

# Consumer Antivirus Performance Benchmarks (2011)

Internet Security Products  
Windows 7

August 2010

**Document:** Consumer Antivirus Performance Benchmarks  
**Authors:** K. Lai, D. Wren  
**Company:** PassMark Software  
**Date:** 24 August 2010  
**Report:** 1  
**File:** antivirus\_11-performance-testing-ed1.docx

# 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>PRODUCTS AND VERSIONS</b> .....	<b>5</b>
<b>RANKING</b> .....	<b>6</b>
<b>PERFORMANCE METRICS SUMMARY</b> .....	<b>7</b>
<b>TEST RESULTS</b> .....	<b>10</b>
BENCHMARK 1 – BOOT TIME.....	10
BENCHMARK 2 – SCAN TIME.....	11
BENCHMARK 3 – USER INTERFACE LAUNCH TIME.....	11
BENCHMARK 4 – MEMORY USAGE DURING SYSTEM IDLE.....	12
BENCHMARK 5 – BROWSE TIME.....	12
BENCHMARK 6 – INTERNET EXPLORER LAUNCH TIME.....	13
BENCHMARK 7 – INSTALLATION TIME.....	13
BENCHMARK 8 – INSTALLATION SIZE.....	14
BENCHMARK 9 – REGISTRY KEYS ADDED.....	14
BENCHMARK 10 – FILE COPY, MOVE AND DELETE.....	15
BENCHMARK 11 – INSTALLATION OF THIRD PARTY APPLICATIONS.....	15
BENCHMARK 12 – NETWORK THROUGHPUT.....	16
BENCHMARK 13 – FILE FORMAT CONVERSION.....	16
BENCHMARK 14 – FILE COMPRESSION AND DECOMPRESSION.....	17
BENCHMARK 15 – FILE WRITE, OPEN AND CLOSE.....	17
<b>DISCLAIMER AND DISCLOSURE</b> .....	<b>18</b>
<b>CONTACT DETAILS</b> .....	<b>18</b>
<b>APPENDIX 1 – TEST ENVIRONMENT</b> .....	<b>19</b>
<b>APPENDIX 2 – METHODOLOGY DESCRIPTION</b> .....	<b>20</b>

## Revision History

Rev	Revision History	Date
Report 1	Initial version of this report, includes new results for Norton Internet Security 2011, G Data Internet Security 2011, Kaspersky Internet Security 2011, AVG Internet Security 9.0, McAfee Internet Security 2010.	24 August 2010

## References

Ref #	Document	Author	Date
1	What Really Slows Windows Down (URL)	O. Warner, The PC Spy	2001-2009

# Executive Summary

PassMark Software® conducted objective performance testing on five, publically available Internet Security software products on Windows 7 Ultimate Edition (64-bit) in July 2010. This report includes newly obtained results from these current software products as well as five previously obtained benchmark results from last generation software products released in 2010. Where applicable, we have tested new versions of software released for the year 2011.

Subsequent editions of this report will include new products released for 2011 as they are made available. Newer results will supersede previously obtained results for antivirus products released for 2010. For more details on which versions were tested, please see the section [\*"Products and Versions"\*](#).

Testing was performed on all products using fifteen performance metrics. These performance metrics are as follows:

- Boot Time;
- Scan Time;
- User Interface Launch Time;
- Memory Usage during System Idle;
- Browse Time;
- Internet Explorer Launch Time;
- Installation Size;
- Installation Time;
- Registry Keys Added;
- File Copy, Move and Delete;
- Installation of Third Party Applications;
- Network Throughput (previously named "Binary Download Test")
- File Format Conversion;
- File Compression and Decompression; and
- File Write, Open and Close.

All testing was performed on a Windows 7 Ultimate (64-bit) hardware platform.

## Products and Versions

Where possible, we have tested the full, retail release of the newest generation (2011 versions) of Internet Security software products. Where newer releases were not yet available, PassMark Software has either re-tested the latest version of the software, or used previously obtained results for the last generation (2010 versions) of Internet Security products.

PassMark Software intends to test new, major releases of Internet Software products as they become publically available. In future editions on this report, results for newer versions of software will supersede previously obtained results for older versions of software.

In this report, we have tested or included the following versions of Internet Security software:

Manufacturer	Product Name	Release Year	Product Version	Date Tested
Symantec Corporation	Norton Internet Security 2011	2011	Current 18.0.0.128	July 2010
G Data Software AG	G Data Internet Security 2011	2011	Current 21.0.2.1	July 2010
Kaspersky Labs	Kaspersky Internet Security 2011	2011	Current 11.0.0.239a	July 2010
AVG Technologies	AVG Internet Security 9.0	2010	Current 9.0.839	July 2010
McAfee, Inc	McAfee Internet Security 2010	2010	Current 10.5.194	July 2010
Trend Micro	Trend Micro Internet Security 2010	2010	Legacy 17.50.1366.0000	October 2009
ESET,LLC	ESET Smart Security 4	2010	Legacy 4.0.467.0	August 2009
F-Secure Corporation	F-Secure Internet Security 2010	2010	Legacy 10.00 build 246	October 2009
SourceNext Corporation	SourceNext Virus Security ZERO	2010	Legacy 9.16.210	October 2009
BitDefender	BitDefender Internet Security 2010	2010	Legacy 13.0.16.313	October 2009

# Ranking

This Overall Score provides a high level indication of that product's performance compared to other products we have tested. In this table, the highest possible score attainable is 150; in a hypothetical situation where a product has attained first place in every metric.

In the following table, Internet Security products have been ranked by their overall score:

Product Name	Overall Score
Norton Internet Security 2011	125
ESET Smart Security 4	102
AVG Internet Security 2010	98
Kaspersky Internet Security 2011	93
G Data Internet Security 2011	88
McAfee Internet Security 2010	78
SourceNext Virus Security Zero 2010	70
BitDefender Internet Security 2010	56
F-Secure Internet Security 2010	51
Trend Micro Internet Security 2010	44

# 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 – 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 2 – 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 1.2 GB 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 3 – 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 4 – 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 5 – 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 6 – 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 8. 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 7 – 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 8 – 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 9 – 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 10 – 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 11 – Installing Third Party Applications

This metric measures the amount of time taken to install and uninstall third party programs. The installation speed of third party applications may be impacted by antivirus behavior such as heuristics or real time malware scanning.

## Benchmark 12 – 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 13 – File Format Conversion**

This test measures the amount of time taken to convert an MP3 file to a WAV and subsequently, convert the same MP3 file to a WMA format.

### **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**

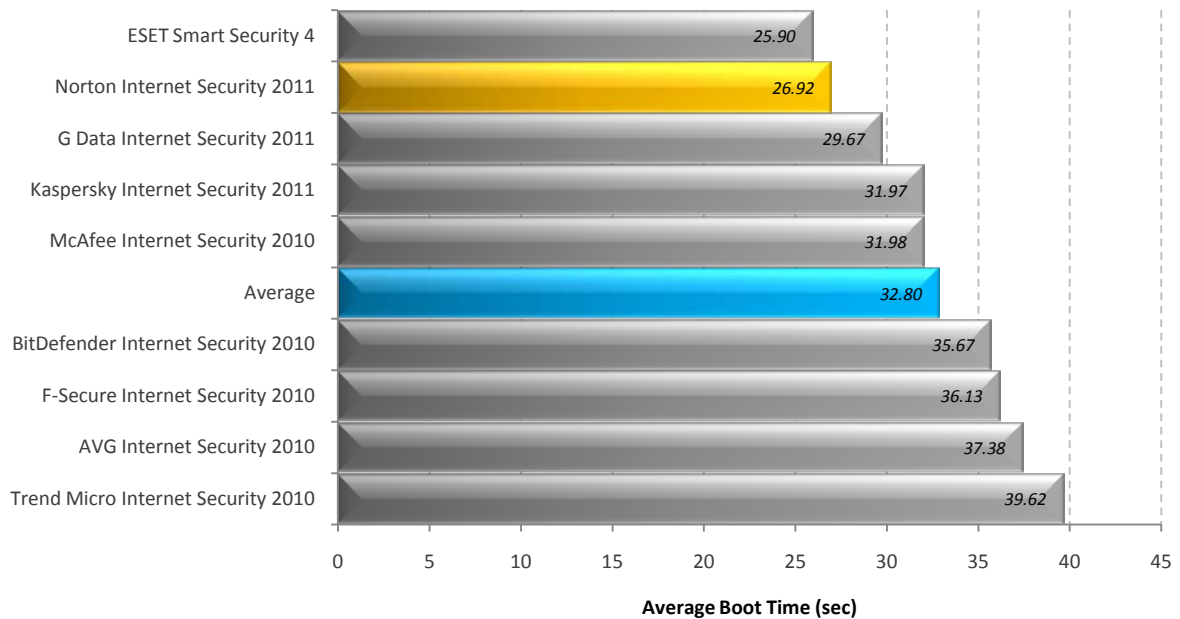
This benchmark was derived from Oli Warner's File I/O test at <http://www.thepcspy.com> (please see *Reference #1: What Really Slows Windows Down*). This metric measures the amount of time taken to write a file, then open and close that file.

# Test Results

In the following charts, we have highlighted the results we obtained for Norton Internet Security 2011 in yellow. The average has also been highlighted in blue for ease of comparison.

## Benchmark 1 – Boot Time

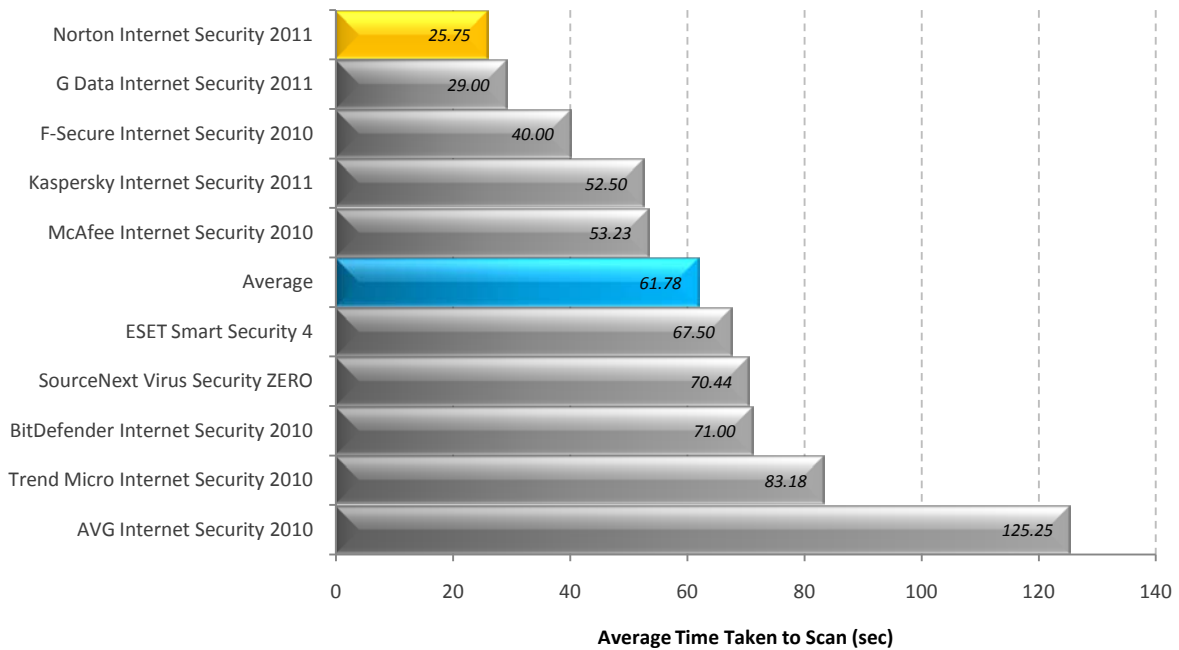
The following chart compares the average time taken for the system to boot (from a sample of five boots) for each Internet Security product tested. Products with lower boot times are considered better performing products in this category.



\* SourceNext Virus Security Zero did not successfully pass the boot time optimization process. As such, we were unable to obtain a comparable boot time result for this Internet Security product and SourceNext Virus Security Zero has been omitted from these results.

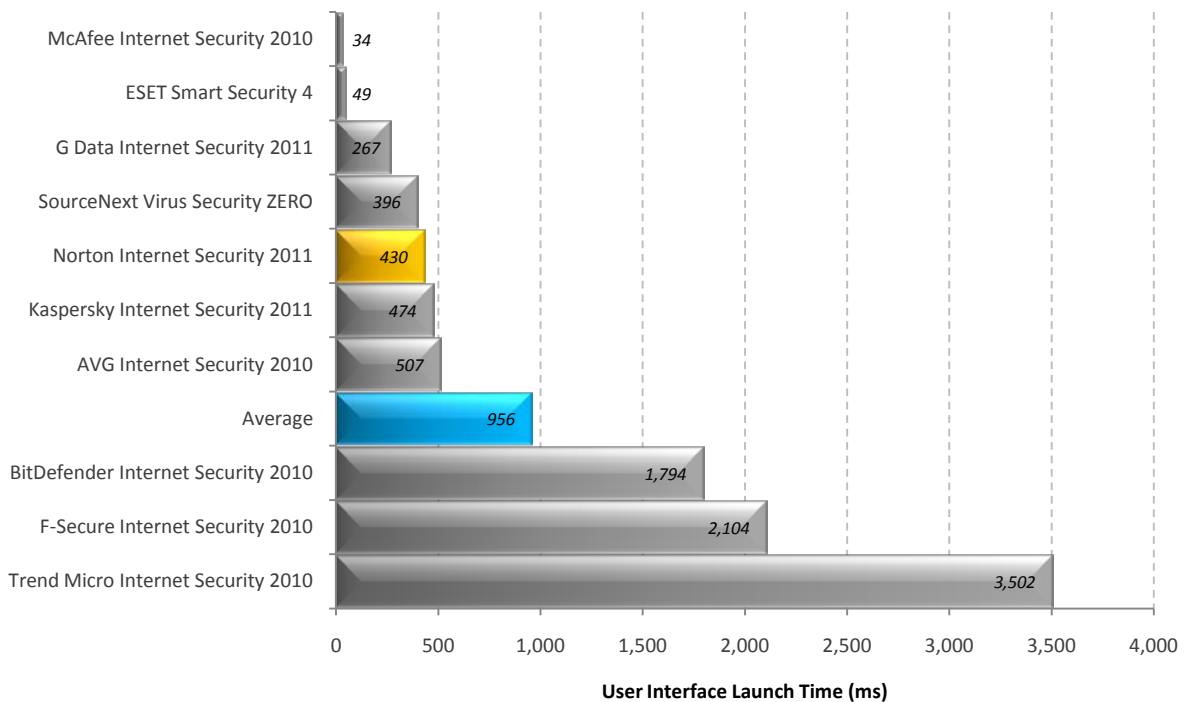
### Benchmark 2 – Scan Time

The following chart compares the average time taken to scan a set of 6159 files (totaling 982 MB) for each Internet Security product tested. This time is calculated by averaging the initial (Run 1) and subsequent (Runs 2-5) scan times. Products with lower scan times are considered better performing products in this category.



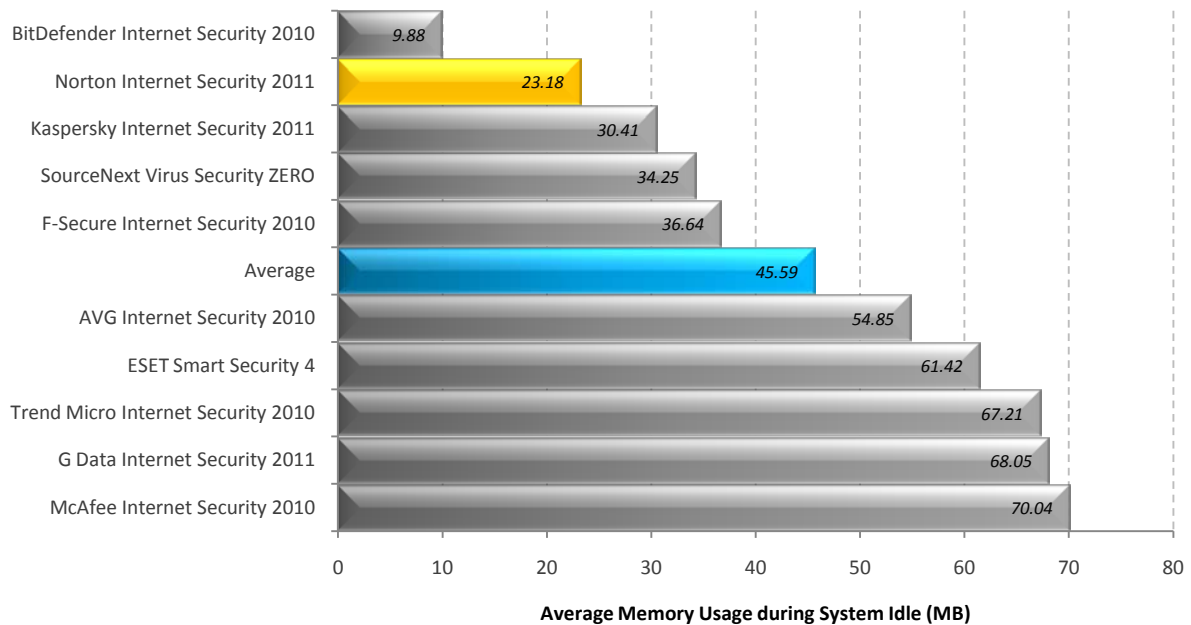
### Benchmark 3 – User Interface Launch Time

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.



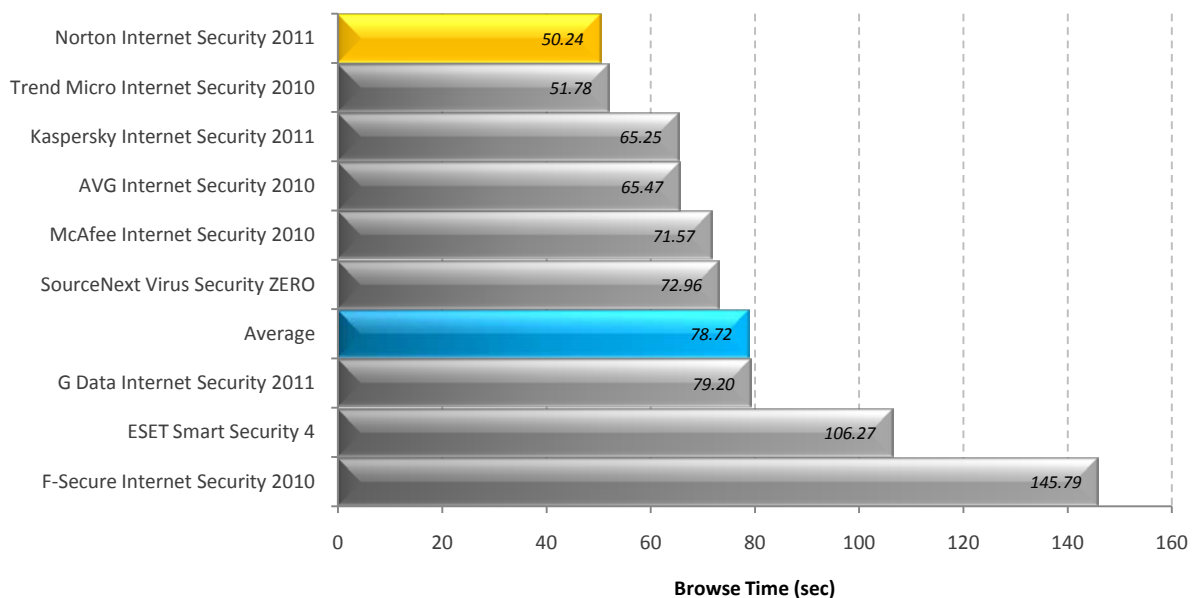
## Benchmark 4 – Memory Usage during System Idle

The following chart compares the average amount of RAM in use by an Internet Security 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 5 – Browse Time

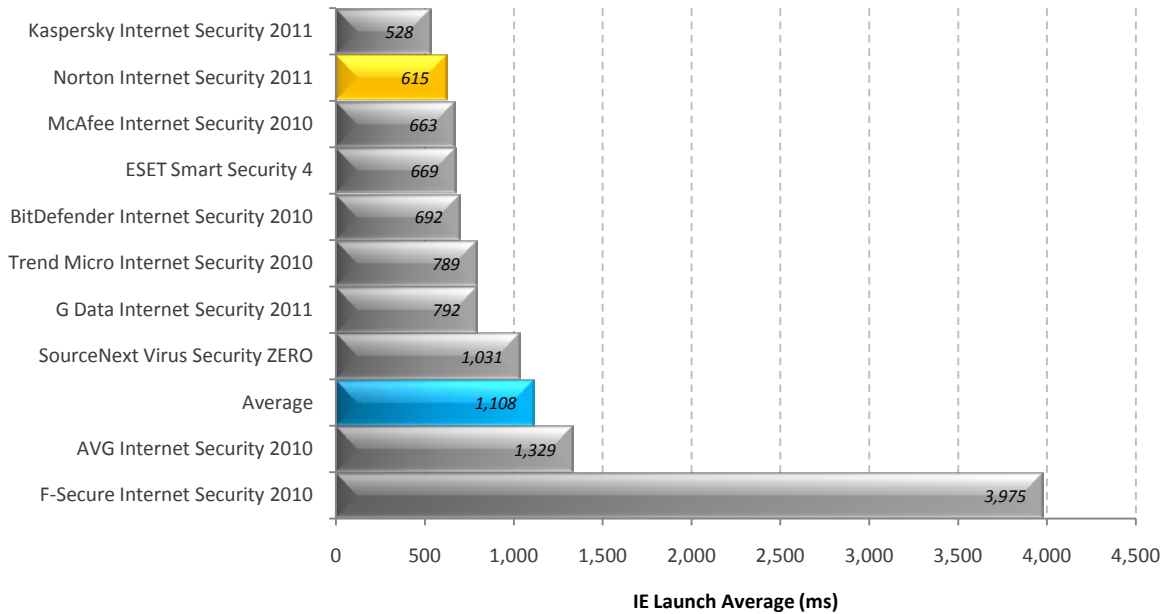
The following chart compares the average time taken for Internet Explorer to successively load a set of popular websites through the local area network from a local server machine. Products with lower browse times are considered better performing products in this category.



\* BitDefender IS 2010 did not successfully pass the Browse Time metric, and has been omitted from these results.

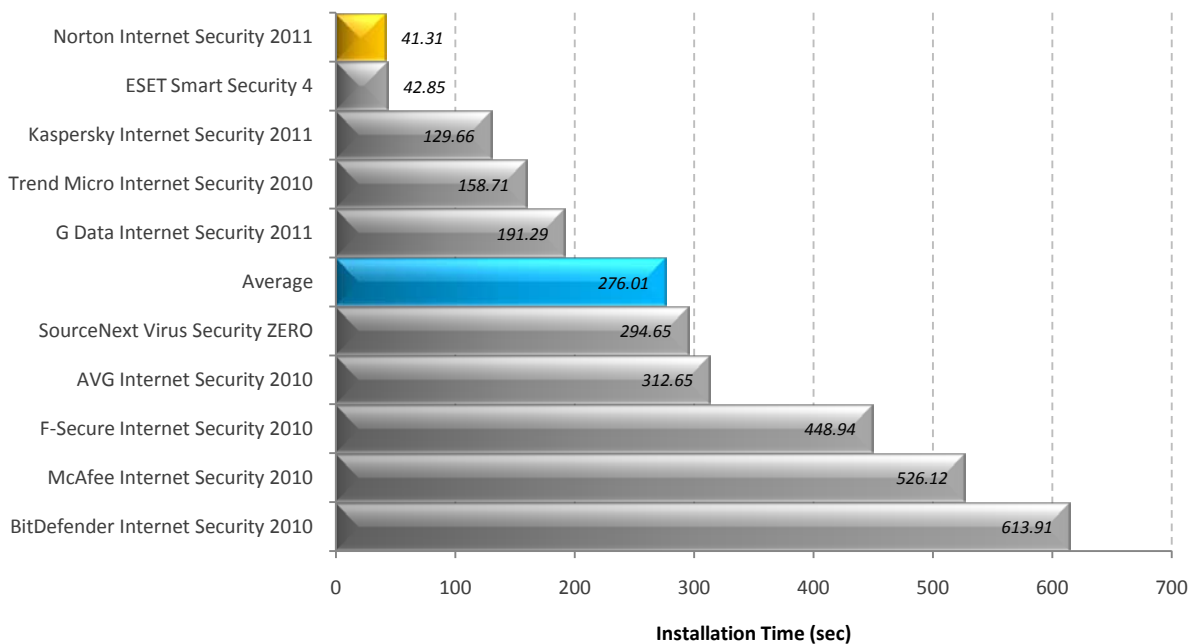
### Benchmark 6 – Internet Explorer Launch Time

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



### Benchmark 7 – Installation Time

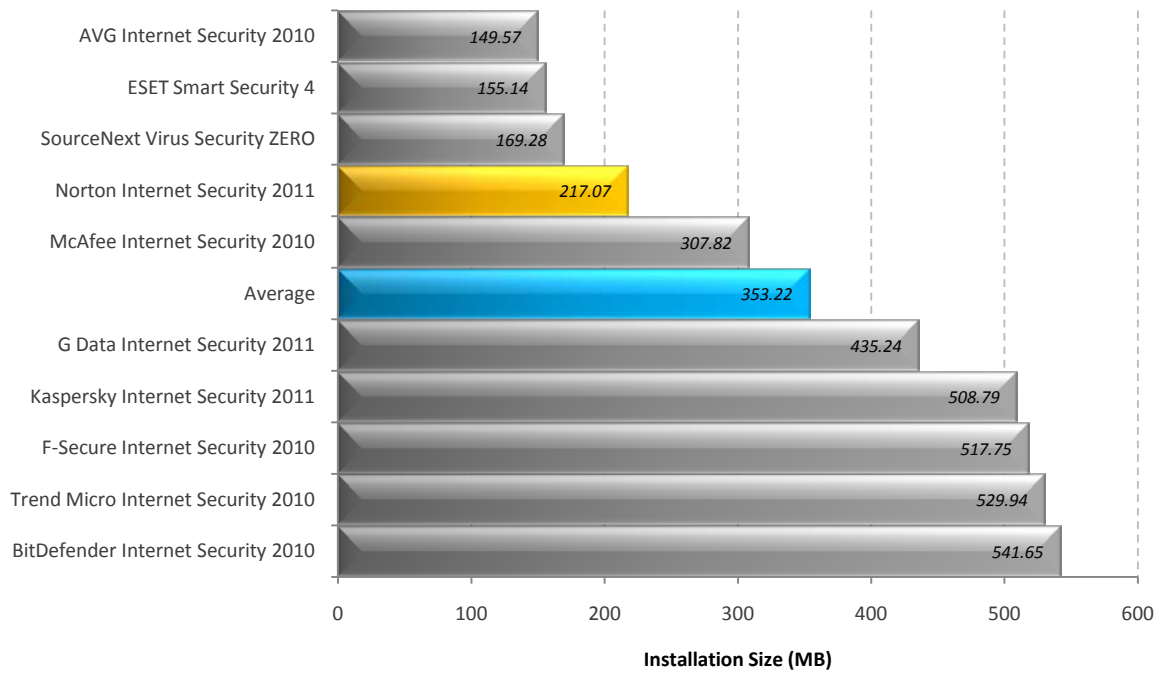
The following chart compares the minimum installation time it takes for Internet Security products to be fully functional and ready for use by the end user. Products with lower installation times are considered better performing products in this category.



\* Our result for **McAfee Internet Security 2010** includes time taken for the installer to download components as part of the installation process.

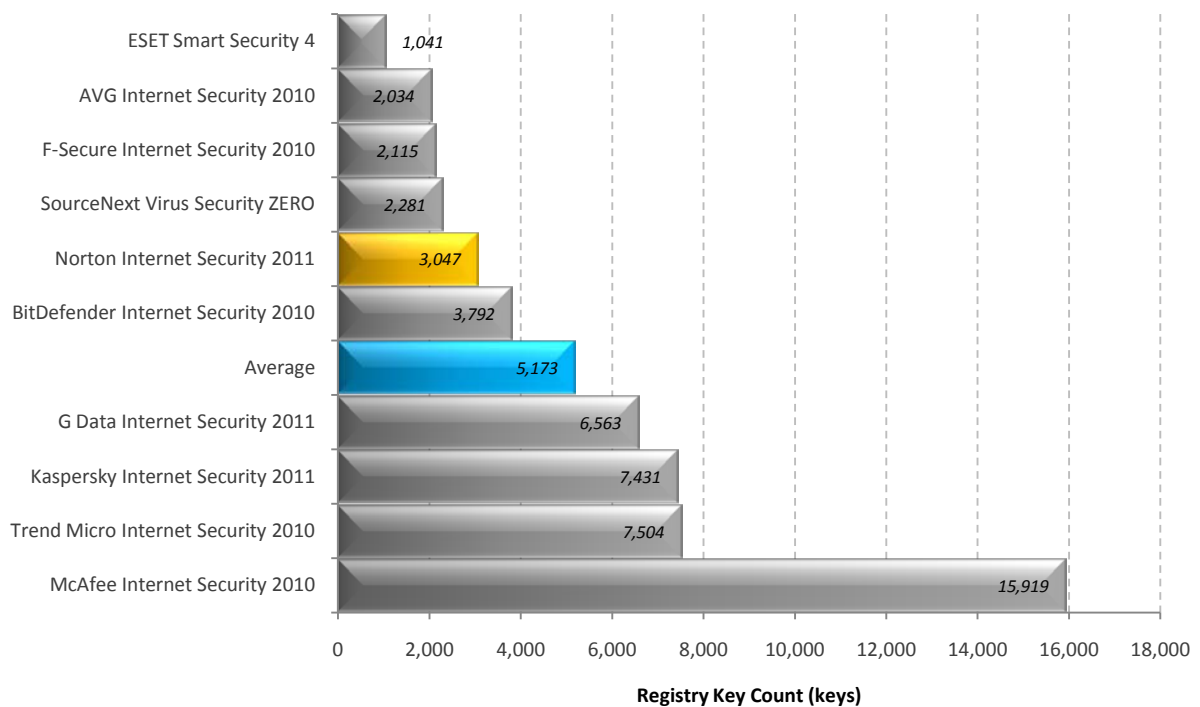
### Benchmark 8 – Installation Size

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



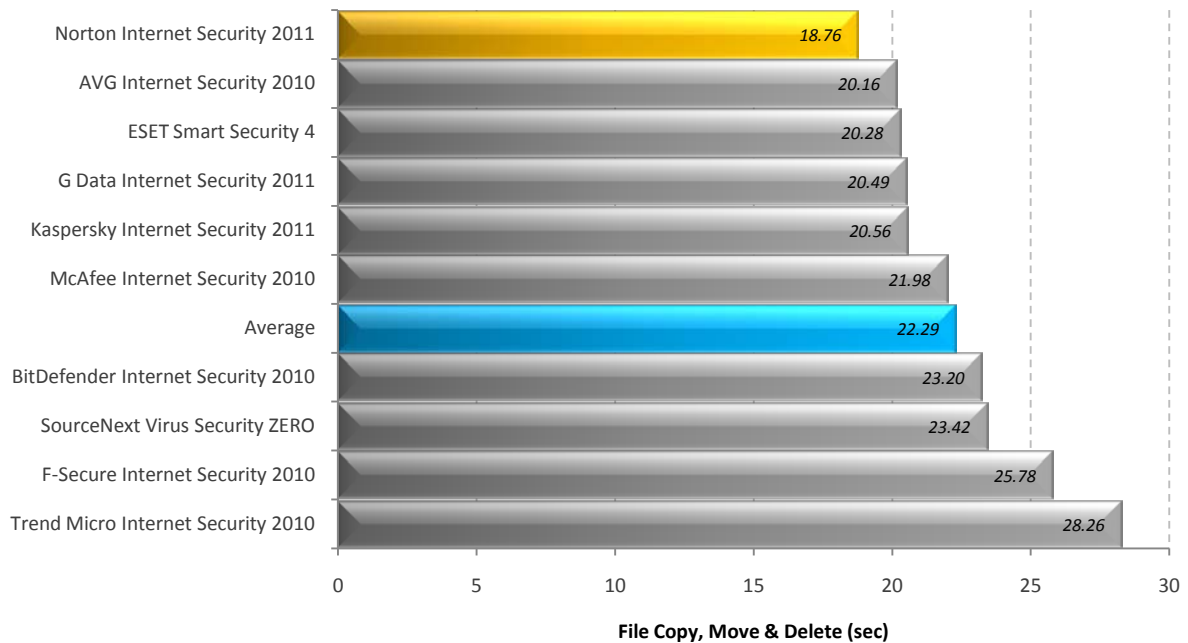
### Benchmark 9 – Registry Keys Added

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



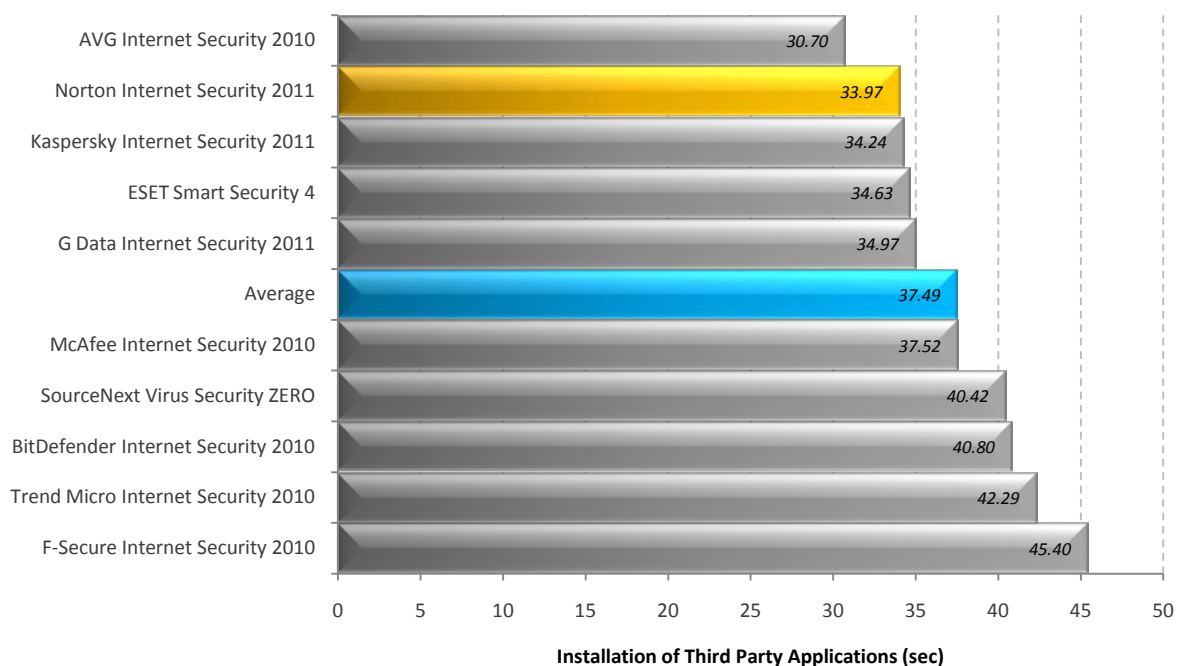
## Benchmark 10 – File Copy, Move and Delete

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



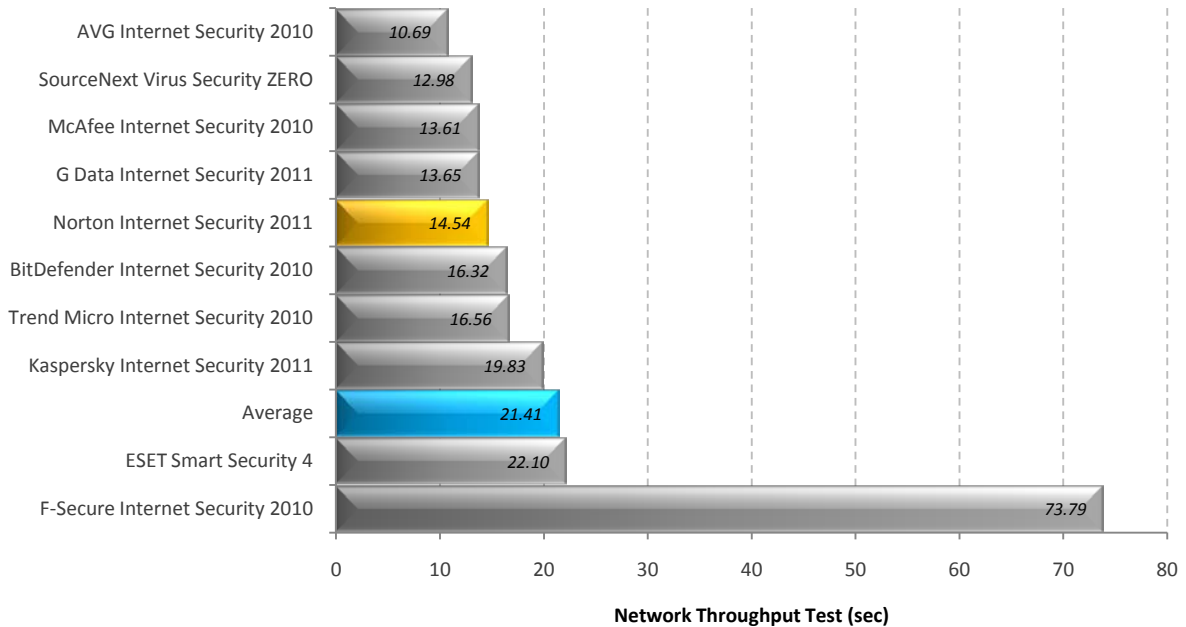
## Benchmark 11 – Installation of Third Party Applications

The following chart compares the average time taken to install a third party application for each Internet Security product tested. Products with lower times are considered better performing products in this category.



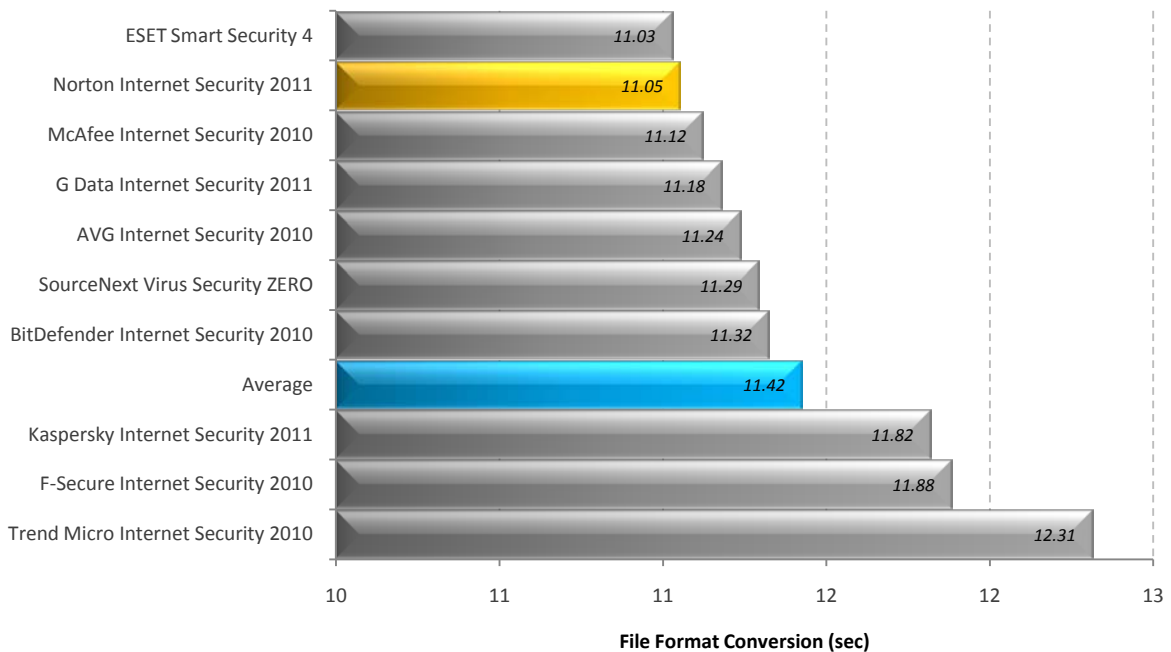
## Benchmark 12 – Network Throughput

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



## Benchmark 13 – File Format Conversion

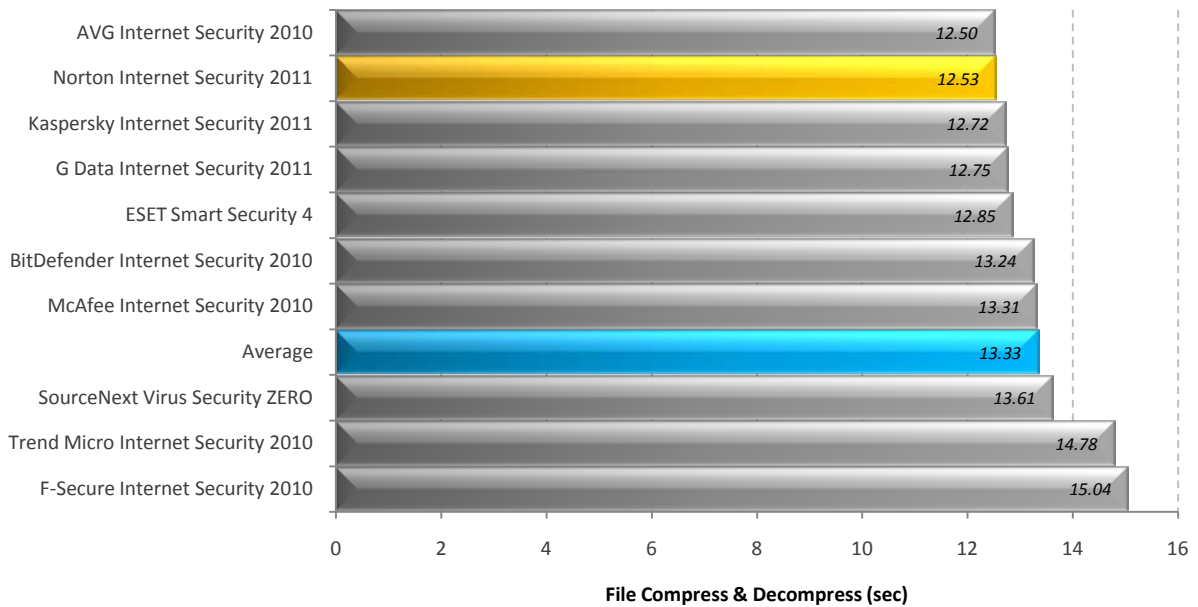
The following chart compares the average time it takes for a sample file to be converted from one file format to another (MP3 ↔ WMA, MP3 ↔ WAV) for each Internet Security product tested. Products with lower times are considered better performing products in this category.





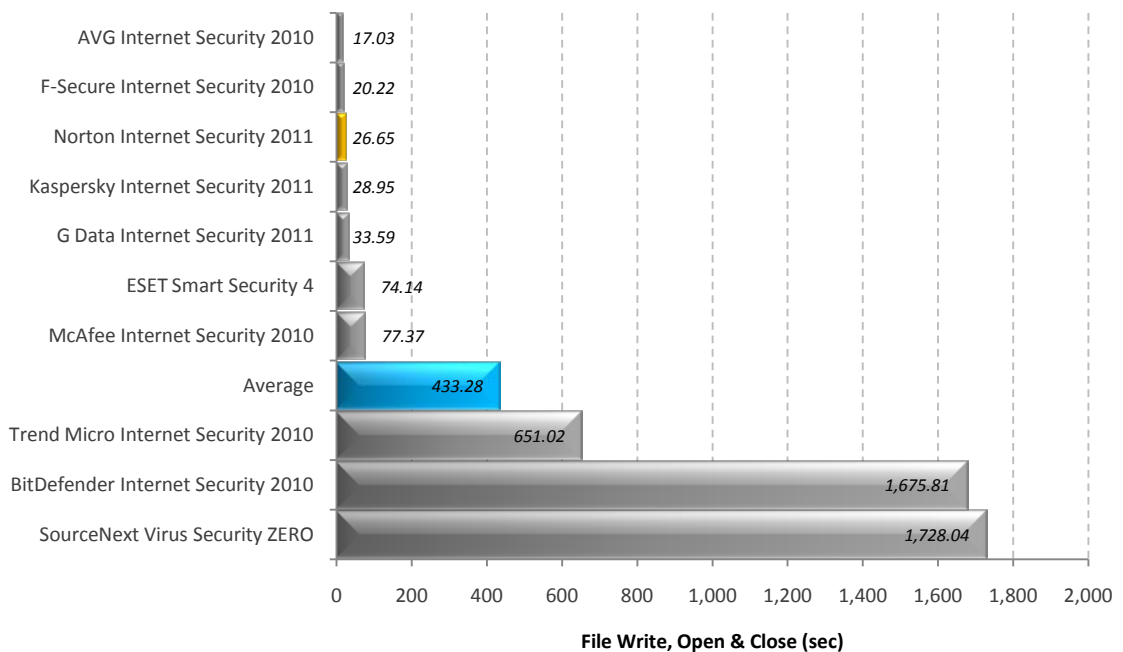
### Benchmark 14 – File Compression and Decompression

The following chart compares the average time it takes for sample files to be compressed and decompressed for each Internet Security 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 Internet Security 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

Symantec Corporation funded the production of this report and supplied some of the test scripts used for the tests.

## Trademarks

All trademarks are the property of their respective owners.

# Contact Details

## PassMark Software Pty Ltd

Suite 202, Level 2  
35 Buckingham St.  
Surry Hills, 2010  
Sydney, Australia

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

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

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

## Download Location

An electronic copy of this report can be found at the following location:

<http://www.passmark.com/avreport>

# Appendix 1 – Test Environment

For our testing, PassMark Software used a test environment running Windows 7 Ultimate (64-bit) with the following hardware specifications:

## Windows 7 (64-bit) System

<b>CPU:</b>	Intel Core i7 920 Quad Core @ 2.67GHz
<b>Video Card:</b>	nVidia GeForce 8800 GT
<b>Motherboard:</b>	Intel x58 Motherboard
<b>RAM:</b>	6GB DDR3 RAM
<b>HDD:</b>	Western Digital 500GB 7200RPM
<b>Network:</b>	Gigabit (1GB/s) switch

# Appendix 2 – Methodology Description

## Windows 7 Image Creation

As with testing on Windows Vista, *Norton Ghost* was used to create a “clean” baseline image prior to testing. Our aim is 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 7 image are as follows:

1. Installation and activation of **Windows 7 Ultimate** Edition.
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. Installed Norton Ghost for imaging purposes.
7. Disabled *Superfetch* to ensure consistent results.
8. Installed *HTTP Watch* for Browse Time testing.
9. Installed *Windows Performance Toolkit x64* for Boot Time testing.
10. Installed Active Perl for interpretation of some test scripts.
11. Disabled updates, accelerators and compatibility view updates in Internet Explorer 8.
12. Created a baseline image using Norton Ghost.

## Benchmark 1 – Boot Time

PassMark Software uses tools available from the **Windows Performance Toolkit version 4.6** (as part of the Microsoft Windows 7 SDK obtainable from the [Microsoft Website](#)) with a view to obtaining more precise and consistent boot time results on the Windows 7 platform.

The boot process is first optimized with **xbootmgr.exe** using the command “*xbootmgr.exe -trace boot –prepSystem*” which prepares the system for the test over six optimization boots. The boot traces obtained from the optimization process are discarded.

After boot optimization, the benchmark is conducted using the command “*xbootmgr.exe -trace boot -numruns 5*”. This command boots the system five times in succession, taking detailed boot traces for each boot cycle.

Finally, a post-processing tool was used to parse the boot traces and obtain the *BootTimeViaPostBoot* value. This value reflects the amount of time it takes the system to complete all (and only) boot time processes. Our final result is an average of five boot traces.

## Benchmark 2 – Scan Time

Scan Time is 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.

.dll	2589	490MB	.ime	35	5MB	.tsp	14	1MB
.exe	695	102MB	.drv	31	1MB	.com	14	<1MB
.sys	332	23MB	.txt	31	1MB	.xsl	14	<1MB
.gif	302	1MB	.chm	30	6MB	.h	13	<1MB
.doc	281	64MB	.cpl	29	4MB	.vsd	12	2MB
.wmf	185	2MB	.mfl	29	3MB	.scr	12	2MB
.png	149	2MB	.inf	26	2MB	.aw	12	2MB
.html	126	1MB	.hlp	22	3MB	.js	12	1MB
.nls	80	6MB	.imd	20	18MB	.zip	11	25MB
.jpg	70	1MB	.py	20	<1MB	.lex	9	10MB
.ini	59	2MB	.msc	18	1MB	.ppt	9	4MB
.ico	58	<1MB	.vbs	18	1MB	.acm	9	1MB
.mof	43	6MB	.xml	18	1MB	.wav	7	5MB
.ax	39	4MB	.rtf	16	62MB			
.xls	38	3MB	.ocx	16	4MB			

This scan was run without launching the product's user interface, 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.

Compared to last year, we noticed many more products showing 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 3 – 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 4 – 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 5 – Browse Time

We used a script in conjunction with *HTTPWatch (Basic Edition, version 6.1)* 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 8 (Version 8.0.6001.18783)* 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 6 – 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 8 (Version 8.0.6001.18783)* 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.

## Benchmark 7 – 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 Internet Security Product).

Installation time includes the time taken by the product installer to download components required in the installation. This may include mandatory updates (e.g. Microsoft Security Essentials) or the delivery of the application itself from a download manager (e.g. McAfee Internet Security 2010, BitDefender Internet Security 2010). 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.

## Benchmark 8 – 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 **OSCheck** (v1.3.1004), 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 OSCheck, we compared the two signatures and calculated the total disk space consumed by all (and only) new files added 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.

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

## Benchmark 9 – 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 **RegistryCounter.exe**, an application which conducts a count of all keys, errors and values under HKEY\_LOCAL\_MACHINE and HKEY\_USERS.

Two Registry Key counts are taken, one prior to installation and a second immediately following a reboot after installation. To obtain our result, we calculated the difference between these two registry key totals.

## Benchmarks 10-15 – 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 10 – 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	Category	Number	Size (bytes)
DOC	Documents	8	30,450,176
DOCX	Documents	4	13,522,409
PPT	Documents	3	5,769,216
PPTX	Documents	3	4,146,421
XLS	Documents	4	2,660,352
XLSX	Documents	4	1,426,054
PDF	Documents	73	136,298,049
ZIP	Documents	4	6,295,987
7Z	Documents	1	92,238
JPG	Media	351	31,375,259
GIF	Media	6	148,182
MOV	Media	7	57,360,371
RM	Media	1	5,658,646
AVI	Media	8	78,703,408
WMV	Media	5	46,126,167
MP3	Media	28	191,580,387
EXE	PE	19	2,952,914
DLL	PE	104	29,261,568
AX	PE	1	18,432
CPL	PE	2	2,109,440

File format	Category	Number	Size (bytes)
CPX	PE	2	4,384
DRV	PE	10	154,864
ICO	PE	1	107,620
MSC	PE	1	41,587
NT	PE	1	1,688
ROM	PE	2	36,611
SCR	PE	2	2,250,240
SYS	PE	1	37,528,093
TLB	PE	3	135,580
TSK	PE	1	1,152
UCE	PE	1	22,984
EXE	PE	19	2,952,914
DLL	PE	104	29,261,568
AX	PE	1	18,432
CPL	PE	2	2,109,440
CPX	PE	2	4,384
DRV	PE	10	154,864
ICO	PE	1	107,620
MSC	PE	1	41,587
NT	PE	1	1,688



File format	Category	Number	Size (bytes)
ROM	PE	2	36,611
SCR	PE	2	2,250,240
SYS	PE	1	37,528,093
TLB	PE	3	135,580

File format	Category	Number	Size (bytes)
TSK	PE	1	1,152
UCE	PE	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 10 – Third Party Program Installation

This test measured how much time was required to install and uninstall a third party application. For this test, *CommandTimer.exe* timed how long it took to install and uninstall the Microsoft .NET Framework 2.0 (\*.msi) application on the test machine.

This test was conducted five times to obtain the average time to install/uninstall a third party program, with the test machine rebooted between each sample to remove potential caching effects.

## Benchmark 11 – 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	Category	Number	Size (bytes)
JPEG	Media	343	30,668,312
GIF	Media	9	360,349
PNG	Media	5	494,780
MOV	Media	7	57,360,371
RM	Media	1	5,658,646
AVI	Media	8	78,703,408
WMV	Media	5	46,126,167
MP3	Media	28	191,580,387
PDF	Documents	73	136,298,049
ZIP	Documents	4	6,295,987
7Z	Documents	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 13 – File Format Conversion (MP3 → WAV, MP3 → WMA)

This test measured how much time was required to convert an MP3 into a WAV file and subsequently, convert the same MP3 sample into a WMA file. The sample MP3 used was 3,375,104 bytes in size.

To encode the MP3 into another format, we used an application called *ffmpeg.exe*. The format conversion process was timed using *CommandTimer.exe*.

This test was conducted five times to obtain the average conversion speed between these formats, 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 404 files over 277,346,661 bytes. 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	Category	Number	Size (bytes)
DOC	Documents	8	30,450,176
DOCX	Documents	4	13,522,409
PPT	Documents	3	5,769,216
PPTX	Documents	3	4,146,421
XLS	Documents	4	2,660,352
XLSX	Documents	4	1,426,054
JPG	Media	351	31,375,259
GIF	Media	6	148,182
MOV	Media	7	57,360,371
RM	Media	1	5,658,646
AVI	Media	8	78,703,408
WMV	Media	5	46,126,167
<b>Total</b>		<b>404</b>	<b>277,346,661</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.