Comparative Assessment of Kaspersky Internet Security 2013

August 2012

Contents:

Introduction:

This report has been commissioned by Kaspersky Lab to serve as an independent, comparative efficacy assessment of ten leading antivirus / internet security applications, including its own, Kaspersky Internet Security 2013 (KIS 2013).

Someone may come across software errors that could result in a system being compromised. When this happens he can react in one of two ways: alert the program developer to the problem, or try to profit from the glitch. In the latter case, information about errors can be sold on specialized forums, or shared with the malicious developers of exploit packs. Antivirus vendors also value this sort of information, so they can add details of known vulnerabilities to their databases to help protect against infections.

Recently the problem of software vulnerabilities became more acute with the emergence of special teams established by government or commercial organizations that actively seek ways to break into the computers of rivals and competitors. Vulnerabilities discovered here are kept secret and used for targeted attacks only. Nobody finds out until the victim of an attack finally realizes what has happened and investigates. Sometimes it can take years for a vulnerability to come to light and be fixed. When a specialist finally uncovers the details of an attack, he also has a choice: inform the antivirus vendors or sell the information to those creating exploits.

Although new exploits pose an obvious danger, most antivirus vendors still focus only on detecting already-known exploits and adding any new discoveries to their databases. It's not easy to add safe, universal exploit detection technology to antivirus products, but solving this problem can raise user protection to a whole new level. However, the lack of authoritative tests which evaluate exploit detection mechanisms, making it hard for customers to choose wisely, while also discouraging antivirus developers from investing in complex new technologies.

The purpose of the test is to assess the effectiveness of the new **Automatic Exploit Prevention** (AEP) technology used in Kaspersky Internet Security 2013 (KIS 2013). As the name suggests, this innovative technology protects users against exploits, which are now the primary method of getting malware onto user's computers. AEP is an umbrella concept and includes several sub-technologies, which are mostly behavior-based. Some of these technologies are unique to Kaspersky Lab and currently have patent-pending status. This behavior-based approach makes it possible to block unknown exploits and even protects users from some zero-day vulnerabilities. AEP is a significant step forward in proactive protection which, in combination with the multilayered security model created by Kaspersky Lab, offers maximum protection to the user.

What is an exploit?

Exploits are not technically malware, rather, they are data, commands or code which use a vulnerability in software to bring about some unintended behaviour. Invariably, the unintended behaviour instigated is used to download or execute a payload, which is itself malware.

Exploits utilise the vulnerabilities in a range of software and applications. The most commonly targeted applications on a PC are; Adobe Acrobat Reader, Java, the Windows OS itself and Adobe Flash, however, any installed application can be a target.

An increasing proportion of malware utilise exploits, these include well known threats such as the Cornflicker worm, Flashback on the Mac and the high profile Stuxnet, which targeted supervisory control and data acquisition functions in Middle Eastern nuclear facilities.

In order to try and counter cybercriminals, vendors release updates for applications once a vulnerability is identified. The problem vendors face however, is that due to its complexity and sophistication, modern software is likely to have many vulnerabilities, so as soon as a known vulnerability is patched, cybercriminals are likely to have found a new one to exploit. This fact is well documented and exemplified by the notion of "Exploit Wednesday", when researchers often find new malicious exploits in the wild following Microsofts Patch Tuesday, when the vendor releases updates on the second Tuesday of each month for known vulnerabilities.

The Kaspersky AEP technology is designed to target malware which uses software vulnerabilities and does not rely on blacklisting to identify this malware, so should be effective against zero day threats of this type.

Security Applications Tested:

The security applications tested were as follows:

- AVG Internet Security 2012
- Avira Internet Security 2012
- Bitdefender Internet Security 2013
- Eset Smart Security V5
- F-Secure Internet Security 2012
- Kaspersky Internet Security 2013
- McAfee Total Protection 2012
- Microsoft Security Essentials V4
- Symantec Norton Internet Security 2012
- Trend Micro Titanium Maximum Security 2012

Methodology Used in the Test

Why we don't use only ITW

Initially we planned to test products on 'in-the-wild' (ITW) exploits only. Later, though we rejected this idea for the following the reasons:

1. Many vendors detect exploits using traditional methods, over a period of time. Our goal was to test the products' ability to detect zero-day threats. Turning off database updates and internet connection would be unrealistic, because many products actively use cloud technologies.
2. Usually vendors do not detect exploits in isolation, but via a method of packing or encryption. Therefore, some ITW exploits would be detected before those appearances in the wild, but there would be no detection if the encryption method was modified.
3. ITW exploits are represented as exploit packs, include a limited set of exploits only, and rely on samples where vulnerabilities have yet to be closed by the majority of potential victims.

That's why we decided to use ITW exploits and samples generated by Metasploit in this test.

What is Metasploit ?

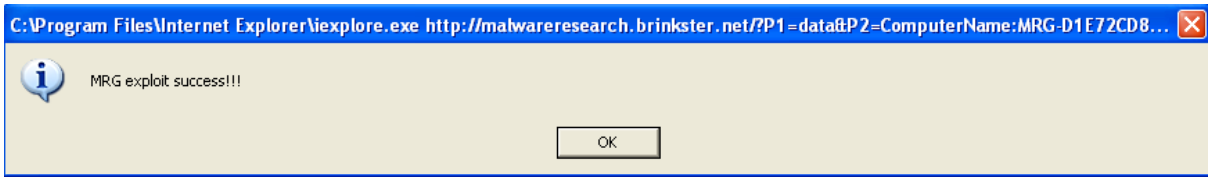
A collaboration between the open source community and Rapid7, Metasploit software helps security and IT professionals identify security issues, verify vulnerability mitigations, and manage expert-driven security assessments to provide true security risk intelligence. Capabilities include smart exploitation, password auditing, web application scanning, and social engineering. Teams can collaborate with Metasploit and present their findings in consolidated reports.

Used vulnerabilities

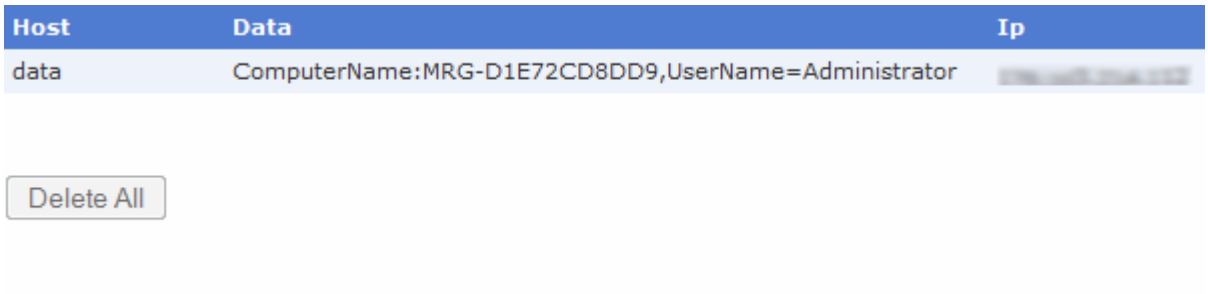
We impartially chose a set of vulnerabilities according to the following criteria:

1. An ability to execute code. We didn't consider vulnerabilities that don't lead to arbitrary code execution on the victim's computer, as they don't result in system infection, just a Denial of Service.
2. We generally tried to use the most recently detected vulnerabilities.
3. Vulnerabilities that are used in ITW exploit packs or that have been used at an earlier date.
4. Vulnerabilities invariably lead to the execution of malware code. If we couldn't reproduce a payload's execution without the antivirus products installed, the vulnerability was excluded from the test.

As a payload we chose to download and execute a PE file constructed by us. Its execution on the victim-machine due to the use of an exploit obviously indicates a security breach.



The malicious simulator also sent the computer name, user name and the IP address to a remote website:



Testing was conducted with each security application being fully updated, having a live internet connection and being configured with default settings,

Security applications were fully functional trials or anonymously registered versions.

No vendors, including Kaspersky were aware of the time or location of any testing to ensure any telemetry data could not be used to influence product performance.

The following exploits were used:

1. CVE2012-0158 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0158

This enables remote attackers to execute arbitrary code via a crafted (a) website, (b) Office document, or (c) .rtf file that triggers "system state" corruption, as exploited in the wild in April 2012, aka "MSCOMCTL.OCX RCE Vulnerability." We crafted a .doc file and opened it in Microsoft Office 2010 on Windows XP SP3

2. CVE2012-0003 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0003
   A vulnerability in winmm.dll in Windows Multimedia Library in Windows Media Player (WMP) in Microsoft Windows XP SP2 and SP3, Server 2003 SP2, Vista SP2, and Server 2008 SP2 allows remote attackers to execute arbitrary code via a crafted MIDI file, aka "MIDI Remote Code Execution Vulnerability". The crafted MIDI file was opened in Windows Media Player on Microsoft Windows XP SP3
3. CVE2012-1875 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1875
   Microsoft Internet Explorer 8 does not properly handle objects in memory, allowing remote attackers to execute arbitrary code by accessing a deleted object, aka "Same ID Property Remote Code Execution Vulnerability". We crafted an HTML file and stored it on a web server. We accessed it with MS IE8.
4. CVE 2011-1260 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-1260
   Microsoft Internet Explorer 8 and 9 do not properly handle objects in memory, allowing remote attackers to execute arbitrary code by accessing an object that (1) was not properly initialized or (2) is deleted, aka "Layout Memory Corruption Vulnerability". We crafted an HTML file and stored it on a web server. We accessed it with MS IE8.
5. CVE 2007-0038 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0038
   A stack-based buffer overflow in the animated cursor code in Microsoft Windows 2000 SP4 through Vista allows remote attackers to execute arbitrary code or cause a denial of service (persistent reboot) via a large length value in the second (or later) 'anih' block of a RIFF .ANI, cur, or .ico file, which results in memory corruption when processing cursors, animated cursors, and icons. The .ani file was executed locally on Windows XP SP3.
6. CVE2010-0188 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0188
   A vulnerability in Adobe Reader and Acrobat 8.x before 8.2.1 and 9.x before 9.3.1 allows attackers to execute arbitrary code. PDF files were crafted and opened in Adobe Reader 8.2.0 locally.
7. CVE2010-1297 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-1297
   A vulnerability in Adobe Flash Player before 9.0.277.0 and 10.x before 10.1.53.64; Adobe AIR before 2.0.2.12610; and Adobe Reader and Acrobat 9.x before 9.3.3, and 8.x before 8.2.3 on Windows and Mac OS X, allow remote attackers to execute arbitrary code. A PDF file was opened locally with Adobe Reader 9.3.0 10.0.
8. CVE2011-2110 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2110
   A vulnerability in Adobe Flash Player before 10.3.181.26 on Windows allows remote attackers to execute arbitrary code. An SWF file was opened locally with Adobe Flash Player 10.0.
9. CVE 2012-0779 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0779
   A vulnerability in Adobe Flash Player before 10.3.183.19 and 11.x before 11.2.202.235 on Windows, Mac OS X, and Linux; before 11.1.111.9 on Android 2.x and 3.x; and before 11.1.115.8 on Android 4.x allows remote attackers to execute arbitrary code via a crafted file, related to an "object confusion vulnerability", as exploited in the wild in May 2012. An SWF file was opened locally with Adobe Flash Player 10.0 on Windows XP SP3.
10. CVE 2007-5659 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5659
    Multiple buffer overflows in Adobe Reader and Acrobat 8.1.1 and earlier allow remote attackers to execute arbitrary code via a PDF file with long arguments to unspecified

JavaScript methods. A PDF file was crafted and opened in Adobe Reader 8.1.1 locally on Windows XP SP3.

11. CVE 2010-2883 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2883
    A stack-based buffer overflow in CoolType.dll in Adobe Reader and Acrobat 9.x before 9.4, and 8.x before 8.2.5 on Windows and Mac OS X, allows remote attackers to execute arbitrary code or cause a denial of service (application crash) via a PDF document with a long field in a Smart INdependent Glyphlets (SING) table in a TTF font, as exploited in the wild in September 2010. A PDF file was crafted and opened in Adobe Reader 9.3.0 locally on Windows XP SP3.
12. CVE 2010-3653 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-3653
    The Director module (dirapi.dll) in Adobe Shockwave Player before 11.5.9.615 allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a Director movie with a crafted rcsL chunk containing a field whose value is used as a pointer offset, as exploited in the wild in October 2010. A crafted DIR file was opened with Shockwave Player 11.5.0.596 on Windows XP SP3.
13. CVE 2006-0476 See http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0476

    A Buffer overflow in Nullsoft Winamp 5.12 allows remote attackers to execute arbitrary code via a playlist (pls) file with a long file name (File1 field). A crafted PLS file was opened in WinAmp 5.12 locally.

In each case an OS and relevant target applications were used which were appropriate for the successful operation of the exploit under test.

Crafted samples were then placed on our web server and accessed via the indicated browser or were executed locally depending on the type of file. If our payload was executed, the system was considered compromised; if not, it was considered protected.

Static detect

There is also a way of testing which tries to avoid detection of all files and URLs which was also included in this study. This meant that neither files nor URLs and payloads should be detected during on-access or on-demand scanning, making it possible to determine the products' ability to combat previously unknown exploits. However, this procedure is so technically complex – and in some cases impossible – that we could not use it in all test cases. However, the signature detection of most vendors was excluded because the samples were generated by us.

Kaspersky AEP Test

Kaspersky Lab also asked to test how its new AEP technology copes with exploits without access to traditional detection methods. They offered a product testing methodology which enabled AEP but disabled rest of detecting techniques and checked the protection available at the moment the exploit began functioning. For KIS 2013 we performed the standard testing procedure, and an additional test with the following adjustments:

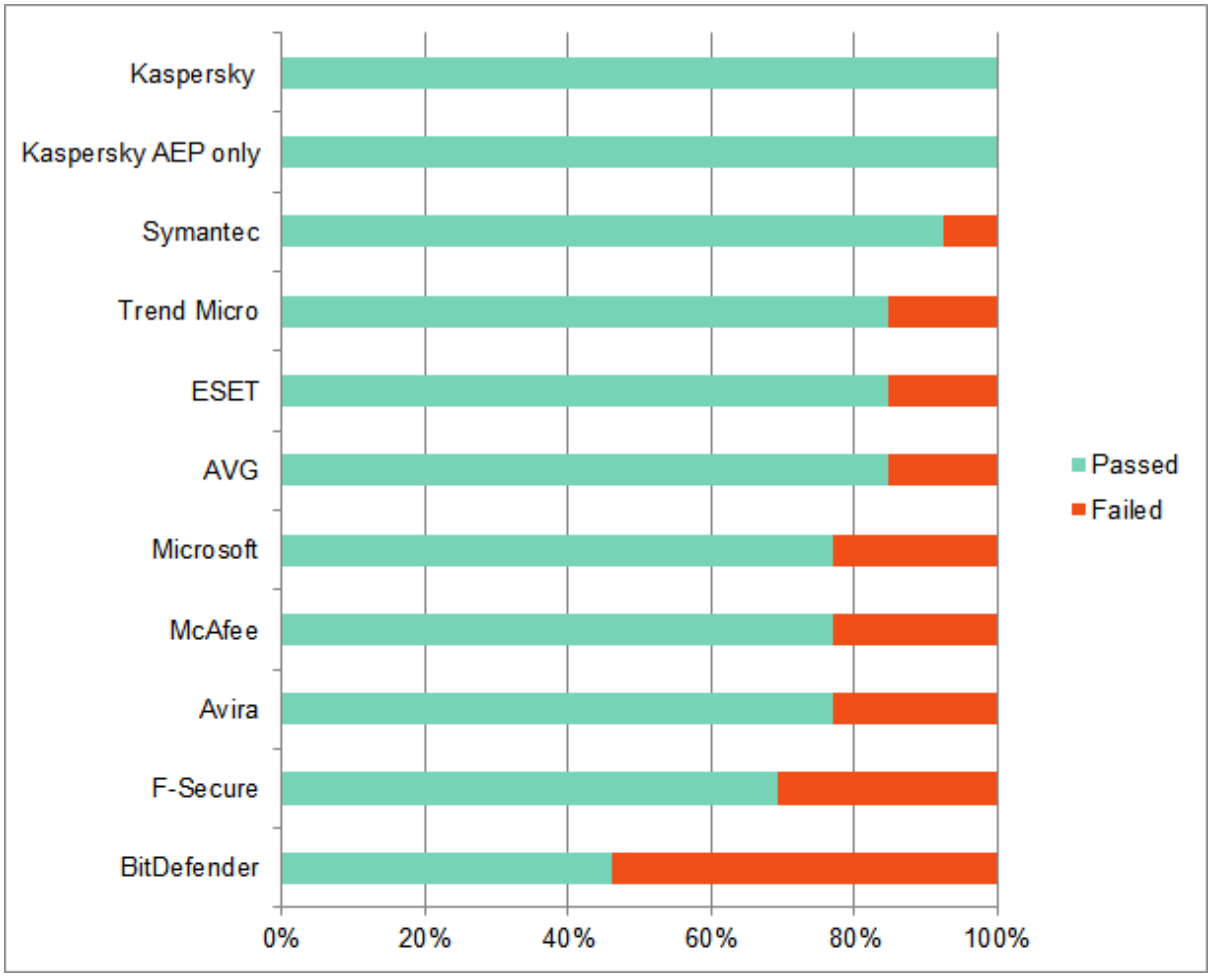
1. In the Settings menu, submenu File Anti-Virus we deactivated File Anti-Virus

2. In submenu Web Anti-Virus.
    a. Activate Web Anti-Virus
    b. Select Allow Download as action upon threat detection
3. Open Web Anti-Virus->Settings and deactivate blocking dangerous scripts in Microsoft Internet Explorer
4. Open the Application Control submenu and deactivate it
5. Open the System Watcher submenu and activate System Watcher.

These settings allow exploits to start running even if they are detected by other technologies, before testing how the new AEP technology can prevent the arbitrary execution of exploits. Once again the successful launch of a payload means that a product failed to protect against the exploit; otherwise, the system is considered protected.

Test Results

The table below shows the cumulative performance of each vendor when tested against the thirteen exploits.



The table below details the performance of each security application against each individual exploit:

	CVE-2012-0158	CVE-2012-0003	CVE-2012-1875	CVE2011-1260	CVE2007-0038	CVE-2010-0188	CVE-2010-1297	CVE-2011-2110	CVE2012-0779	CVE2007-5659	CVE2010-2883	CVE2010-3653	CVE2006-0476
AVG Internet Security 2012	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗
Avira Internet Security 2012	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗
Bitdefender Internet Security 2013	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗
Eset Smart Security V5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
F-Secure Internet Security 2012	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗
Kaspersky Internet Security 2013	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Kaspersky Internet Security 2013 AEP only	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
McAfee Total Protection 2012	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗
Microsoft Security Essentials V4	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Symantec Norton Internet Security 2012	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Trend Micro Titanium Maximum Security 2012	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓

Conclusions:

It should be borne in mind that modifications of the samples and replacement of the payload with the malicious simulator lead to malware itself in most cases would not be detected by signatures and so represent a zero day piece of malware, thus forcing the security applications to rely on detection or blocking of the actual exploit.

Protection against exploits is increasingly important as malware has a greater tendency to use them as an infection mechanism because they allow the silent and stealthy installation of malware on a system with no input from the user.

Testing using a combination of Metasploit and reverse engineering is a methodology which allows us to create conditions that map well to the real world, whilst allowing us to measure weather an exploit or subsequent payload has been blocked.

KIS 2013 successfully managed all samples within the uniform methodology for all products. Then we tested it according to the AEP test methodology which was described above. KIS 2013 again blocked all samples. Within the context of these tests, it is clear that the Kaspersky AEP technology was able to protect the system from the attacks used.

From the results of these tests, it would appear that the AEP technology provides a valuable layer of protection in the absence of blacklisting against new threats.