

# Protection against remote code execution exploits of popular applications in Windows

Jeffrey Wu  
PC Security Labs, China  
jeffrey@pcsecuritylabs.net

Anthony Arrott  
Security Analytics, Veszprog  
Labs, USA  
aarrott@gmail.com

Fernando C. Colón Osorio  
Wireless Systems Security  
Research Laboratory, USA  
fccco@wssrl.org

## Abstract

*The objective of Malicious Remote Code Execution Exploits is to remotely execute code transparently to the user, and without relying on user interaction, in order to infect targeted machines. This comparative study examines the effectiveness of different proactive exploit mitigation technologies included in popular endpoint security products and specialized anti-exploit tools. The study focuses on exploits of popular applications running on Windows XP SP3 with Internet Explorer (IE8). As such, the Microsoft Enhanced Mitigation Experience Toolkit (MS-EMET) is used as a reference standard for all exploit mitigation solutions.*

*The study compares the effectiveness of endpoint security products and anti-exploit tools by separating measurements of protections in common with MS-EMET from measures of protections supplemental to MS-EMET. This is done in order to understand not just the relative competitive effectiveness of the individual products and tools but also to understand the overall capabilities of the Windows endpoint security solutions to combat the remote code execution exploit capabilities of the overall Windows malware ecosystem.*

## 1. Introduction

In today's threat landscape, Exploit Kits and targeted attacks on home users and companies target popular applications such as browsers, browser add-ons (Flash, Silverlight), Java, Acrobat Reader, Microsoft Office Word, Excel, PowerPoint, media players, and so forth. The objective of the attack is to remotely execute code without the users knowledge, and without relying on user interaction, in order to infect the machine with undetected malware. This manuscript describes a comparative study, conducted by PC Security Labs whose sole goal was to evaluate security product effectiveness in minimizing the impact of such attacks.. Specifically, the effectiveness of proactive exploit mitigation technologies included by popular security vendors in

their end-point protection products, as well as that of commercially available stand-alone anti-exploit tools.

## 2. The Problem

In the past, several methods have been used to block malware infections resulting from vulnerability exploits. Amongst these, a pro-active patch strategy as applied by most vendors, has resulted in a dramatic reduction of remote exploit vulnerability infections. In addition, we have also seen the increased presence of exploit mitigation technologies as a native component of end-point security products.

In order to test the exploit blocking capabilities of such technologies, we used a Windows XP SP3 installation with IE8 and popular applications that are vulnerable to a number of exploits. Even though the test was performed under Windows XP SP3 it is worth noting that applications under test may still be vulnerable to exploitation under modern Operating Systems such as Windows 7 and Windows 8. In fact most of the exploits tested correspond to recently discovered vulnerabilities within the last two-(2) years.

The set of test described in this manuscript focus exclusively on the exploit blocking capabilities of the products tested, and the results obtained have no relevance when measuring the overall protection capabilities of the tested products.

The tests were commissioned by MalwareBytes Corp., with the sole goal of measuring the exploit blocking capabilities of different products against relevant vulnerabilities (i.e. vulnerable applications which are targeted typically by Exploit Kits and targeted attacks). The methodology, sample selection, specification of the system under test (SUT), as well as the Stymulus Workload (SW) were exclusively selected and decided by PCSL, and in such selection and design MalwareBytes, Inc. had no say whatsoever in the decisions made.

## 2. Methodology

In order to have a verifiable set of results, PCSL Laboratories has adopted the design test methodology first described in [1]. The methodology requires strict

and verifiable specification of four key components: (i) the system under test, (ii) the characterizing workload, formally name Steady State Workload (SSW), (iii) the Stymulus Workload (SW) and finally, (iii) a valid and agreed upon set of measures. Within the context of this methodology, PCSL Laboratories selected as the SUT an off-the-shelf Windows end-point running Windows XP SP3 with IE8. The characterizing workload consisted of several standard Windows un-patched applications that included but were not restricted to: Excel, Word, Adobe Acrobat Reader, and so forth. Specifically, PCSL used as the Steady State Workload (SSW) a combination of environments as described in [2] and [3], which mimic the behaviors of Internet users. The users fall into 7 broad categories. These are: Internet Addicts, Network Businessman, Socializer, Basic User, Gamer, Self-Presenter and Infrequent User. Based on these broad categories a Steady State synthetic workload was created reflecting the usage of each one of these groups.

As critical as the SUT and SSW specifications are, and in order to have a verifiable set of results, which are well received and acceptable within the Security Effectiveness Measurement Community & Ecosystem (SEMCE), the stimulus/workload used must be a reliable/good proxy for the actual environment that the products are expected to encounter in the wild. In [1] three key characteristics are described to accomplish this goal. These are: (i) **Timely**: exploits used to create the workload must represent as close as possible those found "in the wild" by typical users at the time of the test, (ii) **Prevalent** – they are in widespread use by the Malware Ecosystem (bad guys) and so represent the most common exploits faced in today's threat landscape, and (iii) **I&D- Innovative and Diverse**: they represent the threats faced by a rich variety of user environments, from freewheeling student dormitories to those found in hyper-secure financial services. In addition, they must employ the most innovative techniques at the forefront of *Malicious Remote Code Execution Exploits*.

In our experiment, the majority of the exploits were setup using Metasploit, see [3.] and a selective group came from Timely and Prevalent private sources. At no time, the vendor that commissioned the test were allowed to select or suggest any elements of this methodology or selected exploits. Each exploit was be tested with different payload configurations. Payloads range from execute, download and execute, reverse shells, and other options found in Metasploit.

In our experiment, it was critical that we tested solely the anti-exploit capabilities of the products

tested. Nevertheless, when end-point security products are tested, often, their signature detection capabilities is sufficient to detect and block exploits by simply matching file signatures. However, Malware writers are very good at signature obfuscation using metamorphic and polymorphic techniques, see [4]. Therefore, in order to solely test the anti-exploit capabilities of the products, the on access file detection of the products was disabled.

In table 1 and table 2, below, 58 variants of 21 known exploits used to test 12 endpoint security products and anti-exploit tools, including MS-EMET, are depicted. These 58 variants constituted the Stymulus Workload (SW), and the 12 end-point security products<sup>1</sup> are part of the SUT specification, namely the security protection product descriptions

## 2.1 The Metrics

The final element of the methodology described in [1] requires the strict specification of the measures or metrics to be used in evaluating the products. In our evaluation we consider a simple metric, that is, the percentage of anti-exploits that were successfully blocked by the anti-exploit mechanisms available in the products. A successful block was define as either:

#	exploit	variants	impact (CVSS v2)	public date	ref.
1	CVE-2014-0515	2	10.0 (High)	2014-04-29	[0]
Buffer overflow in Adobe Flash Player before 11.7.700.279 and 11.8.x through 13.0.x before 13.0.0.206 on Windows and OS X, and before 11.2.202.356 on Linux, allows remote attackers to execute arbitrary code via unspecified vectors, as exploited in the wild in April 2014.					
2	CVE-2014-0497	2	10.0 (High)	2014-02-05	[0]
Integer underflow in Adobe Flash Player before 11.7.700.261 and 11.8.x through 12.0.x before 12.0.0.44 on Windows and Mac OS X, and before 11.2.202.336 on Linux, allows remote attackers to execute arbitrary code via unspecified vectors.					
3	CVE-2013-3897	3	9.3 (High)	2013-10-09	[0]
Use-after-free vulnerability in the CDisplayPointer class in mshtml.dll in Microsoft Internet Explorer 6 through 11 allows remote attackers to execute					
4	CVE-2013-3346	3	10.0 (High)	2013-08-30	[0]
Adobe Reader and Acrobat 9.x before 9.5.5, 10.x before 10.1.7, and 11.x before 11.0.03 allow attackers to execute arbitrary code or cause a denial of service (memory corruption) via unspecified vectors.					
5	CVE-2013-3163	2	9.3 (High)	2013-07-09	[0]
Microsoft Internet Explorer 8 through 10 allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a crafted web site, aka "Internet Explorer Memory Corruption Vulnerability".					

Table 1a. Stimulus Workload (SW). Remote code execution exploits used in the testing (1-5)

<sup>1</sup> SurfRight contacted PCSL and requested it be clarified that the version of HitmanPro.Alert3 CTP2 which was tested as part of this evaluation was a pre-release version. As there are still updates to be made to the software before it is deemed as final this may have had a bearing on its test performance.

#	exploit	variants	impact (CVSS v2)	public date	ref.
6	CVE-2013-2465	3	10.0 (High)	2013-06-18	[0]
Unspecified vulnerability in the Java Runtime Environment (JRE) component in Oracle Java SE 7 Update 21 and earlier, 6 Update 45 and earlier, and 5.0 Update 45 and earlier allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors related to 2D.					
7	CVE-2013-2460	3	9.3 (High)	2013-06-18	[0]
Unspecified vulnerability in the Java Runtime Environment (JRE) component in Oracle Java SE 7 Update 21 and earlier, and OpenJDK 7, allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors related to Serviceability. NOTE: the previous information is from the June 2013 CPU. Oracle has not commented on claims from another vendor that this issue allows remote attackers to bypass the Java sandbox via vectors related to "insufficient access checks" in the tracing component.					
8	CVE-2013-2423	3	4.3 (Medium)	2013-04-17	[0]
Unspecified vulnerability in the Java Runtime Environment (JRE) component in Oracle Java SE 7 Update 17 and earlier, and OpenJDK 7, allows remote attackers to affect integrity via unknown vectors related to HotSpot. NOTE: the previous information is from the April 2013 CPU. Oracle has not commented on claims from the original researcher that this vulnerability allows remote attackers to bypass permission checks by the MethodHandles method and modify arbitrary public final fields using reflection and type confusion, as demonstrated using integer and double fields to disable the security manager.					
9	CVE-2013-1488	9	CVE-2013-1488	9	CVE-2013-1488
The Java Runtime Environment (JRE) component in Oracle Java SE 7 Update 17 and earlier, and OpenJDK 6 and 7, allows remote attackers to execute arbitrary code via unspecified vectors involving reflection, Libraries, "improper toString calls," and the JDBC driver manager, as demonstrated by James Forshaw during a Pwn2Own competition at CanSecWest 2013.					
10	CVE-2013-1347	10	CVE-2013-1347	10	CVE-2013-1347
Microsoft Internet Explorer 8 does not properly handle objects in memory, which allows remote attackers to execute arbitrary code by accessing an object that (1) was not properly allocated or (2) is deleted, as exploited in the wild in May 2013.					
11	CVE-2013-1017	11	CVE-2013-1017	11	CVE-2013-1017
Buffer overflow in Apple QuickTime before 7.7.4 allows remote attackers to execute arbitrary code or cause a denial of service (application crash) via crafted dref atoms in a movie file.					
12	CVE-2013-0634	12	CVE-2013-0634	12	CVE-2013-0634
Adobe Flash Player before 10.3.183.51 and 11.x before 11.5.502.149 on Windows and Mac OS X, before 10.3.183.51 and 11.x before 11.2.202.262 on Linux, before 11.1.111.32 on Android 2.x and 3.x, and before 11.1.115.37 on Android 4.x allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via crafted SWF content, as exploited in the wild in February 2013.					

Table 1b. Stimulus Workload (SW). Remote code execution exploits used in the testing (6-12)

#	exploit	variants	impact (CVSS v2)	public date	ref.
13	CVE-2013-0074	3	9.3 (High)	2013-03-12	[0]
Microsoft Silverlight 5, and 5 Developer Runtime, before 5.1.20125.0 does not properly validate pointers during HTML object rendering, which allows remote attackers to execute arbitrary code via a crafted Silverlight application, aka "Silverlight Double Dereference Vulnerability."					
14	CVE-2012-4969	3	9.3 (High)	2012-09-18	[0]
Use-after-free vulnerability in the CMShtmlEd::Exec function in mshtml.dll in Microsoft Internet Explorer 6 through 9 allows remote attackers to execute arbitrary code via a crafted web site, as exploited in the wild in September 2012.					
15	CVE-2012-4792	3	9.3 (High)	2012-12-30	[0]
Use-after-free vulnerability in Microsoft Internet Explorer 6 through 8 allows remote attackers to execute arbitrary code via a crafted web site that triggers access to an object that (1) was not properly allocated or (2) is deleted, as demonstrated by a CDwnBindInfo object, and exploited in the wild in December 2012.					
16	CVE-2012-1856	1	9.3 (High)	2012-08-14	[0]
The TabStrip ActiveX control in the Common Controls in MSCOMCTL.OCX in Microsoft Office 2003 SP3, Office 2003 Web Components SP3, Office 2007 SP2 and SP3, Office 2010 SP1, SQL Server 2000 SP4, SQL Server 2005 SP4, SQL Server 2008 SP2, SP3, R2, R2 SP1, and R2 SP2, Commerce Server 2002 SP4, Commerce Server 2007 SP2, Commerce Server 2009 Gold and R2, Host Integration Server 2004 SP1, Visual FoxPro 8.0 SP1, Visual FoxPro 9.0 SP2, and Visual Basic 6.0 Runtime allows remote attackers to execute arbitrary code via a crafted (1) document or (2) web page that triggers system-state corruption, aka "MSCOMCTL.OCX RCE Vulnerability."					
17	CVE-2012-1535	3	9.3 (High)	2012-08-15	[0]
Unspecified vulnerability in Adobe Flash Player before 11.3.300.271 on Windows and Mac OS X and before 11.2.202.238 on Linux allows remote attackers to execute arbitrary code or cause a denial of service (application crash) via crafted SWF content, as exploited in the wild in August 2012 with SWF content in a Word document.					
18	CVE-2012-0663	3		1900-01-03	[0]
19	CVE-2012-0507	3	10.0 (High)	2012-06-07	[0]
Unspecified vulnerability in the Java Runtime Environment (JRE) component in Oracle Java SE 7 Update 2 and earlier, 6 Update 30 and earlier, and 5.0 Update 33 and earlier allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors related to Concurrency. NOTE: the previous information was obtained from the February 2012 Oracle CPU. Oracle has not commented on claims from a downstream vendor and third party researchers that this issue occurs because the AtomicReferenceArray class implementation does not ensure that the array is of the Object[] type, which allows attackers to cause a denial of service (JVM crash) or bypass Java sandbox restrictions. NOTE: this issue was originally mapped to CVE-2011-3571, but that identifier was already assigned to a different issue.					
20	CVE-2012-0158	3	9.3 (High)	2012-04-10	[0]
The (1) ListView, (2) ListView2, (3) TreeView, and (4) TreeView2 ActiveX controls in MSCOMCTL.OCX in the Common Controls in Microsoft Office 2003 SP3,					
21	CVE-2011-2110	3		1900-01-03	[0]

Table 1c. Stimulus Workload (SW). Remote code execution exploits used in the testing (12-21)

Vendor	Product	Version
AVAST	AVAST Internet Security	2014.9.0.2021
AVG	AVG Internet Security	14.0.0.4744
Bitdefender	Bitdefender Internet Security	17.28.0.1191
Enhanced	Enhanced Mitigation Experience Toolkit 2	4.1.5228.513
ESET	ESET Smart Security	7.0.317.4
Kaspersky	Kaspersky Internet Security	14.0.0.4651(g)
Malwarebytes	Malwarebytes Anti-Exploit Premium 4	1.04.1.1006
McAfee	McAfee Internet Security	12.8.958
Panda	Panda Internet Security	19.01.01
SurfRight	HitmanPro.Alert 3 CTP23	3.0.12.73
Symantec	Norton Internet Security	21.4.0.13
Trend Micro	Titanium Maximum Security	7.0.1255

Table 2. Systems under test (SUTs).  
Commercially-available security products with  
anti-exploit capability

(i) the product under evaluation successfully detected the exploit and prevented the payload from being executed, or (ii) the payload was executed but the security product use some method(s) to shut down the backdoor connection after the payload was executed.

As described earlier, under the description of the SUT, all the tests were executed on end-points running the Windows XP SP3 Operating System (English), without any other additional patches. In addition, all tested security products described in Table 2 were obtained by downloading the latest version of the corresponding security product from the vendors official website, without exception.

### 3. Experimental Results

In this study, 58 variants of 21 known exploits were used to test 12 endpoint security products and anti-exploit tools, including *Microsoft Enhanced Mitigation Experience Toolkit (MS-EMET)*. As shown in Figure 1, the Microsoft MS-EMET stand alone solution was able to block 74% of all the exploit variants considered, while a third party stand-alone anti-exploit product showed the best performance by blocking 93% of all exploits considered. Figure 1 in combination with Figure 2 shows an interesting phenomena, mainly, the combined capabilities of Microsoft MS-EMET and those of 3rd party vendors (at least 4 such vendors) resulted in the best performance of anti-exploit solutions blocking 100% of all threats. For example, in spite the fact that MalwareBytes provided the stand-alone best overall protection, Figure 1, the absolute best protection was obtained by a combination of MS-EMET and one of four stand-alone solutions that did not include MalwareBytes. This fact begs the question of relative performance enhancement testing in the Microsoft Windows anti-

Malware ecosystem. Meaning, it is of considerable advantage to the industry and users when considering purchasing decisions to evaluate anti-Malware products by contrasting their stand-alone capabilities to that of Microsoft. If such products result only in marginal improvements, then, the users must ask the question - why bother?

Figure 3, 4, 5 and 6 present a complete picture in terms of redundancy and completeness amongst all anti-exploit products. It is clear from Figure 3 that while none of the exploit variants was blocked by less than 2 products, and none were blocked by all the products, more than 80% of the exploit variants were blocked by between 4 and 9 of the 12 products and tools tested. Basically, all products do a good job on detecting and blocking a large number of exploit variants. In addition, Figure 6 illustrates the amount of redundancy between stand-alone anti-Exploit products and MS-EMET. Notice from Figure 6 that the redundancy index (meaning the % of exploits that we commonly blocked by MS-EMET and the 3rd party product) reaches a high point of 67%. This combined the enhancement score presented in Figure 5, 6, 7, and 8 will argue for product diversity in your protection strategies.

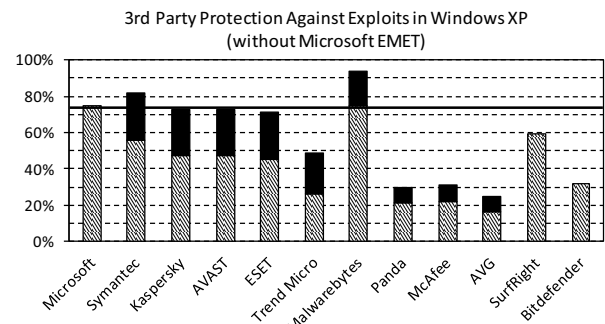


Figure 1. all products.

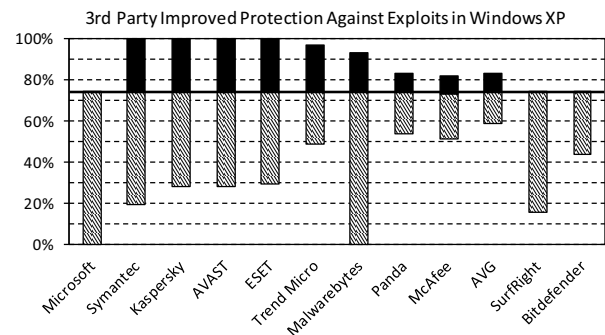


Figure 2. Combined MS-EMET & 3rd Party solutions

Finally, as a word of caution, this manuscript catalogued all anti-Exploit products into four-(4) broad categories. There are: products which are only able to block less than 60% of the tested exploits are catalogued as ***Failed*** in terms of their exploit blocking claims. Products which are able to block between 61% and 80% of all exploits tested are considered to be ***Insufficient*** to provide adequate protection. Further, products which are able to block over 80% of all tested exploits are considered to be ***Effective***.

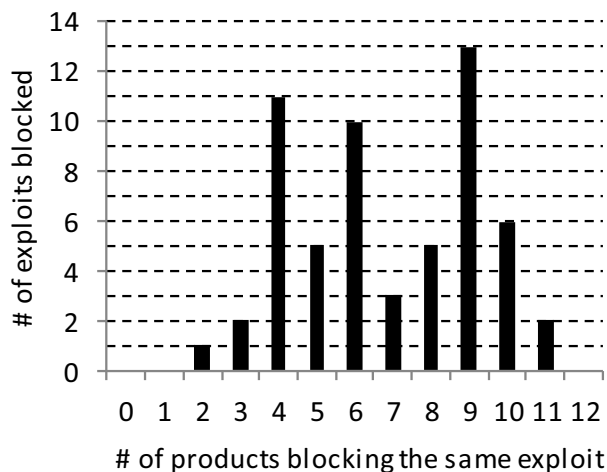


Figure 3. Diversity and redundancy of commercially-available anti-exploit products.

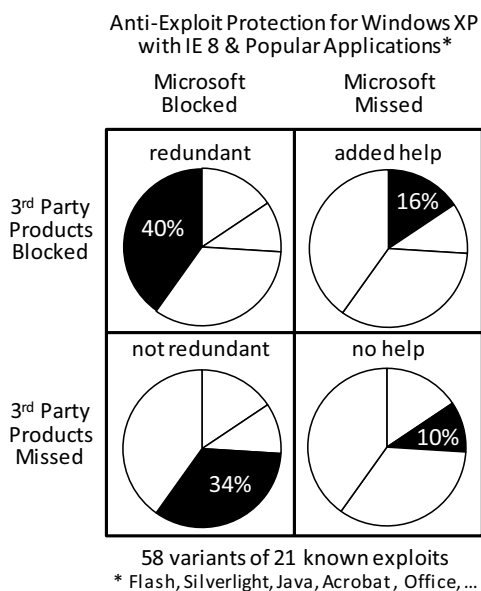


Figure 4. Coverage Analysis of anti-Exploit Products

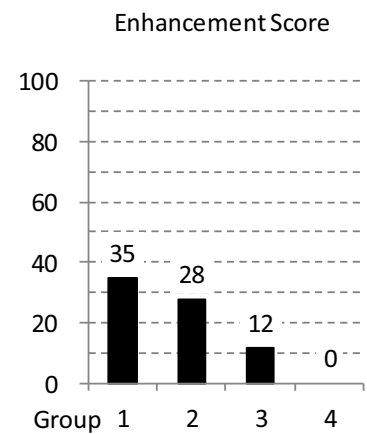


Figure 5. Enhancement score

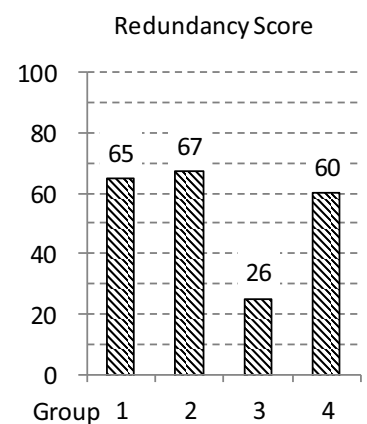


Figure 6. Redundancy score

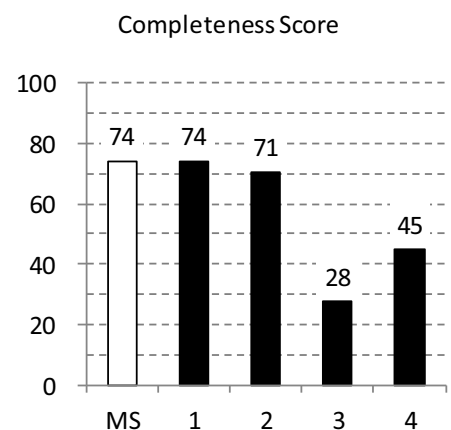


Figure 7. Completeness score.

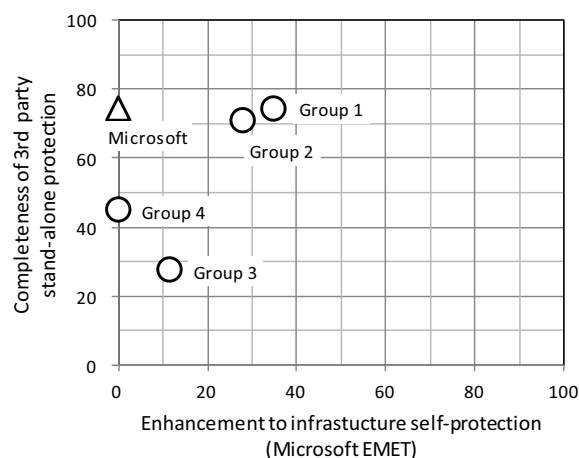


Figure 8. Relative enhancement to MS-EMET by 3rd Party products by Group

#### 4. Conclusions & Recommendations

In this study, 58 variants of 21 known exploits were used to test 12 endpoint security products and anti-exploit tools, including MS-EMET. While the MS-EMET stand alone solution was able to block 74% of the exploit variants considered, and one of the 3rd party anti-exploit tools was able to block 93% of said variants, the combined capabilities of all the products and tools were able to block 100% of the threats. In terms of the Windows anti-Exploit ecosystem, it is worth noting that while none of the exploit variants was blocked by less than 2 products and none were blocked by all the products, more than 80% of the exploit variants were blocked by between 4 and 9 of the 12 products and tools tested. Further, four-(4) products provided perfect 100% supplemental protection to MS-EMET. However, in a stand-alone mode (end protection in isolation without the benefits of MS-EMET) none of these 4 products blocked a higher percentage of the exploit variants than MS-EMET.

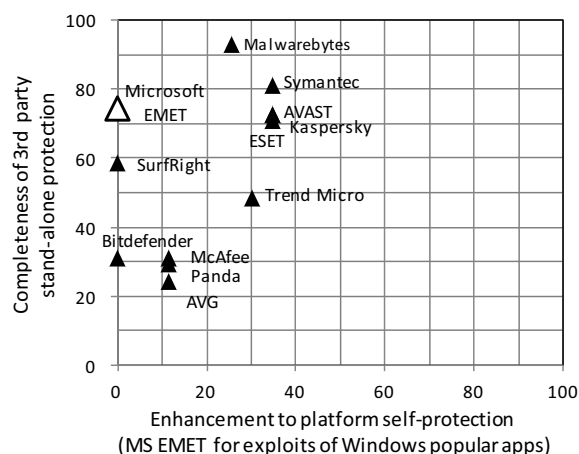


Figure 9. Relative enhancement to MS-EMET by 3rd Party products by stand-alone Product

#### 5. References

- [1] Fernando C. Colon Osorio, et. al., "Measuring the effectiveness of modern security products to detect and contain emerging threats — A consensus-based approach. ... as a measure", 8th International Conference on Malicious and Unwanted Software: "The Americas", MALWARE 2013, Fajardo, PR, USA, October 22-24, 2013. IEEE 2013 ISBN 978-1-4799-2534-6 IEEE 2013 ISBN 978-1-4799-2534-6
- [2] PC Security Labs (PCSL), Security solution review on Windows 8 platform. February 2013. [http://www.pcsecuritylabs.net/document/report/pcsl\\_win8\\_security\\_solution\\_review\\_201302.pdf](http://www.pcsecuritylabs.net/document/report/pcsl_win8_security_solution_review_201302.pdf)
- [3] China Internet Network Information Center (CNNIC), Statistical report on Internet development in China. January 2012. <http://www1.cnnic.cn/IDR/ReportDownloads/201302/P020130221391269963814.pdf>
- [4] Maynor, David, "Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research", Elsevier, 2011, ISBN-10: 1597490741
- [5] Hongyuan Qiu and Fernando C. Colon Osorio, "Static Malware detection with Segmented Sandboxing", 8th International Conference on Malicious and Unwanted Software: "The Americas" (MALWARE), 22-24 Oct. 2013