westcoast labs

Test Report February 2012

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

WCL Corporate Offices and Test Facilities

USA Headquarters and Test Facility

West Coast Labs, 16842 Von Karman Avenue, Suite 125, Irvine, CA 92606, U.S.A. Tel: +1 (949) 870 3250, Fax: +1 (949) 251 1586

European Headquarters and Test Facility

West Coast Labs, Unit 9 Oak Tree Court, Mulberry Drive, Cardiff Gate Business Park, Cardiff, CF23 8RS, UK.
Tel: +44 (0) 29 2054 8400, Fax: +44 (0) 29 2054 8401

Asia Headquarters and Test Facility

West Coast Labs, A2/9 Lower Ground Floor, Safdarjung Enclave, Main Africa Avenue Road, New Delhi 110 029, India. Tel: +91 (0) 11 4602 0622, Fax: +91 (0) 11 4602 0633

Date: 13th February 2012 **Version:** 1.2

Author(s): Andre Hall, Robert Poghen

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

Contents

Introduction 4

Executive Summary 7

Comparative Solutions 8

Test Objectives 9

Test Network 11

Test Cases and Methodologies 12

Test Results 28

Rollups and weighted scores 36

Conclusion 38

Disclaimer 39

Revision History 40

Appendix A 41

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

Introduction

Following discussions between representatives of Kaspersky Lab (hereafter referred to as Kaspersky) and West Coast Labs (hereafter referred to as WCL) a universal test outline was constructed for, and agreed with Kaspersky relating to tests against the Application Control functionality using whitelisting of their product Kaspersky Endpoint 8 compared with other solutions on the market offering both Application Control and Anti-Malware functionality.

This document forms the report of those tests carried out against the versions as detailed below, looking at several aspects of the overall solutions including Whitelist Creation and Management Capabilities, the technical architecture and capabilities thereof, the granular control of applications once they had been launched, more advanced technical capabilities, and aspects of management and reporting for the solution.

Analyst firm Gartner have already tackled the subject of Application Control testing in several documents, stating in July 2011[1] that "*Enterprises should prepare themselves for the reality that their applications and data will be used in unexpected ways, abused, stolen, and attacked by outsiders and insiders. Application security is "a must." Despite the young age of the overall application security space, numerous technology markets are offering relatively mature*

[1] Gartner, [Hype Cycle for Application Security, 2011], [18 July 2011]. Please note that Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

Introduction

application and data security technologies, mitigating the risk of internal and external attacks.", and indeed have offered advice[2] for both end users ("*Default deny application control and Whitelist systems, however, offer some game-changing protection potential versus blacklisting solutions. Default deny Whitelist puts endpoints into a stronger defensive posture by preventing any software not explicitly allowed by policy from installing or launching.*") and the vendors ("*Vendors must combine proven anti-malware tools, data protection capabilities, and new technologies such as live reputation database lookup and Whitelist to provide customers with effective, manageable protection on a growing variety of traditional and emerging endpoint platforms.*") as to their suggestions as to how this technology will impact the market.

In accordance with other industry sources[3] [4], not only are the source results reported back but each result has also been given a relative weighting that was agreed between Kaspersky and WCL.

The inclusion of both the source data and the weighted results allows for readers of this report to be tied not to one interpretation of a set of results, but to be able to construct their own algorithms and weightings based upon the relevance of each

[2] Gartner, [Endpoint Protection Platforms Blending Security, System Management, and Data Protection], [17 May 2011]

[3] http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=COL&ObjectId=11983&tth=DYN&tt=siteemail&iDyn=2

[4] http://eugene.kaspersky.com/2011/09/30 */benchmarking-without-weightings-like-a-burger-without-a-bun/*

Introduction

test case to the business requirements of their organisation if they so require. This is important, as functionality across the solutions may not be exactly like-for like in terms of how particular features are implemented to achieve the end result.

Of course, the final decision for any business should involve a combination of results from independent test reports such as this, and a consideration of whether the required end result for the business situation is reached rather than the usage of a specific technology.

The rest of this document details the high level test objectives, the test environment that was used, the individual test cases and their associated methodology, test results, and finally a conclusion drawing together all of the tests and processes.

Testing was performed in WCL's Lab in Irvine, California during October 2011.

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

Executive Summary

The test cases detailed below were designed and agreed with a focus on using whitelisting technologies to enforce application control and concentrated on a wide range of both functionality and management.

The testing shows that against these specific test cases, Kaspersky generally perform very well and are followed very closely by one of the other vendors, then trailed by the second competitive vendor. This, however, may be to do with the differing approaches used by the second competitive company to perform their application control which does not rely so much on the same types of whitelisting technologies. A further vendor participated in the testing but then, subsequent to the testing being completed, requested to withdraw.

Considered in the full report are eight major areas of functionality, then elements of these were combined to provide a rollup score based around testing a scenario called *Default Deny* which looks at situations where a fully locked down console might be required, for example a PoS terminal in a retail outlet.

Overall, the findings of the engineers were that each of the solutions performs well in terms of application control generally, but the differing approaches highlight the breadth of Kaspersky’s offering and the use of their whitelisting-based technology.

Comparative solutions

At Kaspersky's request for comparative tests, WCL approached several other security vendors with additional Application Control/Whitelisting or similar functionality, such as might be used in a corporate environment.

The vendors who agreed to participate and whose solutions were examined are as detailed below:

- Kaspersky: Security Center 9 & Endpoint Security 8
- Symantec: Endpoint Protection 12.1
- McAfee: Solidcore Application Control 5.2.0 with ePolicy Orchestrator 4.6
- A third vendor who participated in the testing but then, subsequent to the testing being completed, requested to withdraw.

The following vendors were also invited, but declined to participate in any testing:

- Lumension
- Bit9
- Coretrace

Test Objectives

The objective of this testing programme was to determine the overall effectiveness of the solutions by conducting a series of test cases that focus specifically upon their application control and default deny functionality (or similar technology) using whitelisting, with each designed to assess a specific area of functionality. Results are provided in the following areas, along with a discussion:

White List creation

This testing covered the abilities of the solutions to build a whitelist repository of acceptable applications from a number of different sources including trusted sources and network accessible hosts, and to group these by appropriate labels.

White List Compilation

This testing covered the abilities of the solutions to build and compile both black and white lists of acceptable applications based upon specific information about the software such as certificates, filename, and metadata.

Application Control Policies Management

This testing examined the control of the execution of specific software by a number of parameters including the management of known vulnerable software, and considered the trusted users and sources from which whitelist rules could be determined. This area also covered the read/write access capabilities of software subject to the control of the solution and considered memory injection against vulnerable applications, and policy control of various connection types.

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

Test Objectives

Granular Control Policies Management

This testing covered the ability of the solutions to handle access control by restricting access to various parameters of an endpoint system. These parameters included system processes and resources, such as the registry. Other areas included local disk and USB ports as well as network drives or folders.

White List and Application Control administration rights management

This testing considered the ability of the solutions to distribute the management of the administration within the solution to different groups of individuals.

Events monitoring and reports audit

This testing looked at the reporting processes of the solutions.

Administrating White List and Application Control

This testing considered the ability of an administrator or a user with specified rights to be given the opportunity to allow or deny specific pieces of software upon request from another user.

White List and Application Control testing

This testing considered the ability of the solutions to try out new policies and rules in a small sandboxed environment before distributing them to the wider network in order to determine whether the policies would have any deleterious effect on the hosts to which it was to be deployed.

Test Network

Testing was conducted at WCL's United States Headquarters, located in Irvine, California.

Each solution was installed in a client/server configuration (1 server, 1 client) on its own isolated network. All hosts used real, physical machines and no virtualisation of operating systems was used in this test.

The Windows Server 2003 R2 Operating System was installed with the latest system updates and the respective vendors' Application Control management packages on the machine designated as the Server.

Windows XP with Service Pack 3 was installed with the latest system updates for each vendor Application Control or Endpoint solution on the machine designated as the Client.

The hardware used in each case was an Intel based system with 4GB RAM and 500GB of hard disk space, with a system partition of 140GB.

All hosts were allowed internet connectivity, in order to ensure that each solution had the ability to download updates for databases and/or signatures, and to allow access to cloud-based resources if the solutions required it.

Test Cases and Methodologies

More detail is shown on the following pages, along with the methodology that was followed. These methodologies were designed in order to provide high level validation of the solution’s abilities in each individual area. Each of five major sections, delimited by section title in **bold**, is broken down into sub tests, with the details of the testing requested in *italics* and the description of the testing undertaken being in normal font text.

Test Cases and Methodologies

- **Test Case 01** - **White List Creation**

a) *The capability of the solution to fill the White List by collecting information about the files from a trusted system and then representing all the information that was found in a report from this trusted system.*
The solution had its functionality examined by being installed on hosts that already had software pre-installed. The solution was then tested to ensure that some of these pre-installed software installations did not execute. This software was then added into the appropriate list of allowed software by collecting information about it, and the functionality was again tested to ensure compliance.

b) *The capability of the solution to fill the white list by collecting files from a trusted local or network folder(s) and to represent all the information about the files that were found in a report. The capability of the solution to extract files from an installation package from trusted folder(s) and to represent all the information about the files in a report.*
The solution had its functionality examined by setting up a trusted source (such as local or network folder) that contain allowed software and checking that information of all software from the trusted source had been collected correctly. The solution had its functionality further examined by checking that all the information related to the files that were contained in the installation packages from the trusted source had been collected correctly.

Test Cases and Methodologies

c) *The capability of the solution to fill the white list by collecting information about an application's activity on machines in an accessible network and to represent all the information about files that were found in a report.*

The solution had its functionality examined by scanning applications that were running on each machine on the network where the endpoint was installed, and checking that all the information about processes had been collected correctly.

d) *The capability of the solution to fill a white list by collecting information regarding installed (but not necessarily running) applications on machines in an accessible network and to represent all the information about files that were found in a report.*

The solution had its functionality examined by scanning applications that were installed on each machine (inventory case) on the network where the endpoint was installed, and checking that all information about the installed software had been collected correctly.

e) *The capability of the solution to check the status of a file from a global repository and to represent that information in report.*

As part of the test case below where an unknown application was being executed, traffic captures were taken to look for any calls to third party servers. This was then tallied with any publicly available information for the solution about the usage of such trusted third party sites to verify whether this functionality was used.

Test Cases and Methodologies

f) *The capability of the solution to provide a repository of trusted certificates or certificates' meta-information.*
The solution was tested to ensure that certificates and all files signed by them could be added to the white or black list.

g) *The capability of the solution to fill the white list from a local repository using metadata where there is trusted software.*
The solution was tested to ensure that any files signed by the software's metadata were then added into the appropriate list of allowed software, and the functionality was again tested to ensure compliance.

h) *The capability of the solution to fill a white list from a local repository of trusted files.*
The solution was tested to ensure that the files from the repository did not execute. The files from the repository were then added into the appropriate list of allowed software, and the functionality was again tested to ensure compliance.

i) *The capability of the solution to fill a white list from a local repository of categories of trusted software grouped by functional area (such as browsers, IM, games, etc).*
The solution was installed alongside a series of other applications that may be found on a typical endpoint machine, for example productivity software, two variants of internet browsers, and one or two games, and then specific categories of application types were first blocked, then allowed. Whilst these policies were in place, applications within that category type were then

launched, and note was taken of the adherence of the solution to either blocking or allowing the applications.

Further, a policy was set first to allow, then to disallow, applications with specific characteristics to be executed. Execution of the software then took place under each of these scenarios.

Following this, and under each of the policies set above, the application being tested had some of its characteristics above altered, and then an attempted execution took place.

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

Test Cases and Methodologies

- **Test Case 02 - White List Compilation**

  a) *The capability of the solution to add files to the White or Black list by list properties such as file name, file extension, file signature, file path, etc.*
  A policy was set first to allow, then to disallow applications with specific characteristics included in the above named list to be executed. Execution of the software then took place to validate the functionality.

  b) *The capability of the solution to add a certificate to the White or Black list and cover files signed by that certificate.*
  The solution was tested to ensure that software signed by some certificates did not execute. The certificates' metadata was then added into the appropriate list of allowed certificates, and the functionality was again tested to ensure compliance.

  c) *The capability of the solution to add software to a White or Black list by its metadata or file information and to cover files from the same package.*
  The solution was tested to ensure that some component parts of the overall software package did not execute. The software metadata was then added into the appropriate list of allowed software, and the functionality was again tested to ensure compliance.

  d) *The capability of the solution to add a trusted folder to a White list to allow all software from that folder to be in the White list.*

Test Cases and Methodologies

The solution had its functionality examined by setting up a trusted folder which allows software placed into it to be automatically delivered to the whitelist without the need for manual intervention. Software was then placed into the trusted folder to check that all software included in that trusted folder was allowed to execute correctly. Further, the solution had its functionality examined by checking that all the files contained in the installation packages on that trusted folder were also allowed to execute as trusted.

e) *The capability of the solution to add a trusted source to a White list to allow all software from that source to be in the White list.*
The solution had its functionality examined by setting up a trusted source which allows software placed into it to be automatically delivered to the whitelist without the need for manual intervention (such as local, network folder, host, etc.). The software was then placed into the trusted source to check that all software included in that trusted source was allowed to execute correctly. Further, the solution had its functionality examined by checking that all the files contained in the installation packages on that trusted source were also allowed to execute as trusted.

f) *The capability of the solution to add a trusted updater/installer to the White list to allow all files/updates created by that process to be in the White list.*
The solution had its functionality examined by setting up a trusted updater/installer. This trusted updater was then executed to update files as appropriate on the endpoint host machine and those updated files were checked as to whether they were allowed to execute as trusted.

Test Cases and Methodologies

g) *The capability of the solution to add a trusted updater/installer to the White list to provide all files/updates created by the processes chain to be in the White list.*

The solution had its functionality examined by setting up a trusted updater/installer. That trusted updater was then not used to update files subsequently. The individual files which create a chain of updater processes were examined as to whether they were allowed to execute to update files on the machine, and then subsequently updated files were checked as to whether they were allowed to execute as trusted.

h) *The capability of the solution to add a user/group/role (from Active Directory or a custom list) as assigned to approve software and to add it to the White List.*

The solution had its functionality examined by setting up a trusted user, group, or role. The trusted user then added some software into the appropriate list of allowed software and that software was checked as to whether it was allowed to execute as trusted.

Test Cases and Methodologies

- **Test Case 03 - Application Control policies management**

  a) *The capability of the solution to setup a policy for a user/group/role (from Active Directory or a custom group) to allow or deny execution on a specific application or file or category.*
  A policy was set first to allow, then to disallow applications with specific characteristics included in the above named list for specific users/groups/roles to be executed. Execution of the software then took place, and it was observed whether the solution complied with the policy rules.

  b) *The capability of a solution to add a user/group/role (from Active Directory or a custom group) to allow or deny access for specific application or file or category.*
  A policy was set first to allow, then to disallow applications with specific characteristics included in the above named list for specific users/groups/roles to access. Access to the software then took place, and it was observed whether the solution complied with the policy rules.

  c) *The capability of the solution to setup exclusion for user/group/role (from Active Directory or custom) or specific application/file/category from policies.*
  A rule was set to exclude users, or groups, roles, individual applications, files or a category of files with specific characteristics from policies. Access by those users, and to that software then took place, and it was observed whether the solution complied with the policy rules.

Test Cases and Methodologies

- **Test Case 04 – Granular Controls policies management**

  a) *The capabilities of the solution to setup a policy for a specific application, file or category to restrict access to network resources.*

  A policy was first set to allow, and then disallow access to a specific network resource such as a shared drive. Access to the drive was attempted and it was observed whether the solution compiled with the policy rules.

  b) *The capability of the solution to setup a policy for a specific application or file or category to restrict access to system configuration resources.*

  A policy was set first to allow, then to disallow applications with specific characteristics as described above to access system configuration resources such as the Registry Editor. Access to the software was then attempted, and it was observed whether the solution complied with the policy rules.

  c) *The capability of the solution to setup a policy for a specific application or file or category to restrict access to system files.*

  A policy was set, first to allow, then to disallow applications with specific characteristics as described above to access system file resources such as the central DLLs or DirectX. Access to the software then took place, and it was observed whether the solution complied with the policy rules.

  d) *The capability of the solution to setup a policy for a specific application or file or category to restrict access to processes.*

  A policy was set first to allow, then to disallow applications with specific characteristics as described above to access specific running processes on the host machine. Access to the software then took place, and it was observed whether the solution complied with the policy rules.

e) *The capability of the solution to setup a policy for a specific application or file or category to restrict actions for local resources such as read, write, delete, create, etc.*

A policy was set, first to allow, then to disallow applications with specific characteristics to perform various read/write type operations on the local file system. Access to the software then took place, and it was observed whether the solution complied with the policy rules.

f) *The capability of the solution to setup a policy for a specific application or file or category to restrict actions for network resources such as send and receive with specified parameters.*

Testing considered the blocking of specific application ports. A policy or rule was introduced that stopped specific types of traffic being used, such as FTP. Connection attempts were then made from the solution to an appropriate service to ensure that all such connections were both blocked and logged in reports.

g) *The capability of the solution to setup a policy to restrict execution of potentially vulnerable applications. Further, the capability of the solution to provide information about associated vulnerabilities, and the capability of the solution to provide patches (or links to patches) for the vulnerable applications was examined.*

Test Cases and Methodologies

A policy was set first to allow, then to disallow a known vulnerable application to execute. Access to the software then took place, and it was observed whether the solution complied with the policy rules. Further to this, note was taken of any remedial advice offered by the solutions and whether this advice then subsequently allowed the user to easily access appropriate patches for the vulnerable application.

h) *The capability of the solution to setup a policy for a vulnerable application to restrict its actions. The capability of the solutions to handle buffer overflow or memory injection against vulnerable applications.*
A policy was set first to allow, then to disallow a known vulnerable application the ability to conduct specific actions. Access to the software then took place, with those specific actions attempted and any outcomes were noted. Further to this, a known, and unpatched vulnerable application was subjected to buffer overflow and memory injection attempts as appropriate performed against them in order to ascertain whether the solution was capable of protecting the application in question.

Test Cases and Methodologies

- **Test Case 05 - White List and Application Control administration rights management**

  a) *The ability of the solution to setup a policy for a user/group/role (from Active Directory or custom groups) to allow or deny the ability to add or delete applications from the White List.*

     A policy was set first to allow, then to disallow users or groups from having the ability to add or delete applications from the White list. This was then tested to ensure that the functionality was valid and any restrictions were noted.

  b) *The ability of the solution to setup a policy for a user/group/role (from Active Directory or custom groups) to allow or deny the changing of policies of the Application Control functionality.*

     A policy was set first to allow, then to disallow users or groups from having the ability to change application control policies. This was then tested to ensure that the functionality is valid and any restrictions were noted.

  c) *The ability of the solution to setup a policy for a user/group/role (from Active Directory or a custom group) to allow or deny the ability to approve applications by request.*

     A policy was set first to allow, then to disallow users or groups to have the ability to act as an approver of application requests to add details into the White list. This was then tested to ensure that the functionality was valid and any restrictions were noted.

Test Cases and Methodologies

- **Test Case 06 - Events monitoring and reports audit**

a) *The ability of the solution to create reports of applications execution on machines in a network.*

Reports were run to examine application execution.

b) *The ability of the solution to create reports of activity of applications on machines in a network.*

Reports were run to examine application activity.

c) *The ability of the solution to create reports of installations of applications on machines in a network.*

Reports were run to examine application installations.

d) *The ability of the solution to create reports of modifications of White List files or Application Control policies.*

Reports were run to examine modifications of white list files or application policies.

e) *The ability of the solution to create reports of requests from users for approving applications.*

Reports were run to examine application approval requests.

Test Cases and Methodologies

- **Test Case 07 - Administrating White List and Application Control**

  a) *The ability of the solution to create request from a user to an administrator for approval of an application (in the case that the application is blocked).*

  A request for a new application to be added to the whitelist was made from a normal user to an administrative user, and results were noted.

  b) *The ability of the solution to create a request from one user to another user that is assigned tor approve applications (in the case that the application is blocked).*

  A request for a new application to be added to the whitelist was made from a normal user to an appropriately entitled user with policies to approve applications, and results were noted.

Test Cases and Methodologies

- **Test Case 08 - White List and Application Control testing**

  a) *The ability of the solution to setup a testing mode with monitoring of a policies' actions without applying it.*

     The solution was examined for the ability to "sandbox" an application of policies to examine what the likely impact will be upon implementation of that policy. This involves implementing the rules on a real PC in a "test configuration" but without the usual associated blocking or alerts that come with a formally implemented policy. Policies were set as appropriate with already pre-known outcomes, and the results of this procedure were examined as to whether they concurred with what the outcome should have been. Results were noted as such.

  b) *The ability of the solution to check results of applying policies on a report of an audit or inventory to discover what applications may be blocked by current policies.*

     A report was run using a suggested policy against an inventory of installed applications to determine what the effects of the policy are against software that is already installed on the endpoints.

Default Deny mode

The reporting around Default Deny functionality has been constructed by taking results from three separate sections: Whitelist creation, Whitelist compilation and Policy management in order to provide an overview of this type of functionality in the context of whitelisting and application control.

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

Test Results

The following sections contain the results for all testing conducted as part of this report. Tests were broken down into a number of Test Cases as detailed above, and with major overarching groups. These groups are reproduced below for information.

- Whitelist Creation
- Whitelist Compilation
- Application Control Policies Management
- Granular Controls Policy Management
- Whitelist and Application Control Administration Rights Management
- Event Monitoring and Reports Audit
- Administrating Whitelist and Application Control
- Whitelist and Application Control Testing

Scoring was based upon the observations of the engineers conducting the test, and individual components were awarded scores based upon the scale shown in Table 1.0

9-10	**Excellent Capabilities:** The product is provided out of the box and validated by vendor references.
7-8	**Strong Capabilities:** The product appears to satisfy requirements, but may have newly offered functions that are not validated by vendor references.
5-6	**Capable:** The product can meet capabilities through customization. Documentation and consulting resources are available to help design the desired solution.
3-4	**Somewhat Capable:** All functionality may not be available or doesn't provide the desired level of automation.
1-2	**Minimal Capabilities:** The product has significant deficiencies in this area and offers minimal automation.
0	**No Capabilities**

Table 1.0 Scoring system used

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

The scoring system was scaled from 0, where a product had no capability in that area, up to 10 where it fulfilled the exact requirement of the test, was fully validated and documented and worked in an “out of the box” configuration.

Test Case 01 - Whitelist Creation

Test Case	Weight	**Kaspersky**	Vendor A	Vendor B	Industry Average
A	10	10	4	6	6.6
B	10	10	0	6	5.3
C	10	10	4	6	6.6
D	10	10	4	6	6.6
E	5	10	0	10	6.6
F	8	10	0	10	6.6
F	7	10	0	10	6.6
H	5	10	0	10	6.6
I	9	10	0	0	3.3

Table 2.0

Based upon the weighting in the area of Whitelist Creation, Kaspersky’s Application Control solution achieved top scores in each use case. The cases addressed the solution’s ability to collect files and applications from trusted systems and display detailed information in a report. Test cases verified whether the solution contained a global or local repository of trusted applications or files and could add to a white or black list. Additionally, the cases checked the solution’s ability to display information regarding application activity in a report and to add an application to the white or black list. Testing confirmed the solution was able to accurately perform the functions required. The solution also obtained the highest possible rollup scores in this area.

WCL engineers tested each of the above areas by performing a number of operations that might be typical of a business deployment situation – for example applications and trusted folders were added by an administrator and then it was confirmed that the whitelisting functionality performed correctly. This model was extended across into individual files' properties, and again correct operation was confirmed before some of the properties of the file were changed and the correct application of the rules to not allow access in these cases was again confirmed. In each of the cases it was confirmed that the product worked as expected within the confines of the test framework and specified test case.

Test Case 02 - Whitelist Compilation

Test Case	Weight	**Kaspersky**	Vendor A	Vendor B	Industry Average
A	5	10	10	10	10
B	8	10	4	10	8
C	8	10	10	10	10
D	7	10	0	10	6.6
E	7	10	0	10	6.6
F	8	10	0	10	6.6
G	9	10	0	10	6.6
H	8	10	4	10	8

Table 3.0

Test cases which represent the area of Whitelist Compilation demonstrated the solution's ability to add applications, trusted folders and sources to a white or black list by its properties, certificate, file information or metadata. In all of the above test cases Kaspersky's and Vendor B's Application Control achieved the highest possible scores based upon the weighting scores within each use case and obtained the highest total rollup scores in this area.

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

WCL engineers assembled various numbers of files and folders and placed some of these on trusted systems and others on shared network drives on the test network. Policies were created to add these as trusted sources, and then engineers validated the ability of the function to add files to the solution's white or black list from these trusted sources using various parameters - adding files to the white or black list by file properties, certificates, or by metadata were examined and proven by WCL engineers. Further, there was also validation that applications, trusted updaters, and all related files could be added to the white or black list.

Test Case 03 - Application Control Policies Management

Test Case	Weight	**Kaspersky**	Vendor A	Vendor B	Industry Average
A	10	10	10	10	10
B	9	10	10	10	10
C	9	10	10	10	10

Table 4.0

Each solution was tested to verify its ability to setup user and group policies through the use of Active Directory and to allow/deny execution of an application based upon its file type or category. Results for tests performed in this area show that the Kaspersky solution's implementation of Application Controls Policies Management were awarded the highest weighted scores available within every use case. Given this, the highest rollup score was also awarded in this area.

For these use cases engineers selected a specific application and checked its operation before and after the implementation and execution of a policy to restrict access. The application was started, then closed, and a policy to restrict its usage was applied. When the next attempt to start the application was made, the application did not start and a prompt displayed explaining the restriction. Further to this, engineers also created test scenarios where application control rights were *granted or denied to a particular user or group and related applications, and the solution was again shown to be correctly adhering to the implemented policies.*

Test Case 04 – Granular Control Policies Management

Test Case	Weight	**Kaspersky**	Vendor A	Vendor B	Industry Average
A	7	10	10	10	10
B	8	10	10	10	10
C	8	10	10	10	10
D	7	10	10	10	10
E	7	10	10	10	10
F	7	10	10	10	10
G	8	2	5	10	5.66
H	8	4	10	10	8

Table 5.0

Granular Control Policies Management demonstrates a solution's ability to setup policies for applications, files, and respective categories to restrict access to system critical system files and processes. Test results for this area showed that the Kaspersky solution gained top scores in six of the eight use cases whilst in the latter two cases out of eight the solution could perform only some of the features.

To achieve verification, WCL Engineers performed each of the described functions through the utilization of the solution's policy based granular controls. Engineers constructed various policies to restrict an application from accessing a specified network resource, or from accessing system configurations or specified system files. To give an example, engineers created policies which refused the user access to the Windows operating system Control Panel, which was subsequently validated as correctly enforced. Further additional policies were tested and validated regarding *the restriction of read, write, and delete access privileges on files and also for sending and receiving files.*

Also in this area was the requirement to validate that the solution could successfully restrict the execution of a vulnerable application. To satisfy this requirement, engineers executed publicly available applications with known vulnerabilities such as Adobe Reader 5. A policy was set to restrict access to vulnerable applications and, as per this policy, each product was restricted from starting on the system and displayed an appropriate warning prompt giving an appropriate explanation.

Test Case 05 - Whitelist and Application Control Administration Rights Management

Test Case	Weight	**Kaspersky**	Vendor A	Vendor B	Industry Average
A	8	10	10	10	10
B	8	10	10	10	10
C	8	10	10	10	10

Table 6.0

In this test case the Kaspersky solution again performed well in the area of Whitelist and Application Control Administration Rights Management, achieving top scores in all uses cases and possessing the highest possible rollup scores.

Test cases in this area were used to demonstrate that the solution is capable of creating user and group policies which allow for an administrator to allow/deny or add/change rules to whitelisting features. Observations were also made regarding whether the solution is able to create policies via Active Directory, allowing for application approval workflow. In each case, engineers have validated that all features in these test cases are functioning as expected.

Test Case 06 - Event Monitoring And Reports Audit

Test Case	Weight	**Kaspersky**	Vendor A	Vendor B	Industry Average
A	9	10	10	10	10
B	8	10	10	10	10
C	8	10	0	10	6.6
D	7	10	10	10	10
E	9	4	0	0	1.3

Table 7.0

Event Monitoring and Reports Audit testing demonstrates a strong implementation of features, with Kaspersky scoring high marks in four out of five use cases whilst being somewhat capable of functionality in the latter tests.

It is important to note in this case that the functionality described and tested under test cases 6E and 7B is unique to Kaspersky amongst the solutions tested.

Test Case 07 - Administrating Whitelist and Application Control

Test Case	Weight	**Kaspersky**	Vendor A	Vendor B	Industry Average
A	9	10	0	0	3.3
B	10	6	0	0	2

Table 8.0

In this test, the Kaspersky solution performed well, obtaining the highest score possible within the use case of its ability to create requests from a user to an administrator for approval of an application. In the case of the solution’s ability to create a request from user to another user that is assigned for approved applications, the feature is possible, however it does require some customization on the part of the administrator.

None of the competitive solutions offered either functionality – it is important to also look at test case 7B in conjunction with the comments described above, under Test Case 6.

Test Case 08 - Whitelist and Application Control Testing

Test Case	Weight	**Kaspersky**	Vendor A	Vendor B	Industry Average
A	8	10	10	10	10
B	9	0	10	0	3.3

Table 9.0

In the final test, the Kaspersky solution performed well, obtaining the highest score possible for its ability to setup a testing mode with monitoring of policies' actions without applying that policy across a live production network. The solution was not, however, able to check results of applying policies on a report of an audit or inventory to discover what applications may be blocked by current policies in the form tested.

Rollups and weighted scores.

The rollups and weighted scores as seen in Appendix A were applied, leading to an overall score measure as below for each of the areas.

Major area	Total available	Weighted roll up scores, percentage fulfillments			
		Kaspersky	Vendor A	Vendor B	Industry Average
Whitelist Creation	740	740, 100%	120, 16.2%	490, 66.2%	450, 60.8%
Whitelist Compilation	600	600, 100%	194,32.3%	600, 100%	465, 77.5%
Application Control Policies Management	280	280, 100%	280, 100%	280, 100%	280, 100%
Granular control policy management	600	488, 81.3%	560, 93.3%	600, 100%	549, 91.5%
Whitelist and Application Control Administration Rights Management	240	240, 100%	240, 100%	240, 100%	240, 100%
Event Monitoring And Reports Audit	410	356, 86.8%	240, 58.5%	320, 78.0%	305, 74.4%
Administrating Whitelist and Application Control	190	150, 78.9%	0,0%	0,0%	50, 26.3%
Whitelist and Application Control Testing	170	80, 47.1%	170, 100%	80, 47.1%	110, 64.7%
Overall Scores	3230	2934, 90.8%	1804, 55.9%	2610, 80.8%	2449.3, 75.8%

Table 10.0

Furthermore, as also described in Appendix A, a rollup score for a Default Deny mode was agreed and then calculated. For the calculation of this score, WCL used the test cases for Whitelist Creation, Whitelist Compilation, and Application Controls Policy Management.

Major area	**Total available**	**Weighted roll up scores, percentage fulfillments**			
		Kaspersky	Vendor A	Vendor B	Industry Average
Whitelist Creation	740	740, 100%	120, 16.2%	490, 66.2%	450, 60.8%
Whitelist Compilation	600	600, 100%	194, 32.3%	600, 100%	465, 77.5%
Application Control Policies Management	280	280, 100%	280, 100%	280, 100%	280, 100%
Overall Score	1620	1620, 100%	594, 36.7%	1370, 84.6%	1195, 73.7%

Table 11.0

This outcome, using this method of combining scores in this manner, showed that Kaspersky is fully capable of supporting a default deny mode as defined in the by using the Application Control application.

West Coast Labs' Conclusion

These test cases look to highlight both particular areas of functionality, and the products' relative performances in each. No one solution can ever be a panacea for all the potential threats that could affect an endpoint host, but combining technologies in this way offers a multi-layered, defence in depth approach that serves end users well, whoever the provider is.

What this comparative data highlights is that, across the major areas Kaspersky has generally performed extremely well – there are one or two areas that could be flagged up as needing improvement as described in the results above, specifically areas such as vulnerable executable management, some of the functionality and reporting around the management of whitelist administration, and checking the results of applying policies in a test "sandbox" environment. As far as the competitors go, the tests show that at least one other vendor (Vendor B) is following a roughly similar path, whilst Vendor A has decided to go down a different route of Application Control using fewer elements of pure whitelisting in their offerings.

That having been said, the general feeling by the WCL engineers is that each of these products performs well in the general area of application control, even if they go about it in different ways, but that these test cases go towards highlighting the breadth of the Kaspersky offering and showing that their technology is not only a highly useful and effective addition, but also that they are interacting with the multi-layered protection strategy at many layers and delivering a good set of functionality that will go a long way towards ensuring that their users are protected.

West Coast Labs Disclaimer

While West Coast Labs is dedicated to ensuring the highest standard of security product testing in the industry, it is not always possible within the scope of any given test to completely and exhaustively validate every variation of the security capabilities and/or functionality of any particular product tested and/or guarantee that any particular product tested is fit for any given purpose.Therefore, the test results published within any given report should not be taken and accepted in isolation.

Potential customers interested in deploying any particular product tested by West Coast Labs are recommended to seek further confirmation that the said product will meet their individual requirements, technical infrastructure and specific security considerations.All test results represent a snapshot of security capability at one point in time and are not a guarantee of future product effectiveness and security capability.

West Coast Labs provide test results for any particular product tested, most relevant at the time of testing and within the specified scope of testing and relative to the specific test hardware, software, equipment, infrastructure, configurations and tools used during the specific test process.

West Coast Labs is unable to directly endorse or certify the overall worthiness and reliability of any particular product tested for any given situation or deployment.

Kaspersky Application Control and Default Deny using Whitelisting Comparative Test Report

Revision History

Issue	Description of Changes	Date Issued
1.0	Application Control and Whitelisting Test Report	10th Feb 2012
1.1	Minor typographical changes	13th Feb 2012
1.2	Further minor typographical change	13th Feb 2012

Appendix A – weightings applied.

Test Case 01 - White List Creation

a) The capability of the solution to fill the white list by collecting information about the files from a trusted system and then representing all the information that was found in a report from this trusted system. **WEIGHTING APPLIED: 10**

b) The capability of the solution to fill the white list by collecting files from a trusted local or network folder(s) and to represent all the information about the files that were found in a report. The capability of the solution to extract files from an installation package from trusted folder(s) and to represent all the information about the files that in a report. **WEIGHTING APPLIED: 10**

c) The capability of the solution to fill the white list by collecting information about an application's activity on machines in an accessible network and to represent all the information about files that were found in a report. **WEIGHTING APPLIED: 10**

d) The capability of the solution to fill a white list by collecting information regarding installed (but not necessarily running) applications on machines in an accessible network and to represent all the information about files that were found in a report. **WEIGHTING APPLIED: 10**

e) The capability of the solution to check the status of a file from a global repository and to represent that information in report. **WEIGHTING APPLIED: 5**

f) The capability of the solution to provide a repository of trusted certificates or certificates meta-information. **WEIGHTING APPLIED: 8**

g) The capability of the solution to fill the white list from a local repository using metadata where there is trusted software. **WEIGHTING APPLIED: 7**

h) The capability of the solution to fill a white list from a local repository of trusted files. **WEIGHTING APPLIED: 5**

i) The capability of the solution to fill a white list from local repository of categories of trusted software grouped by functional area (such as browsers, IM, games, etc). **WEIGHTING APPLIED: 9**

Test Case 02 - White List Compilation

a) The capability of the solution to add files to the White or Black list by list properties such as file name, file extension, file signature, file path, etc. **WEIGHTING APPLIED: 5**

b) The capability of the solution to add a certificate to the White or Black list and cover files signed by that certificate. **WEIGHTING APPLIED: 8**

c) The capability of the solution to add software to a White or Black list by its metadata or file information and to cover files from the same package. **WEIGHTING APPLIED: 8**

d) The capability of the solution to add a trusted folder to a White list to allow all software from that folder to be in the White list. **WEIGHTING APPLIED: 7**

e) The capability of the solution to add a trusted source to a White list to allow all software from that source to be in the White list. **WEIGHTING APPLIED: 7**

f) The capability of the solution to add a trusted updater/installer to the White list to allow all files/updates created by that process to be in the White list. **WEIGHTING APPLIED: 8**

g) The capability of the solution to add a trusted updater/installer to White list to provide all files/updates created by processes chain to be in White list. **WEIGHTING APPLIED: 9**

h) The capability of the solution to add a user/group/role (from Active Directory or a custom list) as assigned to approve software and to add it to the White List. **WEIGHTING APPLIED: 8**

<u>Test Case 03 - Application Control policies management</u>

a) The capability of the solution to setup a policy for a user/group/role (from Active Directory or a custom group) to allow or deny execution on a specific application or file or category. **WEIGHTING APPLIED: 10**

b) The capability of a solution to add a user/group/role (from Active Directory or a custom group) to allow or deny access for specific application or file or category. **WEIGHTING APPLIED: 9**

c) The capability of the solution to setup exclusion for user/group/role (from AD or custom) or specific application/file/category from policies. **WEIGHTING APPLIED: 9**

<u>Test Case 04 – Granular Controls policies management</u>

a) The capabilities of the solution to setup a policy for a specific application, file or category to restrict access to network resources. **WEIGHTING APPLIED: 7**

b) The capability of the solution to setup a policy for a specific application or file or category to restrict access to system configuration resources. **WEIGHTING APPLIED: 8**

c) The capability of the solution to setup a policy for specific application or file or category to restrict access to system files. **WEIGHTING APPLIED: 8**

d) The capability of the solution to setup a policy for a specific application or file or category to restrict access to processes. **WEIGHTING APPLIED: 7**

e) The capability of the solution to setup a policy for a specific application or file or category to restrict actions for local resources such as read, write, delete, create, etc. **WEIGHTING APPLIED: 7**

f) The capability of the solution to setup a policy for specific application or file or category to restrict actions for network resources such as send, receive with specified parameters. **WEIGHTING APPLIED: 7**

g) The capability of the solution to setup a policy to restrict execution of potentially vulnerable applications. Further, the capability of the solution to provide information about associated vulnerabilities, and the capability of the solution to provide patches (or links to patches) for the vulnerable applications was examined. **WEIGHTING APPLIED: 8**

h) The capability of the solution to setup a policy for a vulnerable application to restrict its actions. The capability of the solutions to handle buffer overflow or memory injection against vulnerable applications. **WEIGHTING APPLIED: 8**

Test Case 05 - White List and Application Control administration rights management

a) The ability of the solution to setup a policy for a user/group/role (from Active Directory or custom groups) to allow or deny the ability to add or delete applications from the White List. **WEIGHTING APPLIED: 8**

b) The ability of the solution to setup a policy for a user/group/role (from Active Directory or custom groups) to allow or deny the changing of policies of the Application Control functionality. **WEIGHTING APPLIED: 8**

c) The ability of the solution to setup a policy for a user/group/role (from Active Directory or a custom group) to allow or deny the ability to approve applications by request. **WEIGHTING APPLIED: 8**

Test Case 06 - Events monitoring and reports audit

a) The ability of the solution to create reports of applications execution on machines in a network. **WEIGHTING APPLIED: 9**

b) The ability of the solution to create reports of activity of applications on machines in a network. **WEIGHTING APPLIED: 8**

c) The ability of the solution to create reports of installations of applications on machines in a network. **WEIGHTING APPLIED: 8**

d) The ability of the solution to create reports of modifications of White List files or Application Control policies. **WEIGHTING APPLIED: 7**

e) The ability of the solution to create reports of requests from users for approving applications. **WEIGHTING APPLIED: 9**

Test Case 07 - Administrating White List and Application Control

a) The ability of the solution to create request from a user to an administrator for approval of an application (in the case that the application is blocked). **WEIGHTING APPLIED: 9**

b) The ability of the solution to create a request from user to another user that is assigned for approve applications (in the case that the application is blocked). **WEIGHTING APPLIED: 10**

Test Case 08 - White List and Application Control testing

a) The ability of the solution to setup testing mode with monitoring of policies actions without applying it. **WEIGHTING APPLIED: 8**

b) The ability of the solution to check results of applying policies on a report of an audit or inventory to discover what applications may be blocked by current policies. **WEIGHTING APPLIED: 9**

Default Deny Mode

Default Deny mode was calculated by combining the scores from Whitelist Creation, Whitelist Compilation and Application Controls Policy Management.

USA SALES

T +1 (949) 870 3250

EUROPE SALES

T +44 (0) 2920 548400

CHINA, KOREA, JAPAN, TAIWAN SALES

T +86 1 343 921 7464

REST OF THE WORLD SALES

T +44 (0) 2920 548400

CORPORATE OFFICES AND TEST FACILITIES

US Headquarters and Test Facility

West Coast Labs
16842 Von Karman Avenue, Suite 125,
Irvine, California, CA92606, USA
T +1 (949) 870 3250, **F** +1 (949) 251 1586

European Headquarters and Test Facility

West Coast Labs
Unit 9, Oak Tree Court, Mulberry Drive
Cardiff Gate Business Park, Cardiff CF23 8RS, UK
T +44 (0) 2920 548400, **F** +44 (0) 2920 548401

Asia Headquarters and Test Facility

A2/9 Lower Ground floor, Safdarjung Enclave,
Main Africa Avenue Road, New Delhi 110 029, India.
T +91 (0) 11 4602 0622, **F** +44 (0) 11 4602 0633

E info@westcoast.com

W www.westcoastlabs.com