Corporate AV/EPP Comparative Analysis: Exploit Protection (PDF)

RebootTwice LLC
Phone: 949-831-8821
mmcrae@reboottwice.com

CORPORATE AV / EPP COMPARATIVE ANALYSIS

Exploit Protection

2013 – Randy Abrams, Dipti Ghimire, Joshua Smith

Tested Vendors

AVG, ESET, F-Secure, Kaspersky, McAfee, Microsoft, Norman, Panda, Sophos, Symantec, Trend Micro

Overview

Endpoint Protection Products (EPP) are designed to protect against a broad spectrum of threats. Products originally developed to detect self-replicating code (viruses and worms) have added protection against adware, spyware, rootkits, bootkits, phishing attacks, and exploits, in addition to providing firewall capabilities and more.

The ability to block exploits is one of the most significant tasks required of EPP products. When a new vulnerability is exploited, not only can malware, known or unknown, be silently installed, criminals can take over the exploited computer manually, thereby evading signatures and heuristics designed to detect malicious code. If an EPP can block an exploit, it has effectively blocked any and all malware that the exploit may attempt to execute or install. The ability to catch the payload an exploit delivers has value but provides far less protection than blocking the exploit itself.

Exploit kits such as Blackhole have essentially made the mass exploitation of websites a low cost franchise operation with a low buy-in and an immediate lucrative return. Software such as Oracle's Java, Adobe's Flash and Reader/Acrobat, in addition to web browsers, keep a fresh supply of exploitable vulnerabilities available even as old exploits continue to plague consumers and corporations alike.

The exploitation of vulnerabilities in common software programs enables attackers to breach networks, steal intellectual property, hijack email and social network accounts, and engaging in several other types cybercrimes. NSS vulnerability research reveals that the number of reported vulnerabilities rose significantly in 2012 and the vulnerability landscape is going through significant transformations[1].

Enterprises have several tools to help prevent the exploitation of vulnerabilities. Patching is one of the most important defenses. However many corporations fail to patch all of the applications on their desktops and often are slow to deploy the most current software versions. Intrusion prevention systems (IPS), and in some scenarios next generation firewalls (NGFW), can provide a valuable line of defense against exploits for enterprises. NSS provides extensive comparative testing for IPS and NGFW products. The use of current web browsers is another line of defense. The most widely used browsers have added features such as reputation systems and application blocking to help defend against the exploitation of vulnerabilities. The use of endpoint protection products, colloquially known as antivirus, is also a common defense.

NSS tested 11 popular enterprise EPP products to measure their effectiveness in protecting Windows computers against exploits. All of the exploits used during this test have been publicly available for months (and sometimes years) prior to the test, and have also been observed in use on the Internet.

Enterprises, especially those employing the BYOD model, that seek protection from exploit driven attacks against desktop PCs and laptops should closely examine results from this test.

[1] https://www.nsslabs.com/reports/vulnerability-threat-trends

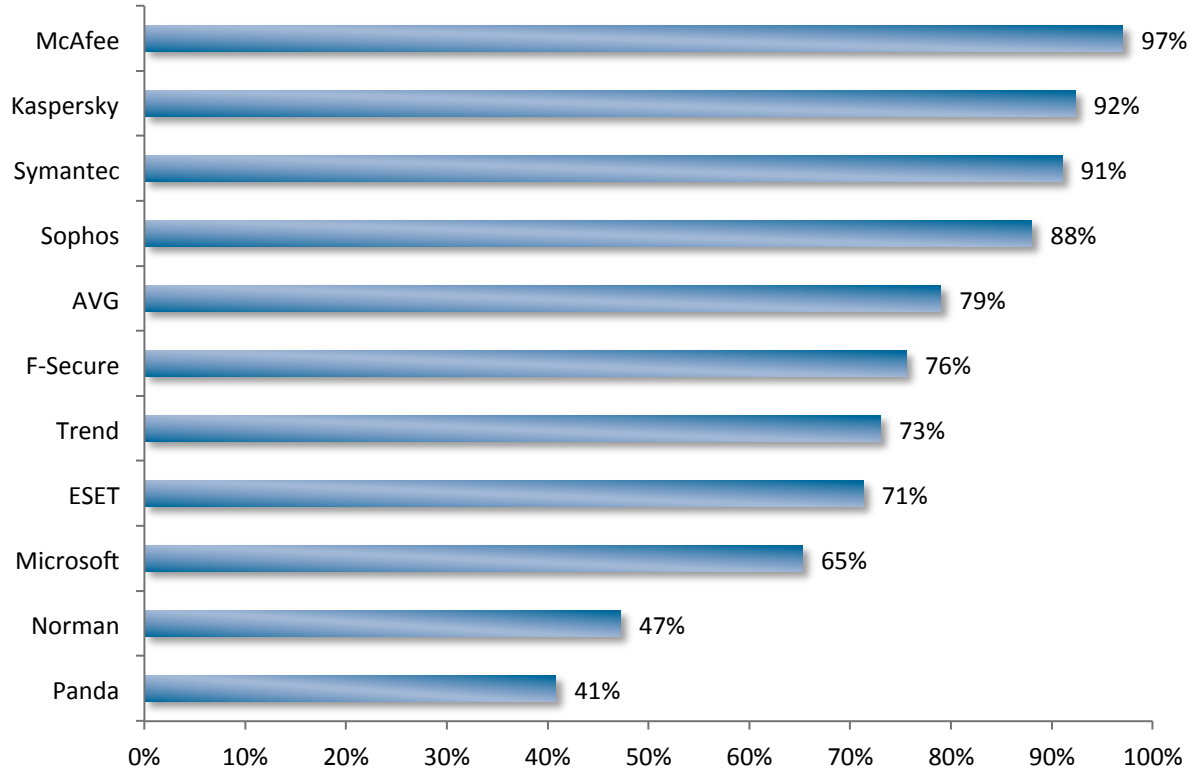


Figure 1 - Combined Block Rates (including alternate vectors)

Figure 1 combines 203 exploit download and payload execution tests with 30 alternate vector attacks to provide the overall exploit protection rate for the tested EPP products.

Key Findings:

- With a few notable exceptions, endpoint products are not providing adequate protection from exploits.
- Enterprise EPP products differ up to 53% in effectiveness at blocking exploits, with protection levels varying between 44% and 97%
- Keeping AV software up-to-date does not yield adequate protection against exploits, as evidenced by gaps in coverage for vulnerabilities found to be several years old.
- Java is a significant attack vector

Table of Contents

Table of Figures

Analysis

The results of NSS' in-depth testing of 41 individual exploits and over 200 attack scenarios revealed significant differences in the defensive capabilities of 11 leading endpoint protection solutions. Results are provided for exploits that require Internet Explorer 6 and those that do not. Given that many enterprises are forced to support IE6 because of legacy applications, this capability may be a determining factor in selecting an EPP product.

Excluding exploits requiring IE6, the average block rate was 77%, with the weakest product blocking 44% and the best product blocking 98% of the attacks. For exploits requiring IE6 to execute, the average blocking ability was 65%, with the weakest performer blocking 20% of the attacks and the top products blocking 100% of the attacks.

Enterprises rely on endpoint security products to help provide a virtual shield against exploits. The number of potentially vulnerable applications that need to be patched taxes the resources of most IT departments and may allow vulnerabilities to persist longer than they ordinarily might on a consumer computer. NSS testing shows that the majority of EPP products fail to block some of the most widely used and dangerous exploits from recent years.

Given the importance and growing prevalence of this class of threat, NSS recommends that enterprises give appropriate weight to the quality of exploit prevention technology, as well as performance and threat detection, when selecting EPP products.

Test Background – Threat Landscape

The layers of defense used in enterprises vary widely. The extent to which technologies such as IPS, NGFW, web and application whitelisting, thin clients, and other measures are employed will affect how critical it is that an EPP product is capable of blocking exploits. Where employees work from home, or the "bring your own device" (BYOD) model is adopted, the importance of exploit prevention in EPP products may be significantly increased.

Exploit detection and prevention is a difficult problem and requires a different set of skills and focus than traditional malware protection.

In this test NSS demonstrates the capabilities of 11 popular enterprise-level endpoint protection products.

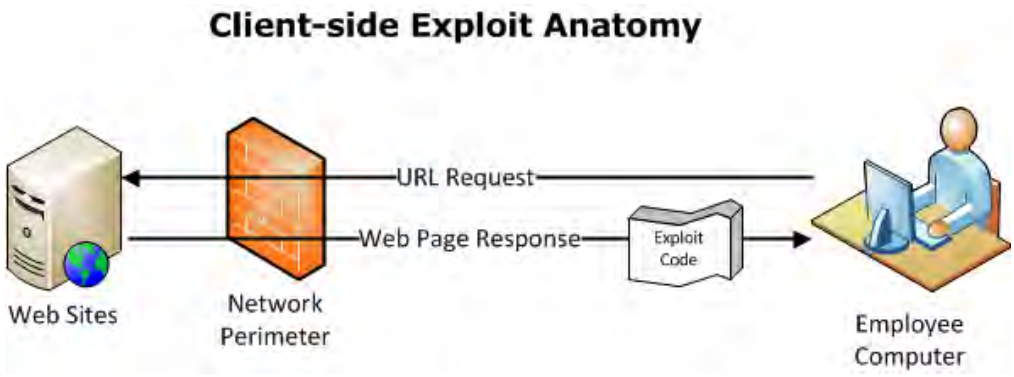


Figure 2 - How a desktop/laptop computer is exploited

Stages of Protection

The following table outlines pros and cons of stopping the threat at the various stages.

Stage of Protection	Pros	Cons
Stage Ø **Vulnerability**	*Provides the best protection—prevents the vulnerability from triggering* **90% proactive:** Can develop protection before exploits based upon the vulnerability are released ALL alternate exploit variants of the vulnerability are blocked Nearly impossible to evade Very accurate Generates the least false positives	*Requires a lot of work and is hard to do* **10% reactive:** Must know vulnerability Requires complex application or protocol decoding Must understand the vulnerability Most processor-intensive
Stage 1 **Exploit**	*Offers targeted protection—prevents the (known) exploit* No need to understand the vulnerability or the protocol beyond a cursory level Can be done easily through regular expression matching Fast Generates few false positives	*Provides limited targeted protection* **50% reactive:** Must see the exploit first Only prevents the specific (known) exploit Easy for attackers to find alternatives to bypass Maximum coverage = many signatures Requires tuning to prevent false positives
Stage 2 **Payload**	*Focuses on the malicious payload (malware)* Detects malware that is delivered by other means (i.e. USB) Simple pattern matching Fast Based on mature technology	*Detection occurs after a successful attack has put malicious code on an endpoint* **100% reactive**: Must see the payload first Does not detect "non-standard" attacks Easy for attackers to obfuscate attacks and bypass Requires the most signatures + constant updates to be effective Only provides limited protection

How This Test Was Conducted

Between October and December 2012, NSS tested 11 enterprise endpoint protection products, assessing their respective protection capabilities against exploits. Vulnerabilities used in this test were exploited when a user visited an infected web page hosting the attack code. The attacks occurred in two stages:

1. The attacker caused a specially crafted stream of data and code to be delivered to a precise location. This exploited the victim's computer, gaining the attacker the ability to perform arbitrary code execution.
2. Malicious code was silently executed on the victim's computer.

If the attack can be thwarted in stage one (successful exploit), then it cannot progress to stage two. As long as the exploit is not defeated, then installing malware is just one of many possible actions the attacker can take. Prior to exploiting a vulnerability, attackers have the ability to use services such as Google's VirusTotal and even the products themselves, to ensure the payload will not be detected by any antivirus product. Since cybercriminals have the time and resources to ensure custom malware will go undetected, it is imperative that attacks be defeated in the earliest possible stage. Those products that are unable to prevent the exploitation of vulnerabilities are also unable to provide significant protection against the infinite number of payloads that can be delivered.

Protection From Exploits Across Protocols

The Firefox add-on, Firesheep, brought substantial media attention to session hijacking attacks, and forced many social media sites to implement encrypted sessions. Today, Gmail, Twitter, and Facebook all offer end-to-end HTTPS sessions, as does virtually every financial site. When trusted SSL sites are compromised, products that cannot penetrate SSL encryption are blind to the attacks and to the malware being delivered through the HTTPS transport protocol. Detection of exploits delivered across HTTP versus HTTPS protocols can vary by as much a 39% in a single product.

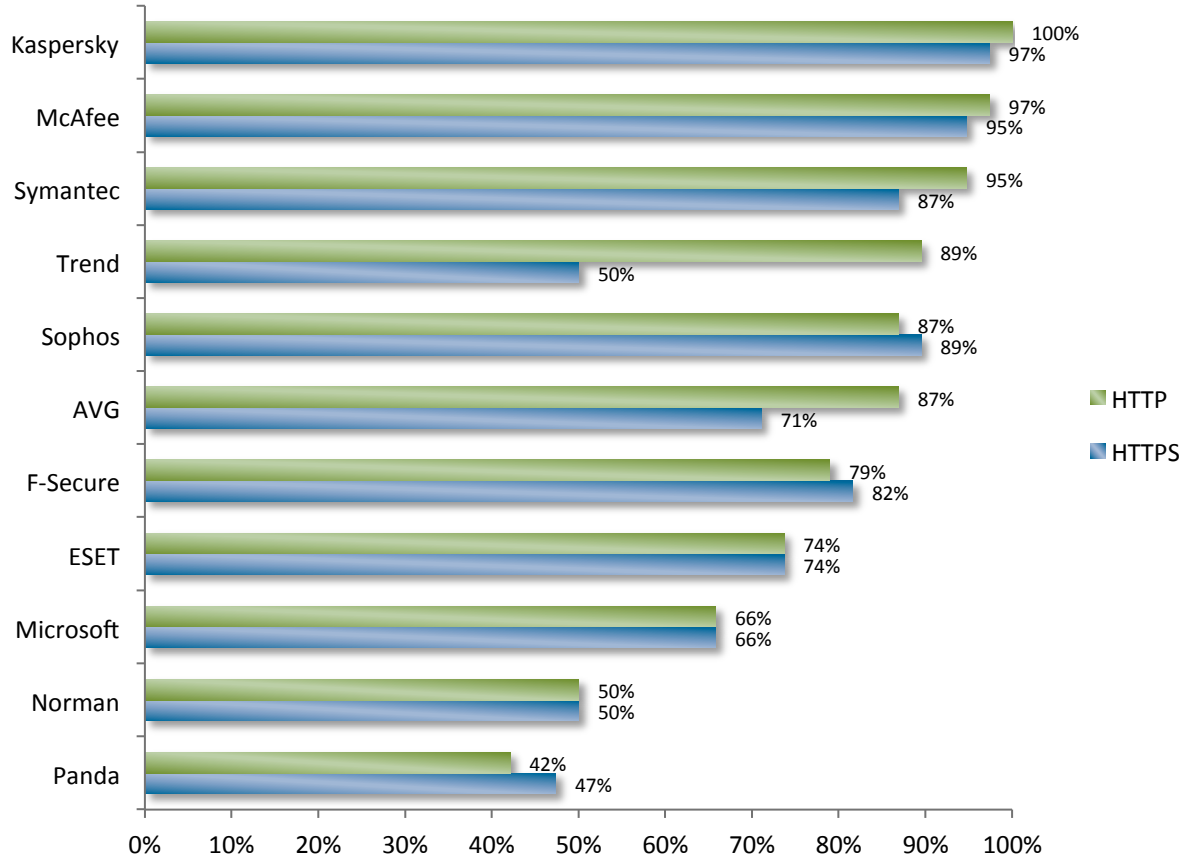


Figure 3 - HTTP vs. HTTPS block rates

NSS protocol testing utilized a payload that was proven to be detectable by all products in at least some cases. The payload was delivered via both HTTP and HTTPS leveraging 39 different exploits. Browsers used in the protocol testing included multiple versions of Microsoft Internet Explorer, Mozilla Firefox, Apple Safari, and Google Chrome. Vulnerable applications included versions of .NET, Flash, Java, Office, Shockwave, RealPlayer, Reader, QuickTime, WMItools, and WMP. To provide the best protection, security products should ideally protect against all exploits for a given vulnerability, regardless of transport protocol.

Exploit Blocking Results

In the non-IE6 tests, no product was able to block all of the exploits, and only three products, Kaspersky (98%), McAfee (96%) and Symantec (92%) were able to block more than 90% of the exploits. Four products, ESET (74%), Microsoft (66%), Norman (52%), and Panda 44% failed to block at least 75% of the exploits.

For most products, the problem was not whether or not the traffic was encrypted, but rather a failure to detect exploits at all (over both HTTP and HTTPS). A few products even demonstrated more effective exploit blocking performance over HTTPS than over HTTP. On average, there was a 7% difference in the ability of products across the board to block HTTPS versus HTTP exploit attacks.

Trend Micro blocked 39% fewer attacks delivered via HTTPS than through HTTP and AVG blocked 16% fewer attacks when SSL was used. Only ESET, Microsoft, and Norman consistently blocked the same attacks delivered through HTTPS as they did when SSL was not used.

The overall effectiveness of the 11 products in blocking (non-IE6) exploits is as follows:

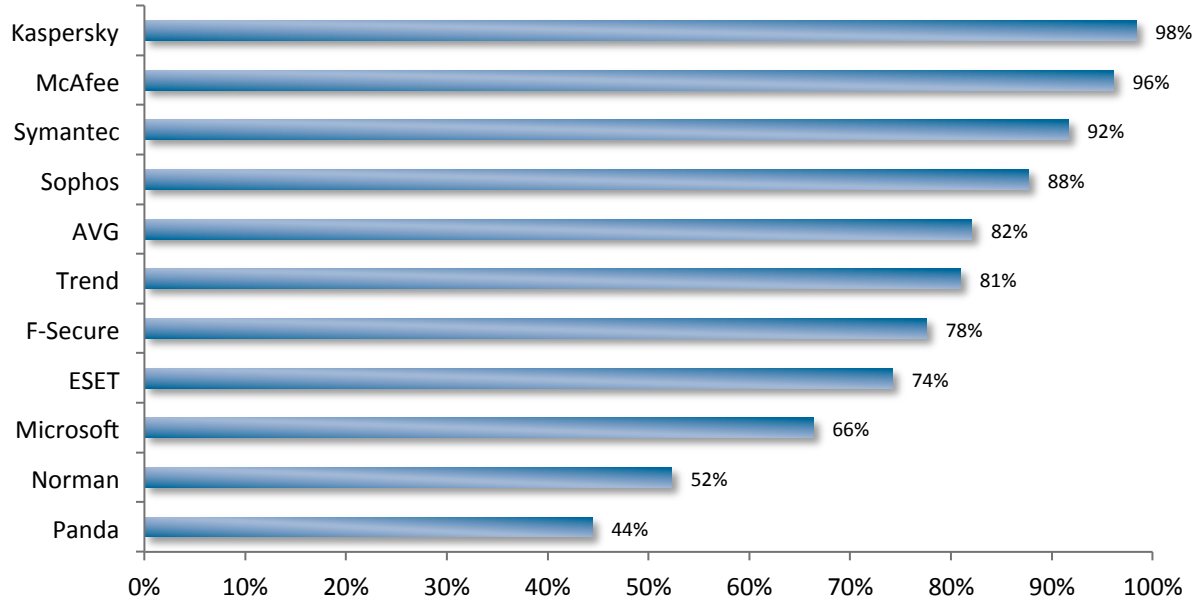


Figure 4 - Non- IE6 Overall Exploit Block Rate

When testing protection against exploits that require the use of Internet Explorer 6.0, three products, McAfee, Sophos, and Symantec were able to block 100% of the exploits. The average detection of these exploits was 65%. Among the IE6 driven attacks that Microsoft failed to block was an exploit that affects Microsoft Office 2003. There were 5 products that able to block more than 75% of the attacks. Five products failed to block 50% of the exploits that affect IE6 uses.

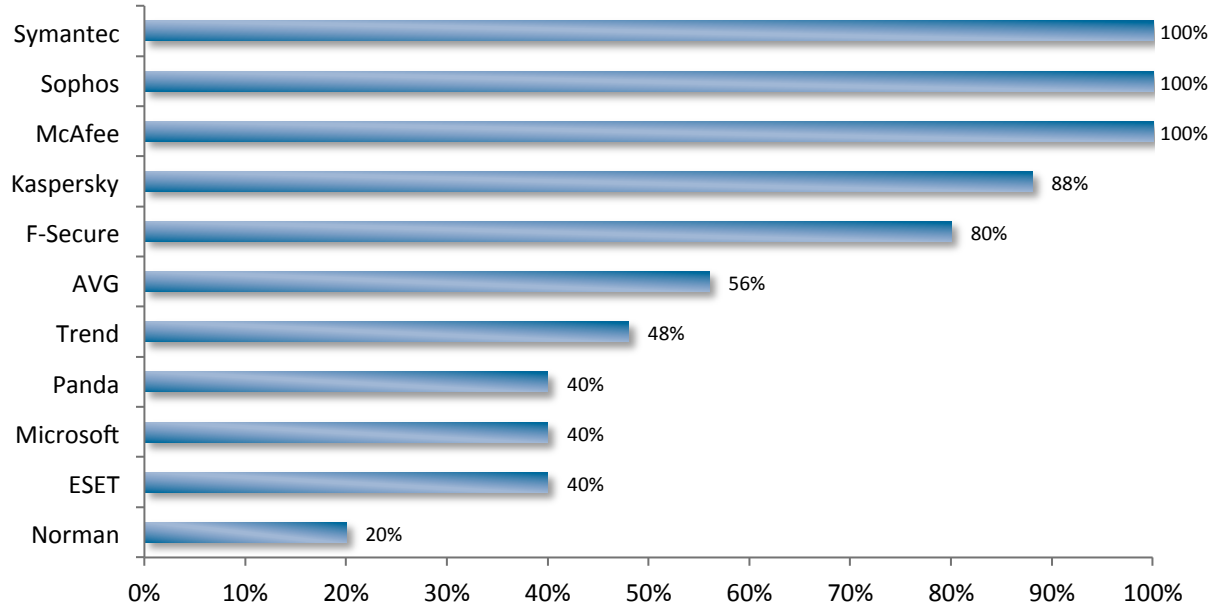


Figure 5 - IE6 Overall Block Rate

The combined exploit protection of figures 4 and 5 are shown in figure 6 below.

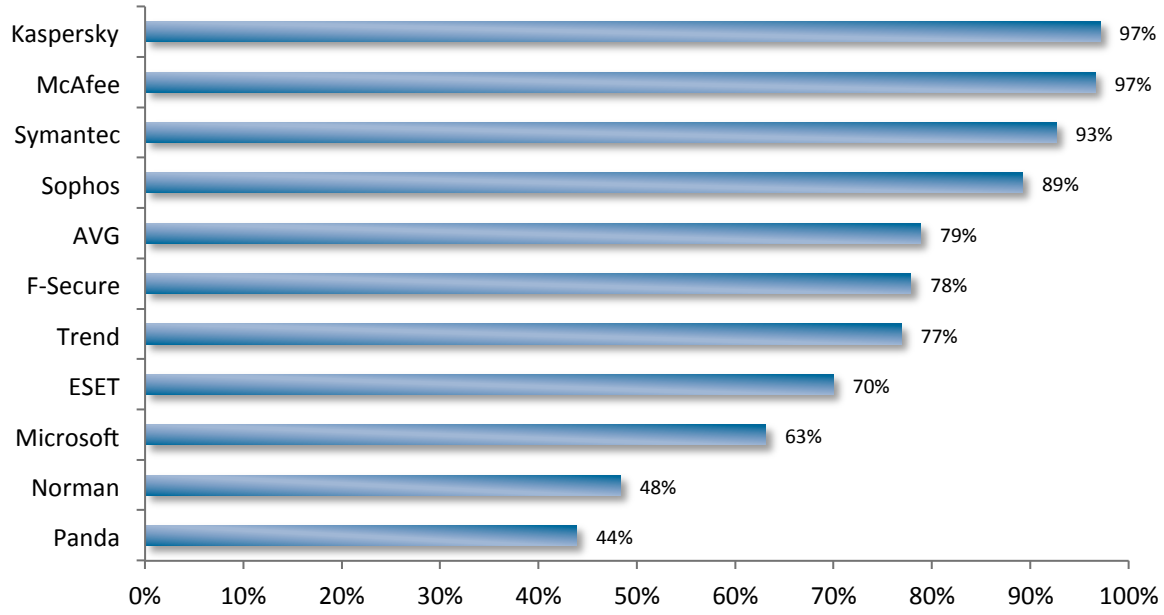


Figure 6 - Overall Exploit Block Rate

Alternative Attack Vectors

In testing EPP products against exploits the primary tests were performed using a variety of web browsers. NSS engineers also performed a few tests using alternate attack vectors. However the sample set was too small to present qualitative product differences on those criteria alone, but did reveal some gaps in protection.

NSS engineers tested 5 exploits, each executed from an Outlook email message, executed from a network share, and copied from a network share and executed from the desktop. Most products would block the exploits when delivered from alternate vectors if they blocked the exploit on download. There were a couple of notable and interesting exceptions, however.

For one exploit Kaspersky failed to block an exploit when run from alternate vectors. On further inspection it was determined that the initial block when a browser was used was based upon a heuristic that detected the exploit script rather than the exploit itself. F-Secure failed to block two exploits if they were executed from a network share but detected the exploits when downloaded, opened from email or opened from the desktop. Trend Micro blocked an exploit on download and when opened from Outlook, but not when executed from a network share or the desktop. The small sample set precludes conclusions that the other products would not have similar issues if a statistically significant sample set were used in these tests. However the testing does conclusively demonstrate that the ability to block an exploit on download does not automatically translate to protection against alternate delivery methods.

NSS engineers noted other disconcerting behaviors while conducting the tests. There were several instances where products flagged an exploit but the payload was still executed. These cases were tabulated as failures. Initial indications point to a probable race condition where a temp file is written to disc and sometimes the EPP detects prior to payload execution and sometimes the payload wins the race.

The standard NSS testing methodology calls for the use of standard ports for browsing and exploit delivery. However, when NSS engineers tried the same tests over non-standard ports, Kaspersky failed to detect the exploits. This may be attributable to configuration options and underscores the need to block unused ports at the firewall as well as the need to test implementations of security products in the actual enterprise environment.

In testing exploits delivered over HTTPS, NSS engineers noted that the browser would often crash when testing the Kaspersky product. While this did prevent the payload from executing, it is not the ideal approach to exploit protection and can result in excessive helpdesk calls.

Test Methodology

Methodology Version: Endpoint Protection Test Methodology v3.0

This test report is one of a series of several tests in our "**Whole Product Test**" series. The scope of this particular report is limited to **Host Intrusion Prevention** vs. **Exploits**. No Zero-Day exploits against unknown vulnerabilities were included in this test.

Other tests in this series include:

1. **Socially engineered Malware –** Web-based malware that tricks users into downloading and installing it.
2. **Host Intrusion Prevention – This report**
3. **Evasion Defenses** – Preventing attempts to circumvent AV and HIPS
4. **Anti-Malware (classic)** – Email, Network Share, and USB infection vectors
5. **Live Web-Based "Drive-By" Exploits** – Live testing using Internet-borne exploits that insert malware payloads. Also known as "Drive-by" or "non-consensualll downloads"
6. **Performance** – Increase in Memory, CPU, Boot Time, and Application Load Time.

The Tested Products

The following is a current list of the products that were tested and are sorted alphabetically:

1. AVG Internet Security Business Edition 2012 2012.0.2221
2. ESET Endpoint Security 5 5.0.2126.0
3. F-Secure Client Security 9.31
4. Kaspersky Endpoint Security 2012 12.0.0.374 8.1.0.831 (a)
5. McAfee Endpoint Protection 8.8.0
6. MS System Center 2012 Endpoint Protection 2.2.903.0
7. Norman Endpoint Protection 9.00.000
8. Panda Cloud Antivirus Pro 2.0.0
9. Sophos Endpoint Security & Control 10.0
10. Symantec Endpoint Protection 12.1.1101.401 RU1 MP1
11. Trend Micro Office Scan 10.6.2401 Service Pack 1

Vendors were allowed to make configuration changes if it was determined that the default settings were not optimal. Product settings were verified by browsing to real websites on the Internet that utilize common applications used during the test. This ensured vendors applied realistic policies and did not skew the test by simply setting their product to "block all."

Products were connected to the live Internet, and had access to vendor cloud services. Updates were enabled with whatever frequency was set by the manufacturer.

Once testing began, the product version was frozen, in order to preserve the integrity of the test. Given the nature of endpoint protection platforms, virus signatures and definition updates as well as HIPS updates were enabled with whatever frequency was set by the manufacturer.

Client Host Description

All tested software was installed on identical machines, with the following specifications:

- Microsoft Windows XP SP3, and Windows 7 32-bit operating systems
- 2 GB RAM (XP SP3), 4 GB RAM (Windows 7)
- 20 GB HD (XP SP3), 40 GB HD (Windows 7)

The Vulnerabilities

Vulnerabilities were primarily selected based upon their severity and prevalence. They include vulnerabilities found in Microsoft Windows Internet Explorer, Mozilla Firefox, Adobe Acrobat, Apple QuickTime and other widely used applications.

All of the vulnerabilities selected by NSS had been public for several months (or years). The test set did not contain any zero-day vulnerabilities. Each of the selected vulnerabilities permitted arbitrary code execution. All exploits were validated on vulnerable systems.

The following list contains some examples of the vulnerabilities tested (this list is not exhaustive, and is provided only to give an indication of the *types* of vulnerabilities used in testing):

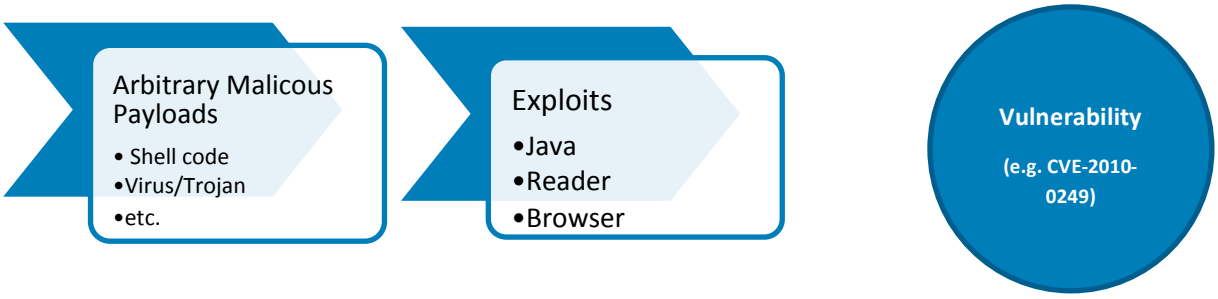
Vulnerabilities	Descriptions
CVE-2012-1875	Microsoft Internet Explorer 8 does not properly handle objects in memory, which allows remote attackers to execute arbitrary code by accessing a deleted object, aka "Same ID Property Remote Code Execution Vulnerability."
CVE-2011-1276	Buffer overflow in Microsoft Excel 2002 SP3, 2003 SP3, and 2007 SP2; Office 2004 and 2008 for Mac; Open XML File Format Converter for Mac; Excel Viewer SP2; and Office Compatibility Pack for Word, Excel, and PowerPoint 2007 File Formats SP2 allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a crafted Excel spreadsheet, related to improper validation of record information, aka "Excel Buffer Overrun Vulnerability."
CVE-2011-2371	Integer overflow in the Array.reduceRight method in Mozilla Firefox before 3.6.18 and 4.x through 4.0.1, Thunderbird before 3.1.11, and SeaMonkey through 2.0.14 allows remote attackers to execute arbitrary code via vectors involving a long JavaScript Array object.

CVE-2011-3544	Unspecified vulnerability in the Java Runtime Environment component in Oracle Java SE JDK and JRE 7 and 6 Update 27 and earlier allows remote untrusted Java Web Start applications and untrusted Java applets to affect confidentiality, integrity, and availability via unknown vectors related to Scripting.
CVE-2010-1297	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 or cause a denial of service (memory corruption) via crafted SWF content, related to authplay.dll and the ActionScript Virtual Machine 2 (AVM2) newfunction instruction, as exploited in the wild in June 2010.
CVE-2010-0886	Unspecified vulnerability in the Java Deployment Toolkit component in Oracle Java SE and Java for Business JDK and JRE 6 Update 10 through 19 allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors.
CVE-2010-0806	Use-after-free vulnerability in the Peer Objects component (aka iepeers.dll) in Microsoft Internet Explorer 6, 6 SP1, and 7 allows remote attackers to execute arbitrary code via vectors involving access to an invalid pointer after the deletion of an object, as exploited in the wild in March 2010, aka "Uninitialized Memory Corruption Vulnerability."
CVE-2009-0927	Stack-based buffer overflow in Adobe Reader and Adobe Acrobat 9 before 9.1, 8 before 8.1.3, and 7 before 7.1.1 allows remote attackers to execute arbitrary code
CVE-2009-0075	Microsoft Internet Explorer 7 does not properly handle errors during attempted access to deleted objects, which allows remote attackers to execute arbitrary code via a crafted HTML document, related to CFunctionPointer and the appending of document objects, aka "Uninitialized Memory Corruption Vulnerability."
CVE- 2008-5353	The Java Runtime Environment (JRE) for Sun JDK and JRE 6 Update 10 and earlier; JDK and JRE 5.0 Update 16 and earlier; and SDK and JRE 1.4.2_18 and earlier does not properly enforce context of ZoneInfo objects during deserialization, which allows remote attackers to run untrusted applets and applications in a privileged context, as demonstrated by "deserializing Calendar objects"
CVE-2008-4844	Use-after-free vulnerability in mshtml.dll in Microsoft Internet Explorer 5.01, 6, and 7 on Windows XP SP2 and SP3, Server 2003 SP1 and SP2, Vista Gold and SP1, and Server 2008 allows remote attackers to execute arbitrary code via a crafted XML document containing nested SPAN elements, as exploited in the wild in December 2008.

Further information about vulnerabilities can be found at http://cve.mitre.org, a public, government-funded web site established as a clearinghouse for vulnerability information.

Appendix A: Definitions

The following definitions and analogies are provided in an effort to provide clarification, as well as to bridge an ongoing communication gap between security vendors and their customers.



Vulnerability

A perfect lock can only be opened by a key with a specific pattern. If a lock can be opened with a different key then it has a vulnerability. If nobody can actually build the alternate key that will open the lock then the vulnerability cannot be exploited. An example of a software vulnerability is an improperly defined memory usage within a function that enables unauthorized content to be sent to a specific memory location and then executed with privileged rights.

Exploit

An exploit is a specially crafted code sequence which can ”trigger” or ”unlock” a vulnerability within an application, such as a heap spray, buffer overflow attack, etc. In the context of the above vulnerability example, an exploit is using an incorrect key to unlock the vulnerable lock. When such a key is built exclusively to prove that lock is vulnerable it is called a “proof of concept”. When such a key is used to criminally exploit systems it is said to be “in the wild.” Practically speaking, virtually any exploit for which there is a viable “proof of concept” is being exploited in the wild and poses a threat to consumers, corporations and governments. An exploit can be planted in a compromised website where it silently infects visiting computers, can be embedded in an attachment delivered though email, or can be launched from an another computer (remote attack) automatically via software or manually by a hacker.

Payload

The payload is the content that is delivered once the vulnerable application has been exploited. Payloads can range from inactive political or religious statements to the complete remote control of the affected computer. For automated attacks the payload maybe something as relatively innocuous as adware or as costly as a rootkit combined with a banking or gaming password-stealing trojan. For a manual attack the payload may provide a remote hacker with complete control of the compromised system and access to all information on the system. In a home environment the payload may result in identity theft, or compromise of email or social networking accounts. In a business environment, including those allowing BYOD network access, compromise of a workstation may allow an attacker to tunnel deeper into a network.

Contact Information

NSS Labs, Inc.
206 Wild Basin Road, Suite 200A
Austin, TX 78746
+1 (512) 961-5300
info@nsslabs.com
www.nsslabs.com