Real World Enterprise Security Exploit Prevention Test

February 2014

1 Contents

2 Introduction

Web browsing is an integral part of both home and corporate internet users' daily activity. The web is almost ubiquitous and people use it for communication, social life, gaming, business, shopping, education, etc. People browse the web very often with outdated software (both at home and in the enterprise) and these outdated applications have known vulnerabilities. Some of these vulnerabilities let the attackers run code on the victim's computer, without any warning on the victim side. After the victim's computer is infected, the attackers can use this malicious code to steal money from their internet banking application, steal credit card data, steal personal information, steal confidential corporate information, or even lock the computer until the victim pays a ransom.

Drive-by download exploits are one of the biggest threats and concerns in an enterprise environment because no user interaction is needed in order to start the malware on the victim machine. Even traditional, legitimate sites used by enterprises on a daily basis get infected by malware. Java based exploits are especially popular among organized criminals because traditional memory corruption based protection methods are not effective against Java exploits. Outdated Java runtime environments are very "popular" in enterprise environments because enterprise Java applications are not compatible with the newest JRE versions, thus enterprises can't upgrade to the new versions. Exploits and drive-by download attacks are commonly used in Advanced Persistent Threat (APT) attacks.

Home users and small to medium businesses often lack the knowledge and awareness about exploits, exploit prevention, targeted attacks and the importance of software updates. Big enterprises face the challenge of managing complex IT systems and consequently, they run a high probability of becoming a target of exploit and malware based attacks.

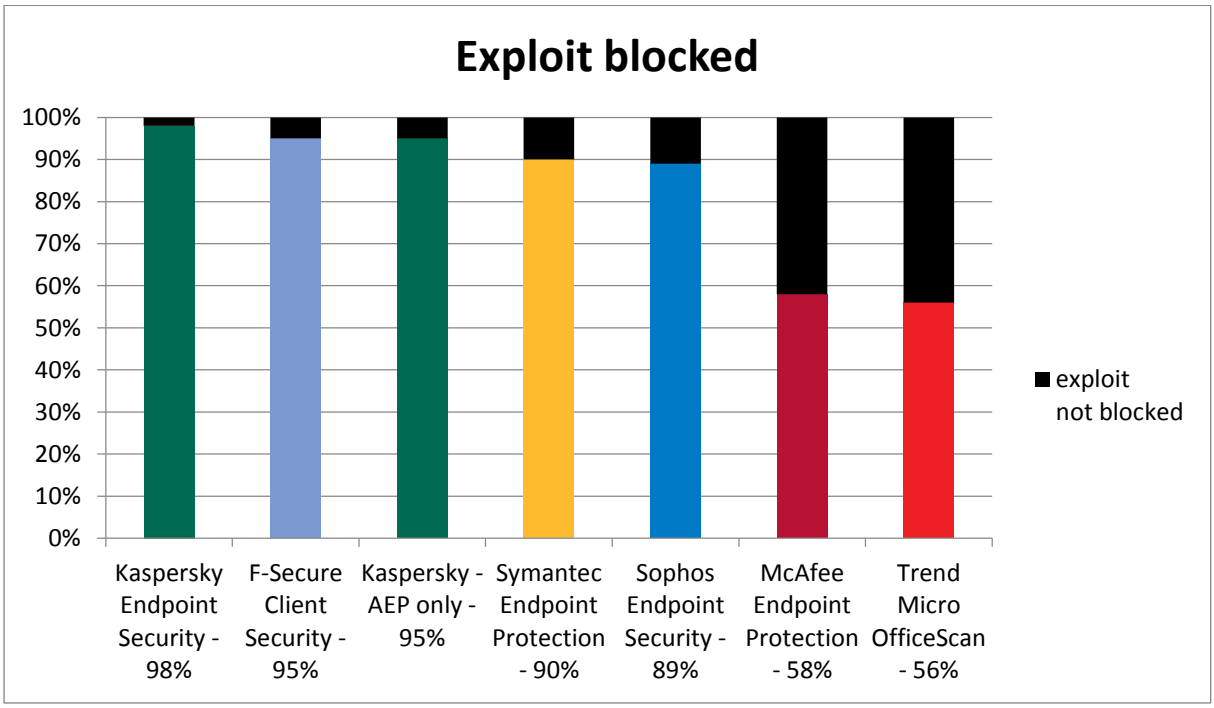
Endpoint protection systems have had a long journey from traditional signature based protection to that which is implemented in a modern protection system. Advanced heuristics, sandboxing, intrusion prevention systems, URL filtering, cloud based reputation systems, Javascript analysers, memory corruption protection, etc. are now used to combat modern malware threats. In order to test an endpoint protection system, one has to test all modules of the protection employed by that system, and the test has to be done in a way which emulates standard user behaviour accurately. Today the vast majority of threats are delivered via the web, this is the reason why our test focuses exclusively on web based exploits. When an endpoint protection system cannot protect its users against malicious software (malware), the damage might be catastrophic. To cite a few examples of threats which can cause catastrophic damage, there is malware which steals confidential information, or malware which can wipe important documents or whole workstations. Attacks like these can cause huge damage to corporate intellectual property or can block business processes for weeks. Our test incorporated a wide range of different malware types, thus emulating a real world scenario as closely as possible.

This assessment was commissioned and sponsored by Kaspersky Lab to serve as an independent efficacy assessment of its Kaspersky Endpoint Security (KES) for Windows product and it's Automatic Exploit Prevention (AEP) module. KES is an endpoint protection solution, integrating anti-malware solutions like traditional signature matching, proactive defense technologies, cloud reputation services, personal firewall and IPS, etc. The purpose of the AEP module of KES is to monitor the system for known or unknown exploit behaviour and when detected, block the exploit code payload execution.

The objective of this report is to provide an assessment of the ability of KES with full functionality and the AEP technology inside KIS to prevent drive-by exploitation when KES is installed on an endpoint. In order to put performance in perspective, it was tested alongside, six competitor products. Each of the products was installed on an endpoint and tested against 110 unique exploit sites.

The brief final result of the exploit protection test is shown in the table below.

(For detailed results along each test case please refer to chapter 3)



Graph 1 - Distribution of exploit detection

From the results we can conclude that three endpoint protection systems stand out of the crowd with their excellent protection:

- Kaspersky Endpoint Security
- F-secure Client Security
- Kaspersky Endpoint Security with AEP module only

2.1 Test methodology

The test was conducted as follows:

1. One default install Windows 7 Enterprise 64 Service Pack 1 virtual machine (Virtualbox) endpoint was created. (Windows 7 64-bit was the most popular OS for the target audience.) One host-only and one NAT interface was configured. The default HTTP/HTTPS proxy was configured to point to proxy running on the host OS using the host-only interface. SSL/TLS traffic was not intercepted on the proxy, because it would unnecessarily increase the complexity of testing and drive-by-exploit sites with valid SSL certificates are very rare.
2. The security of the OS was weakened by the following actions:
    a. User Account Control was disabled
    b. Microsoft Defender was disabled
    c. Internet Explorer Smartscreen was disabled
3. The following vulnerable software was installed, based on 2013 Q3 statistics:
    a. Java 1.7.0
    b. Adobe Reader 9.3.0
    c. Flash Player 10.1.102.64
    d. Silverlight 5.1.10411
    e. Internet Explorer 8.0.7601.17514

    These version numbers have been specified with the following two requirements:
        1. The highest number of in-the-wild exploits should be able to exploit this specific version, thus increasing the coverage of the tests.
        2. The version must currently be popular among users.
4. Windows Update was enabled, but only patches which did not affect Internet Explorer were installed. No patches were installed after November 19, 2013.
5. From this point, 8 different snapshots were created from this virtual machine, each with the different endpoint protection products and one with none. This procedure ensures that the base system is exactly the same between the test systems. The following endpoint security suites in the following configuration were defined for this test:
    a. No additional protection, this snapshot has been used to infect the OS and to verify the exploit replay (see 2.4 for details).
    b. Kaspersky Endpoint Security 10.1.0.867 and later with default configuration
    c. Kaspersky Endpoint Security 10.1.0.867 with AEP module only. This means the following modules has been turned off: File anti-virus, Web Anti-virus, Application Privilege Control, Application Startup Control and Web control.
    The System Watcher has been configured as default, as seen on the following screenshot:

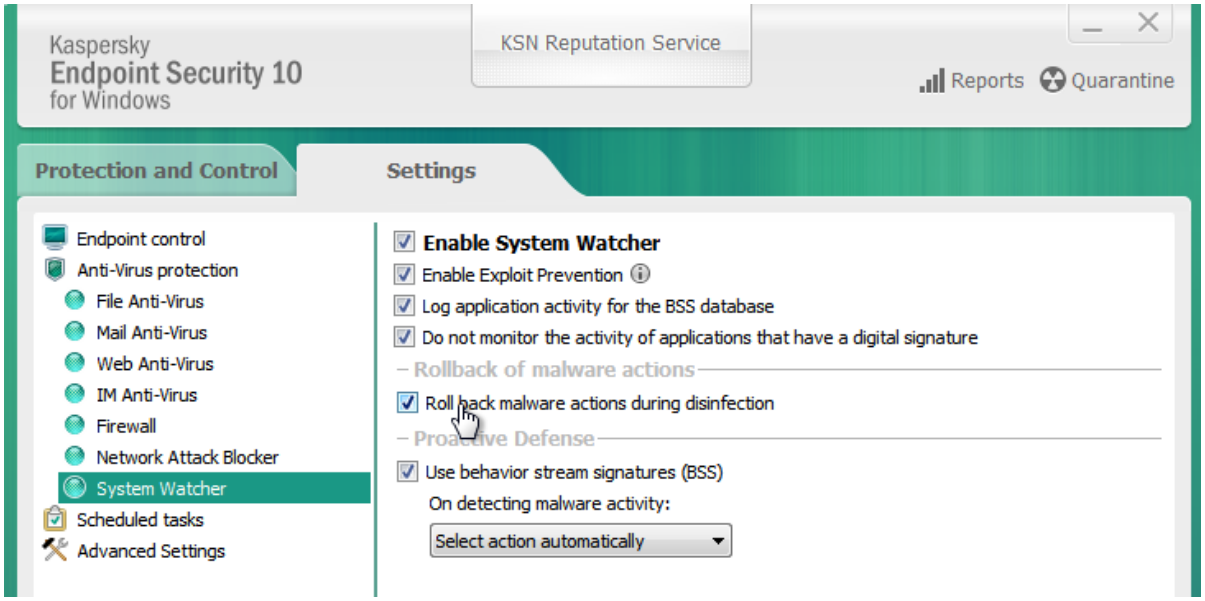


Figure 1 – KES configuration

d. McAfee Endpoint Protection including VirusScan Enterprise + AntiSpyware Enterprise 8.8.0.975, Siteadvisor 3.5.0.724, Host Intrusion Prevention 8.0 and DLP Endpoint 9.2, because these components are in the default suite.
e. F-Secure Client Security 11.00 build 332
f. Symantec Endpoint Protection 12.1.3001.165
g. Sophos Endpoint Security and Control 10.3.1
h. Trend Micro OfficeScan 10.6.3205 SP2

The endpoint systems were installed with default configuration, potentially unwanted software removal has been enabled, and if it was an option during install, cloud/community participation was enabled. The management servers were installed onto a different server. The purpose of management servers are to centrally administer, update and analyse logs in an enterprise environment. Installing the management server on a different server is highly recommended by the vendors, so it does not interfere with the testing, the machine resources are not used by the management server, etc.

6. Using the non-protected operating system, malicious URL's were crawled, and the tester waited for new processes (malware) to start. Mouse movement was emulated in order to simulate a real user navigating the site. Besides the traditional drive-by download exploits (where no user interaction is needed to start the malware, only visiting the infected URL), we simulated users who opened Office files (e.g. .xls, .doc, .pdf) and users allowing the signed applets (both with valid and invalid/self-signed certificates). The whole exploit traffic was recorded by the proxy server. In spite of other "real world protection tests", no binary downloads (e.g. exe) files were directly started. ActiveX, VBscript based downloaders and Office macro documents were out of scope.
7. After successful exploitation, the virtual machine was reverted to a clean state and traffic was replayed by the proxy server. The replay means that the browser was used as previously, but instead of the original webservers, the proxy server answered the requests based on the recorded traffic. In this replay, no other traffic was allowed, which means unmatched requests (previously not recorded) were answered with HTTP 404 codes. When the "replayed exploit" was able to infect the OS, the exploit traffic was marked as a source for the tests. This method guarantees that exactly same traffic will be seen by the endpoint protection systems, even if the original exploit kit goes down during the tests. Although this might be axiomatic, it is important to note that no exploit traffic test case has been deleted after this step of the test, every test is included in the final results. In the case of HTTPS traffic, the original site has been contacted, without replaying.
8. After new exploit traffic was recorded, the endpoint protection systems were tested, in a random order. Before the exploit site has been tested, it was verified that the endpoint protection had been

updated to the latest version with the latest signatures and that every cloud connection was working. If there was a need to restart the system, it was restarted. In the proxy setup, unmatched requests were allowed to pass through and SSL/TLS was not decrypted in order to ensure the cloud connectivity. No VPN has been used at this stage of the test. When user interaction was needed (e.g. site is not recommended to visit, etc.), the block/deny action was chosen. No other processes were running on the system, except the Process Monitor from Sysinternals.

9. After navigating to the exploit site, the system was monitored to check for new processes. For the analysis of the results, please see section 2.4.
10. After an endpoint protection suite was tested, a new endpoint protection was randomly selected for the test until all endpoint protection products had been tested.
11. The process goes back to step 7. until all 110 exploit site test cases have been reached.

The following hardware has been dedicated to the virtual machine:

- 2 GB RAM memory
- 1 core of the Core-i5 1.7GHZ CPU
- 20 GB free space
- 1 NAT and 1 host-only interface

2.2 Source of exploits

Only exploit traffic which drops and immediately starts malware was included in the test. This was verified via Process Monitor, looking for Operation = Process Create, either direct malware execution, or via regsrv32, cmd.exe, wscript.exe, java.exe, etc. Modules loading ("Load image" in Process Monitor) has not been monitored.

100% of the exploit traffic was sourced from MRG Effitas own malicious URL feed (a special part of it which is not shared with others). 80% of the exploit traffic was replayed within 24 hours of collection. The different endpoint protection systems were tested with a delay of not more than 6 hours and the endpoint protection systems were tested in a random order.

The "110 different exploit traffics used" means that both the domain of the infected site and the domain of the exploit kit site were distinct from the other test cases.

In a small percent of the cases (15%), the replay from the original infected URL was not successful (for example the redirection URL's were dynamically changing) or the original infected site was not available. In these cases the replay was started from the landing page of the exploit site. In these cases, the infected URL and redirection chain could not be seen by the endpoint protection system, so it had no chance to block the exploit at these early stages, but still it had a lot of other opportunities to block the malware.

While collecting the exploit traffic, the client has been emulated via VPN so as to appear to reside in different countries of the world (UK, US, Germany, Russia, etc.) in order to simulate the global threat more accurately.

For the diversity of the exploits used, please see section 2.5.

Details of the samples, including their URLs and code, were provided to partner vendors only after the test was complete.

2.3 False positive test

No false positive test has been carried out because there is no relevant false positive test which can truly measure the false positive ratio of such a complex scenario. Doing a false positive test on the URL block component cannot be considered as a valid false positive test because it only measures one component of the endpoint protection system. For example in the case where only the AEP module of Kaspersky has been

turned on, a false positive test cannot be carried out via visiting clean URLs, or by running legitimate applications.

2.4 Analysis of the exploit results

The testing was carried out between November 19, 2013 and February 7, 2014.

We have defined the following stages where the exploit can be prevented by the endpoint protection system:

1. Either by blocking the URL (infected URL, exploit kit URL, redirection URL) by the URL database (local or cloud), or by analysing and blocking the page containing malicious Javascripts (redirects, iframes, obfuscated Javascripts, etc). For example a typical result is when the browser displays a "site has been blocked" message by the endpoint protection. In the case of Java exploits, no java.exe starts on the victim, because the applet HTML code was not able to reach the browser renderer process.
2. Blocking the exploit before the exploit payload (e.g. "download malware and execute") can be executed. E.g. in the case of Java exploits, the java.exe process starts, but in the proxy traffic it can be verified that the malware payload request has not been initiated.
3. Blocking the downloaded payload by analysing the malware before it is started (Process Create). E.g. the malware payload download (either the clear-text binary or the encrypted/encoded binary) can be seen in the proxy traffic, but no malware process starts.
4. There was a successful Process Create by the dropped malware.

The first and second exploit prevention stage has been counted together to simplify the results. At this stage no malicious commands have run on the victim computer. This is the expected behaviour of an endpoint protection system; the attacker has no chance to execute any untrusted code on the victim.

The third stage is the final chance of the endpoint protection system to block the malware. It is important to mention that in this stage of block, malicious code (the exploit code's payload) was able to run on the victim machine. Although this is usually some kind of "download and execute" code to drop malware on the victim machine, this payload can be changed by the attackers easily (especially in the case of Java exploits), which might go undetected by the endpoint protection system - especially in the case of a targeted attack.

When the endpoint protection system did not block the exploit, let the payload to download malware and let it run, it was a complete fail of the product. In most of the cases, the endpoint protection systems were able to detect some or all parts of the malware, but this has not been recorded/counted because of the following reasons:

- The scope of the test was exploit prevention and not the detection of malware running on the system.
- It cannot be determined what kind of commands has been executed by the malware or what information has been exfiltrated by the malware. Data exfiltration cannot be undone or remediated.
- It cannot be determined if the malware exited because endpoint protection system blocked it, or malware quit because it detected monitor processes (procmon.exe), virtualization, or the malware quit because it did not find its target environment.
- Checking for malware remediation can be too time consuming and scoring of the remediation can be highly difficult. For example we experienced several alerts that the endpoint protection system blocked a URL/page/exploit/malware but still, the malware was able to execute and run on the system. On other occasions, the malware code was deleted from the disk by the endpoint protection system, but the malware process was still running, or some parts of the malware was detected and killed but others not.
- Sometimes the products blocked some or all parts of the malware from running, but failed to notify/alert the user or administrator about the incident.

We believe that this zero-tolerance scoring helps enterprises to choose the best products by using simple metrics. Manually verifying the successful remediation of the malware in an enterprise environment is a very resource intensive process and costs a lot of money. In our view, malware has to be blocked before it has a chance to run. If an enterprise antivirus administrator receives alert that malware has been blocked on the workstation, it is highly recommended to investigate this alert more deeply, which is time and resource consuming. When an enterprise antivirus administrator receives alert that page or exploit has been blocked on the workstation, the tasks to be performed are significantly lower.

Illustrative of the above is one of our test cases, where a totally new malware family (a Java based multi-platform malware) was dropped via a Java exploit. Although none of the endpoint protection systems were able to detect the malware itself, some endpoint protection systems were able to block the exploit before the malware had a chance to start.

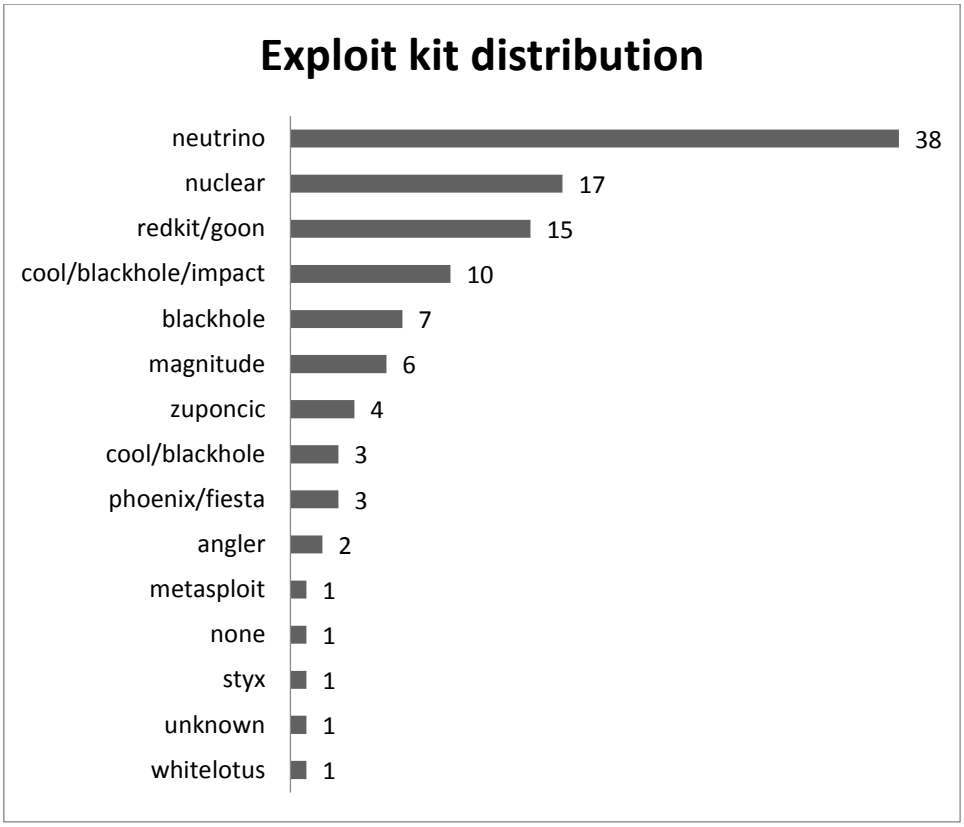
In 15% of the cases, the payload download process was not proxy aware (e.g. javaw.exe, wscript.exe) and it was using direct connection through the NAT interface, thus the binary payload was not recorded and could not be replayed. This means the exploit would fail in a corporate environment, but it would be successful if the mobile device (notebook) leaves the company. In 11% percent of the test cases, the malware was not served by the exploit kit at the time of the replay test. The scoring has been changed in these cases to the following:

When javaw.exe or wscript.exe started and there was a payload request on the network, this means the exploit was successful. Because the malware could not be downloaded (the exploit kit served HTTP 404 instead of the binary), we calculated as if the payload has been blocked by the endpoint protection system (even if the malware might have evaded the endpoint protection system).

We know that by detecting malicious activity only by a "new malicious process has been created" might miss some highly sophisticated exploit and malware where no process creation is done, but these samples have statistically very low probability. These examples include when a library is loaded, or a shellcode is directly executed without any files dropped on a disk.

2.5 Analysis of the exploit kits used in the test

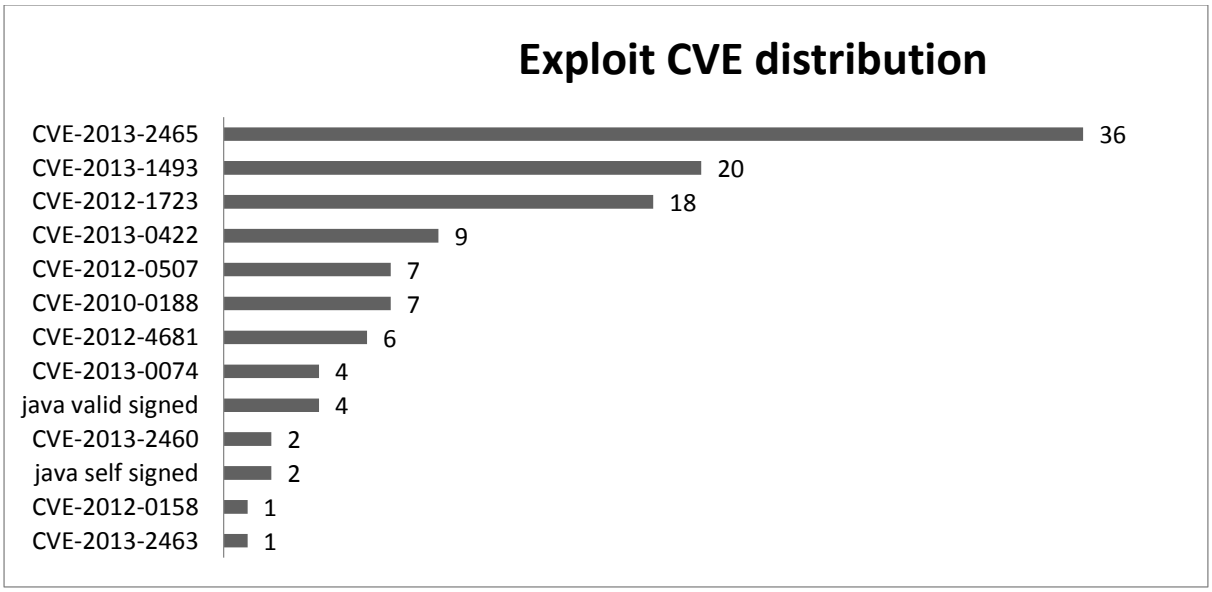
The following graph displays the distribution of the detected exploit kits. The following distribution is the result of choosing random malicious sites at the time of testing. The exploit kit names have been determined via Emerging Threats Pro IDS alerts and manual analysis.



Graph 2 - Distribution of the exploit kits

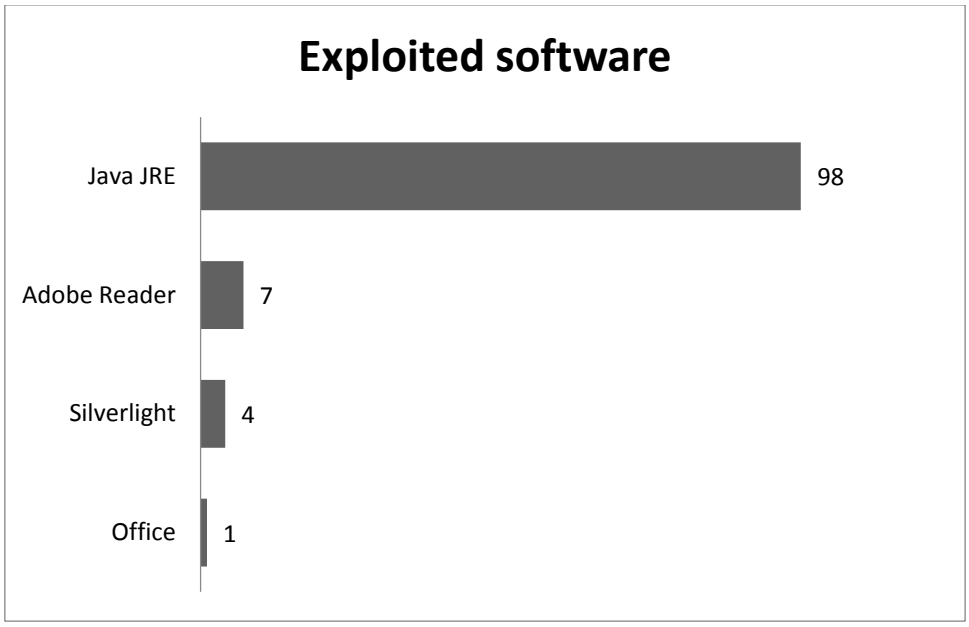
2.6 Analysis of the exploits used in the test

The following graph displays the distribution of the detected exploits. Some exploit kits delivered more than one exploit. When multiple exploits were delivered, but it was easy to determine which one was successful (e.g. there was a PDF exploit, but Java started the malware), only the successful has been included in the following bar chart. We used multiple AV engines to classify the exploits.



Graph 3 - Distribution of the exploit CVEs

The following graph displays the distribution of the exploited software. When multiple exploits were delivered, but it was easy to determine which one was successful, only the successful has been included.



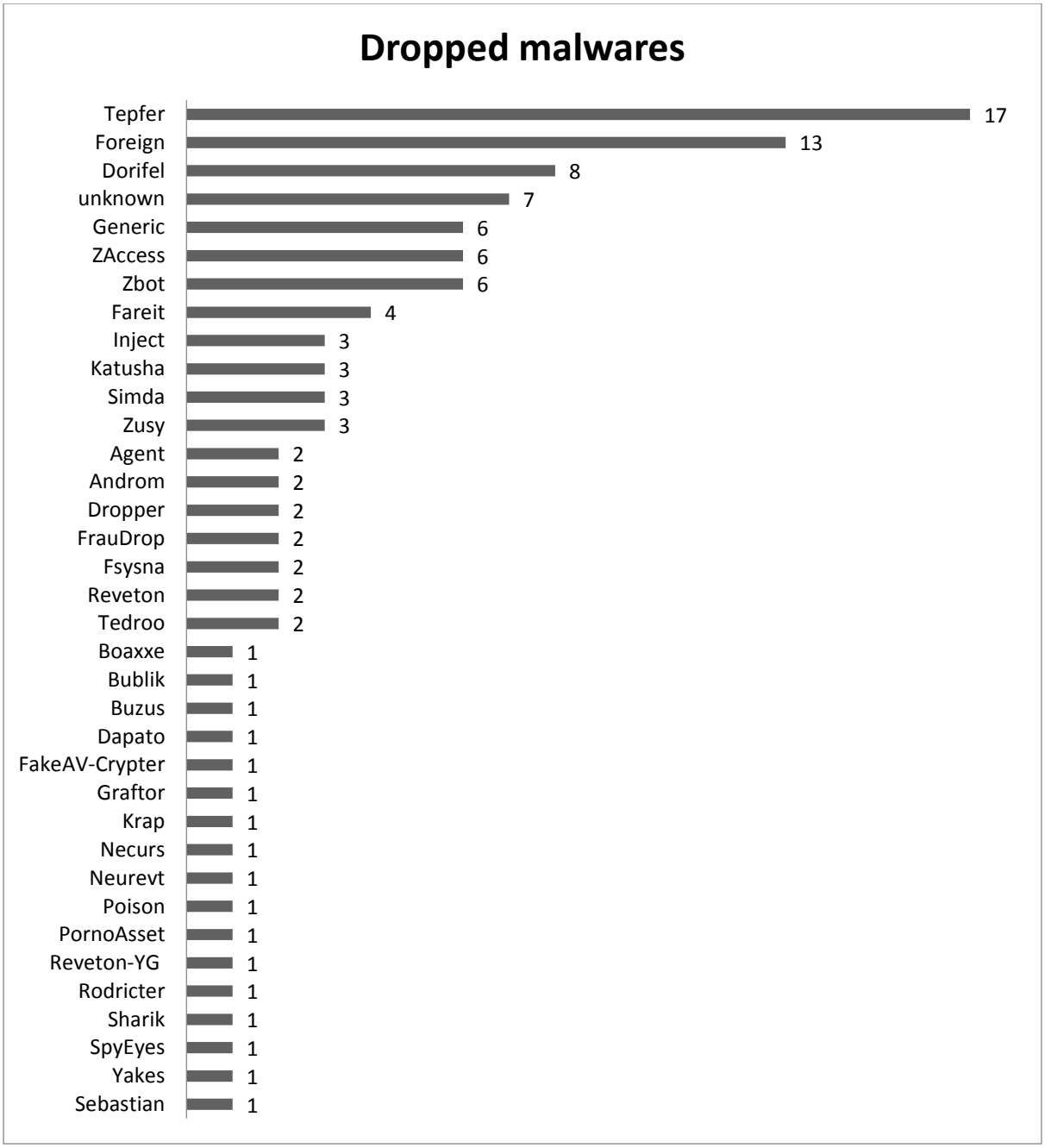
Graph 4 - Distribution of the exploited software's

Although it might be surprising that the number of Java Runtime Environment exploits is so high and the others are so low, we concluded that the following reasons might be behind this result:

- exploit can't bypass both DEP and ASLR on Windows7
- exploit/payload is not 64-bit compatible
- exploit needs Java Runtime 6 to bypass ASLR
- exploit needs specific dll version (e.g. ntdll), which is different on the test computer
- multiple exploit is running in parallel and one exploit crashes the browser before the other exploit is successful
- exploit kits favour Java exploits.

2.7 Analysis of the dropped malware

The following graph displays the distribution of the detected malware family name. When it was not a generic but a specific name, the Kaspersky malware family name has been used. When it was a generic name or not detected by Kaspersky, we looked at other AV detected family names. When other AV vendors had Generic as well, generic has been used. We were not able to recover some malware, this has been tagged as unknown.



Graph 5 - Distribution of the dropped malware

We would like to highlight some of the top malware in the test:

Tepfer is a multi-component malware family, it can steal passwords and confidential information from the infected machine, and send it back to the attackers.

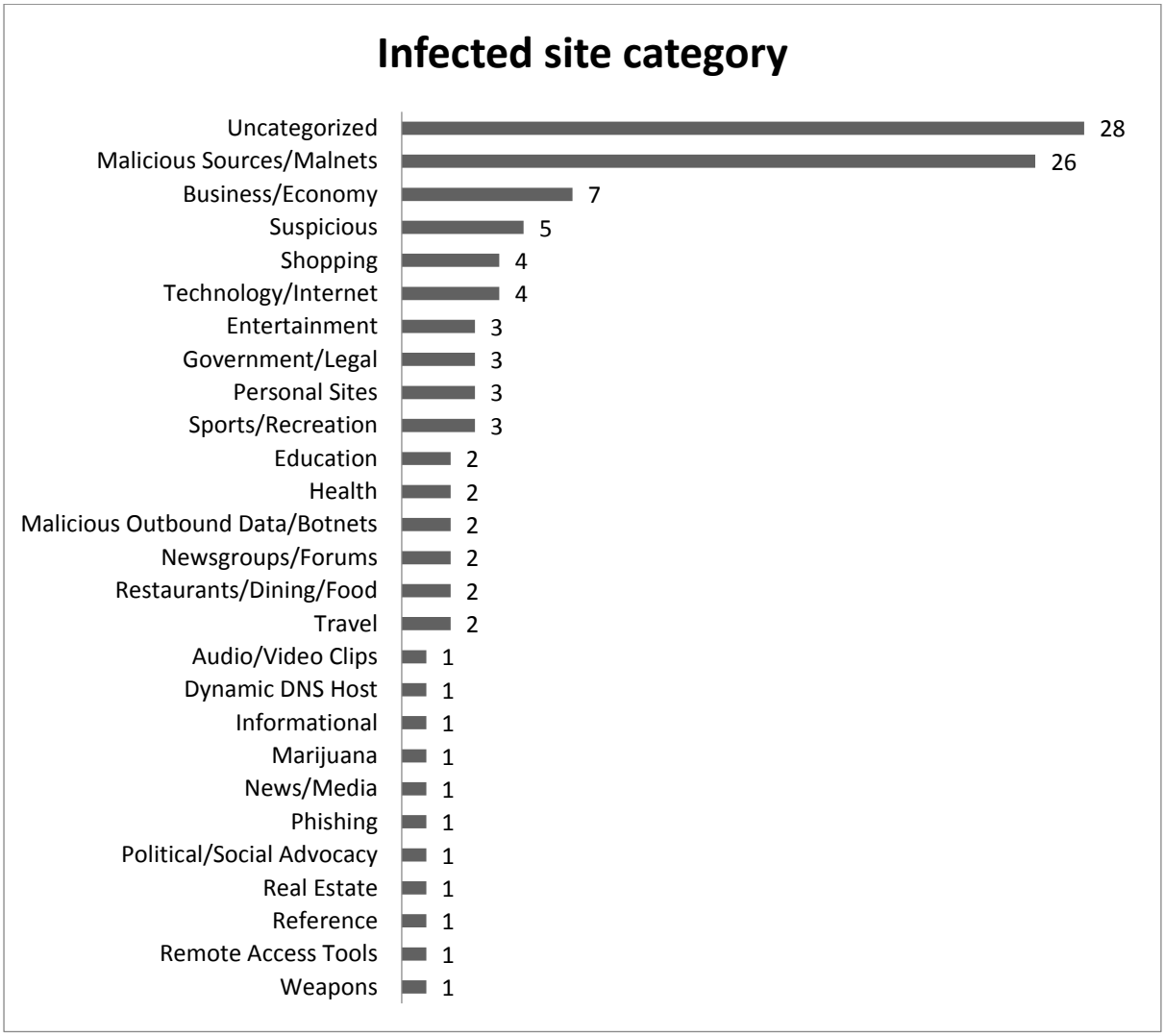
ZAccess is a rootkit malware, and it is used to push fake applications, adware and other malware components to the infected machine.

ZBot refers to the financial stealer malware family based on Zeus. Zeus can steal passwords from the browser, inject web forms for additional data stealing, steal private certificates, etc.

All malware families in the test are in the wild malware, and most of them can be considered as high risk when infecting a workstation.

2.8 Analysis of the infected sites used as entry URLs

The following graph displays the distribution of the categories used as infected URLs (not including the redirect pages and the site serving the exploit). We used the "Blue Coat WebPulse Site Review" service to categorize the URL's.



Graph 6 - Distribution of the infected site categories

As it can be seen on the graphs, a lot of legitimate sites have been infected and served as an entry point to the exploits, which means casual internet browsing can put the users at risk. In these cases there is no need for a user to click on spam links.

3 Final results

We divided the results into two sections.

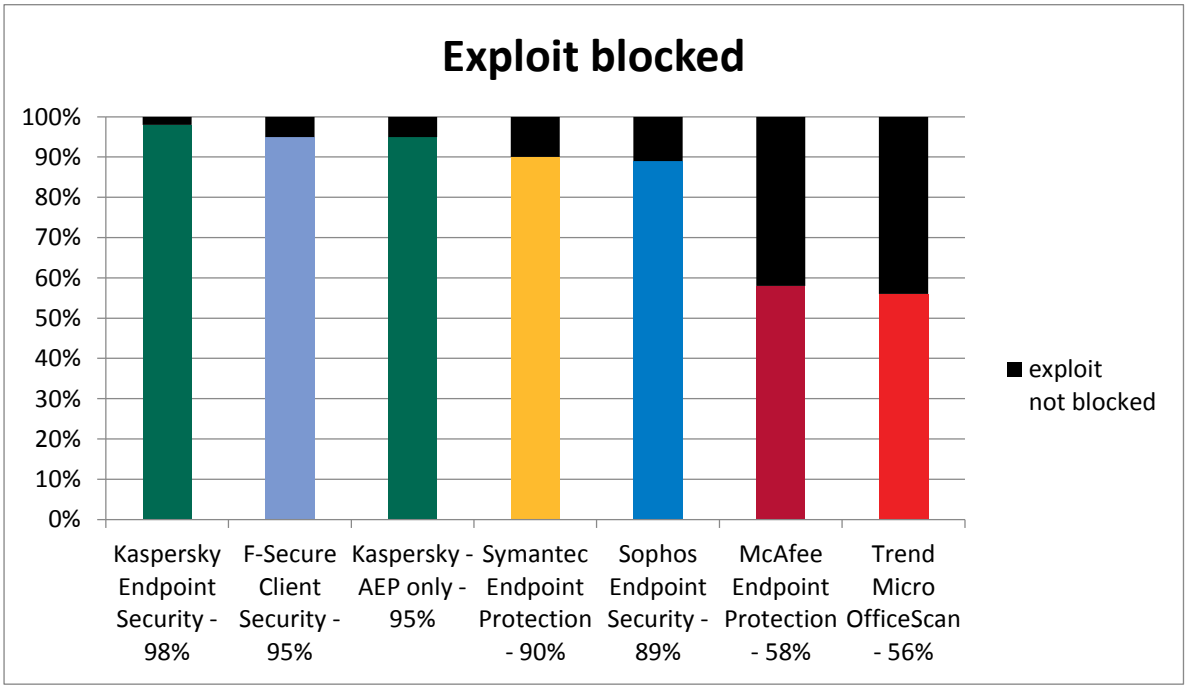
The first section is the true exploit block capability of the product, where the threat has been blocked either at the malicious URL, HTML/Javascript or at the exploit stage.

The second section is where the threat has been blocked either at the URL, HTML/Javascript the exploit or payload delivery/start stage.

The difference between the two categories is that if the results in the second section are higher in a case of a product, this means that the product lacks of the detection of that particular exploit, and only the binary payload has been detected. The problem with this protection is that the malware has been blocked on the final stage of the defense, which can be easily circumvented by the malware by using a new variant of the malware or by dropping a new or targeted malware. Developing new undetectable malware is a lot easier from an attacker point of view than finding and exploiting a new vulnerability. Another problem is that the exploit's payload was already able to run on the machine, which can have different functions other than "download and execute"(e.g. creating backdoor users, non-persistent backdoor shells, in memory malware, etc.). In the case of targeted attacks, attackers might notice that exploit was successful, but payload has been detected, so they can change the binary so it is not detected by the endpoint protection system.

3.1 Malicious page blocked or exploit blocked

In this result set, only 104 of the exploit traffics were included. The Java signed applets (trusted and untrusted ones) have been excluded from this calculation, because these threats lack the traditional "exploit" stage, so it would not be fair to say that an endpoint protection system did not block the exploit if there was no exploit.



Graph 7 - Distribution of the exploit blocking capability

From the results we can conclude that three endpoint protection systems stand out of the crowd with their excellent protection:

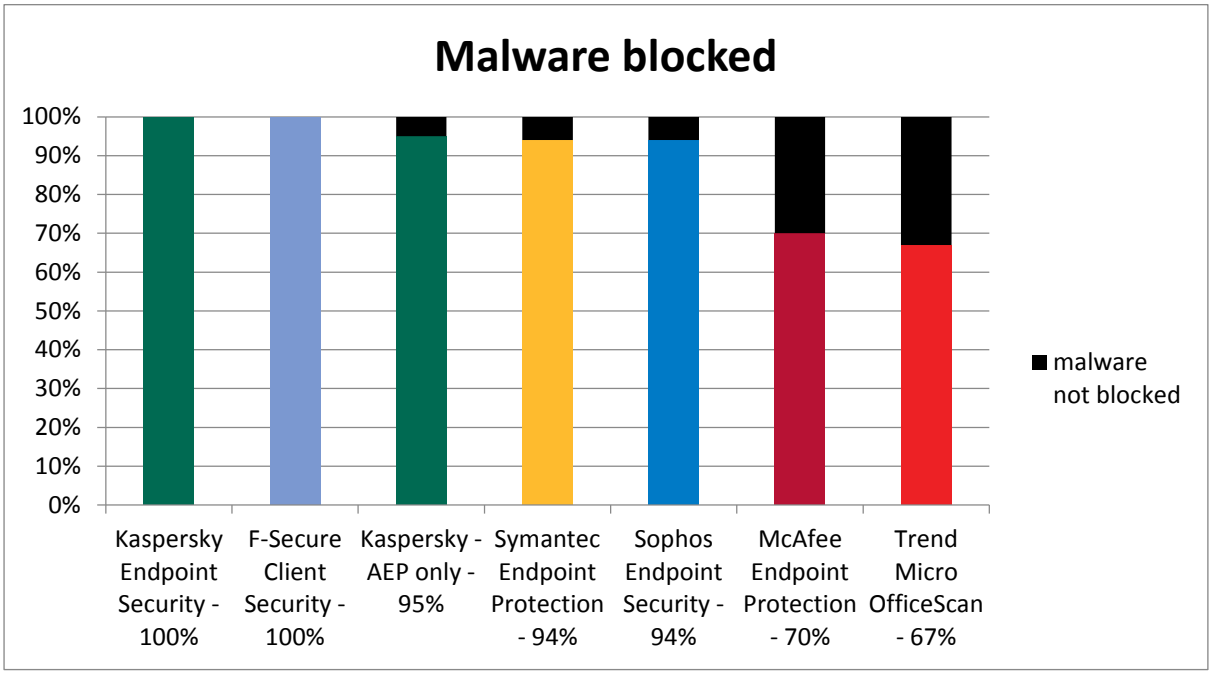
- Kaspersky Endpoint Security
- F-secure Client Security
- Kaspersky Endpoint Security with AEP module only

It is important to note that Kaspersky Endpoint Security with only the AEP module turned on still had far better protection than 4 of the other security suites having all components turned on (URL filter, web filter, web cloud reputation, file based antivirus, etc.)

3.2 Page blocked or exploit blocked or payload execution blocked

In this result set, all of the 110 exploit traffics has been included, except the case of the Kaspersky AEP only module, where again the same 104 exploit traffic has been counted. Because the AEP module has been designed to prevent exploits only and signed applets don't include exploits, it is not fair to test the AEP

module against signed applets.



Graph 8 - Distribution of the exploit and malware blocking capability

The results are better than in the previous graph, because more vendors were able to detect and block the payload after successful exploit. Please refer to Chapter 2.4 for additional information on this analysis.

Similarly to the previous section, from the results we can conclude that three endpoint protection systems stand out of the crowd with their excellent protection:

- Kaspersky Endpoint Security
- F-secure Client Security
- Kaspersky Endpoint Security with AEP module only

Again, it is important to note that Kaspersky Endpoint Security with only the AEP module turned on still had far better protection than the other four security suites having all components turned on.

If you want to contact us regarding this test, you can use the form on our web page:

http://www.mrg-effitas.com/contact/

4 Certifications

The following certification is given to Kaspersky Endpoint Security and F-secure Client Security: