fifteen samples from five sets of three Internet Explorer launches, with a reboot before each set to clear caching effects by the operating system. When compiling the results the first of each set was separated out so that there was a set of values for the initial launch after reboot and a set for subsequent launches.

For this test, we have used *Internet Explorer 11 (11.63.10586.0)* as our test browser.

We have averaged the subsequent launch times to obtain an average subsequent launch time. Our final result for this test is an average of the subsequent launch average and the initial launch time.

*AppTimer* is publically available from the [PassMark Website](#).

## Benchmark 8 – Memory Usage during System Idle

The *Perflog++* utility was used to record process memory usage on the system at boot, and then every minute for another fifteen minutes after. This was done only once per product and resulted in a total of 15 samples. The first sample taken at boot is discarded.

The *PerfLog++* utility records memory usage of all processes, not just those of the anti-malware product. As a result of this, an anti-malware product's processes needed to be isolated from all other running system processes. To isolate relevant process, we used a program called *Process Explorer* which was run immediately upon the completion of memory usage logging by *PerfLog++*. *Process Explorer* is a Microsoft Windows Sysinternals software tool which shows a list of all DLL processes currently loaded on the system.

## Benchmark 9 – Memory Usage during Initial Scan

The *PerfLog++* utility was used to record memory usage on the system while a malware scan is in progress. Please refer to the metric "**Memory usage – System Idle**" above for a description of the *PerfLog++* utility and an explanation of the method by which memory usage is calculated.

As some products cache scan locations, we take reasonable precautions to ensure that the antivirus software does not scan the C:\ drive at any point before conducting this test. A manual scan on the C:\ drive is initiated at the same time as the *PerfLog++* utility, enabling *PerfLog++* to record memory usage for 60 seconds at five second intervals.

## Benchmark 10 – Browse Time

We used a script in conjunction with *HTTPWatch (Basic Edition, version 9.3)* to record the amount of time it takes for a set of 106 'popular' websites to load consecutively from a local server. This script feeds a list of URLs into *HTTPWatch*, which instructs the browser to load pages in sequence and monitors the amount of time it takes for the browser to load all items on one page.

For this test, we have used *Internet Explorer 11* as our browser.

The set of websites used in this test include front pages of high traffic pages. This includes shopping, social, news, finance and reference websites.

The Browse Time test is executed five times and our final result is an average of these five samples. The local server is restarted between different products and one initial 'test' run is conducted prior to testing to install *Adobe Flash Player*, an add-on which is used by many popular websites.

## Benchmark 11 – Installation Time

This test measures the minimum Installation Time a product requires to be fully functional and ready for use by the end user. Installation time can usually be divided in three major phases:

- The **Extraction and Setup phase** consists of file extraction, the EULA prompt, product activation and user configurable options for installation.
- The **File Copy phase** occurs when the product is being installed; usually this phase is indicated by a progress bar.
- The **Post-Installation phase** is any part of the installation that occurs after the File Copy phase. This phase varies widely between products; the time recorded in this phase may include a required reboot to finalize the installation or include the time the program takes to become idle in the system tray.

To reduce the impact of disk drive variables, each product was copied to the Desktop before initializing installation. Each step of the installation process was manually timed with a stopwatch and recorded in as much detail as possible. Where input was required by the end user, the stopwatch was paused and the input noted in the raw results in parenthesis after the phase description.

Where possible, all requests by products to pre-scan or post-install scan were declined or skipped. Where it was not possible to skip a scan, the time to scan was included as part of the installation time. Where an optional component of the installation formed a reasonable part of the functionality of the software, it was also installed (e.g. website link checking software as part of an Antivirus Product).

Installation time includes the time taken by the product installer to download components required in the installation. This may include mandatory updates or the delivery of the application itself from a download manager. We have noted in our results where a product has downloaded components for product installation.

We have excluded product activation times due to network variability in contacting vendor servers or time taken in account creation.

## Benchmarks 12– 16 - Real-Time Performance

We used a single script in testing Benchmarks 10-15. The script consecutively executes tests for Benchmarks 10-15. The script times each phase in these benchmarks using *CommandTimer.exe* and appends results to a log file.

## Benchmarks 12 – File Copy, Move and Delete

This test measures the amount of time required for the system to copy, move and delete samples of files in various file formats. This sample was made up of 812 files over 760,867,636 bytes and can be categorized as documents [26% of total], media files [54% of total] and PE files (i.e. System Files) [20% of total].

The breakdown of the main file types, file numbers and total sizes of the files in the sample set is shown in the following table:

File format	Number	Size (bytes)
DOC	8	30,450,176
DOCX	4	13,522,409
PPT	3	5,769,216
PPTX	3	4,146,421
XLS	4	2,660,352
XLSX	4	1,426,054
PDF	73	136,298,049
ZIP	4	6,295,987
7Z	1	92,238
JPG	351	31,375,259
GIF	6	148,182
MOV	7	57,360,371
RM	1	5,658,646
AVI	8	78,703,408
WMV	5	46,126,167
MP3	28	191,580,387
EXE	19	2,952,914
DLL	104	29,261,568
AX	1	18,432
CPL	2	2,109,440
CPX	2	4,384
DRV	10	154,864
ICO	1	107,620
MSC	1	41,587
NT	1	1,688
ROM	2	36,611
SCR	2	2,250,240
SYS	1	37,528,093
TLB	3	135,580
TSK	1	1,152

UCE	1	22,984
EXE	19	2,952,914
DLL	104	29,261,568
AX	1	18,432
CPL	2	2,109,440
CPX	2	4,384
DRV	10	154,864
ICO	1	107,620
MSC	1	41,587
NT	1	1,688
ROM	2	36,611
SCR	2	2,250,240
SYS	1	37,528,093
TLB	3	135,580
TSK	1	1,152
UCE	1	22,984
<b>Total</b>	<b>812</b>	<b>760,867,636</b>

This test was conducted five times to obtain the average time to copy, move and delete the sample files, with the test machine rebooted between each sample to remove potential caching effects.

### Benchmark 13 – Network Throughput

This benchmark measured how much time was required to download a sample set of binary files of various sizes and types over a 100MB/s network connection. The files were hosted on a server machine running Windows Server 2008 and IIS 7. *CommandTimer.exe* was used in conjunction with *GNU Wget* (version 1.10.1) to time and conduct the download test.

The complete sample set of files was made up of 553,638,694 bytes over 484 files and two file type categories: media files [74% of total] and documents [26% of total]. The breakdown of the file types, file numbers and total sizes of the files in the sample set is shown in the following table:

File format	Number	Size (bytes)
JPEG	343	30,668,312
GIF	9	360,349
PNG	5	494,780
MOV	7	57,360,371
RM	1	5,658,646
AVI	8	78,703,408
WMV	5	46,126,167
MP3	28	191,580,387

PDF	73	136,298,049
ZIP	4	6,295,987
7Z	1	92,238
<b>Total</b>	<b>484</b>	<b>553,638,694</b>

This test was conducted five times to obtain the average time to download this sample of files, with the test machine rebooted between each sample to remove potential caching effects.

## Benchmark 14 – File Compression and Decompression

This test measured the amount of time required to compress and decompress a sample set of files. For this test, we used a subset of the media and documents files used in the *File Copy, Move and Delete* benchmark. *CommandTimer.exe* recorded the amount of time required for *7zip.exe* to compress the files into a \*.zip and subsequently decompress the created \*.zip file.

This subset comprised 1,218 files over 783 MB. The breakdown of the file types, file numbers and total sizes of the files in the sample set is shown in the following table:

File Type	File Number	Total Size
.xls	13	9.23 MB
.xlsx	9	3.51 MB
.ppt	9	7.37 MB
.pptx	11	17.4 MB
.doc	17	35.9 MB
.docx	19	24.5 MB
.gif	177	1.10 MB
.jpg	737	66.2 MB
.png	159	48.9 MB
.mov	7	54.7 MB
.rm	1	5.39 MB
.avi	46	459 MB
.wma	11	48.6 MB
.avi	46	459 MB
.wma	11	48.6 MB
<b>Total</b>	<b>1218</b>	<b>783 MB</b>

This test was conducted five times to obtain the average file compression and decompression speed, with the test machine rebooted between each sample to remove potential caching effects.

## Benchmark 15 – File Write, Open and Close

This benchmark was derived from Oli Warner's File I/O test at <http://www.thepcspy.com> (please see *Reference #1: What Really Slows Windows Down*).

For this test, we developed *OpenClose.exe*, an application that looped writing a small file to disk, then opening and closing that file. *CommandTimer.exe* was used to time how long the process took to complete 180,000 cycles.

This test was conducted five times to obtain the average file writing, opening and closing speed, with the test machine rebooted between each sample to remove potential caching effects.