Bitdefender

Anti-Virus Comparative

AV comparatives

Advanced Endpoint Protection Test

Language: English
November 2017

Last Revision: 23rd March 2018

www.av-comparatives.org

Commissioned by Bitdefender

Table of Contents

Executive Summary

Bitdefender commissioned AV-Comparatives to perform an advanced Endpoint Protection Test. Bitdefender GravityZone Elite Security was tested against competitor endpoint products selected by Bitdefender. The tests were performed in November 2017 - January 2018. The primary goal was to compare the automatic prevention and detection capabilities of different endpoint protection solutions. The following tests were performed:

- Proactive Protection Test, including False Alarm Test
- PowerShell-based File-less Attacks and File-based Exploits Test, including False Alarm Test
- Real-World Protection Test, including False Alarm Test
- Ransomware Test

Tested products

All products were purchased from distributors, except for the Bitdefender product, which was provided by the vendor itself. The following up-to-date licensed product versions were tested:

- Bitdefender Endpoint Security Elite 6.2
- Carbon Black Cb DEFENSE 3.0
- CrowdStrike FalconHost 3.7
- Cylance CylancePROTECT 2.0
- Kaspersky Endpoint Security for Business 10.3
- McAfee Endpoint Security 10.5.2
- SentinelOne Endpoint Protection 1.8.4[1]
- Sophos Central Endpoint Advanced Protection and Intercept X 11.5.9-3.6.10
- Symantec Endpoint Protection Standard 14.0

Settings

The following settings were applied to the products. Please note that for products which have high protection settings by default, or do not allow the relevant settings to be changed, we left the configuration at its default values.

- Bitdefender: all features activated.
- CrowdStrike: all settings were enabled and set to maximum.
- SentinelOne: all settings were enabled and set to maximum.
- Sophos: Deep Learning malware-detection module and False Positive Suppression were activated (these features have been available - in beta state - since early 2017).
- Carbon Black, Cylance, Kaspersky Lab, McAfee and Symantec products were tested with default settings.

[1] SentinelOne provided version 2.0 to new customers already in September 2017, but did not force a console upgrade to existing customers. For existing customers, version 1.8.4 (as used for this test) was the latest available version in the console at time of testing (November). In version 1.8.4, the default policy used to require cloud connectivity for mitigation. In version 2.x, the cloud validation policy concept was improved.

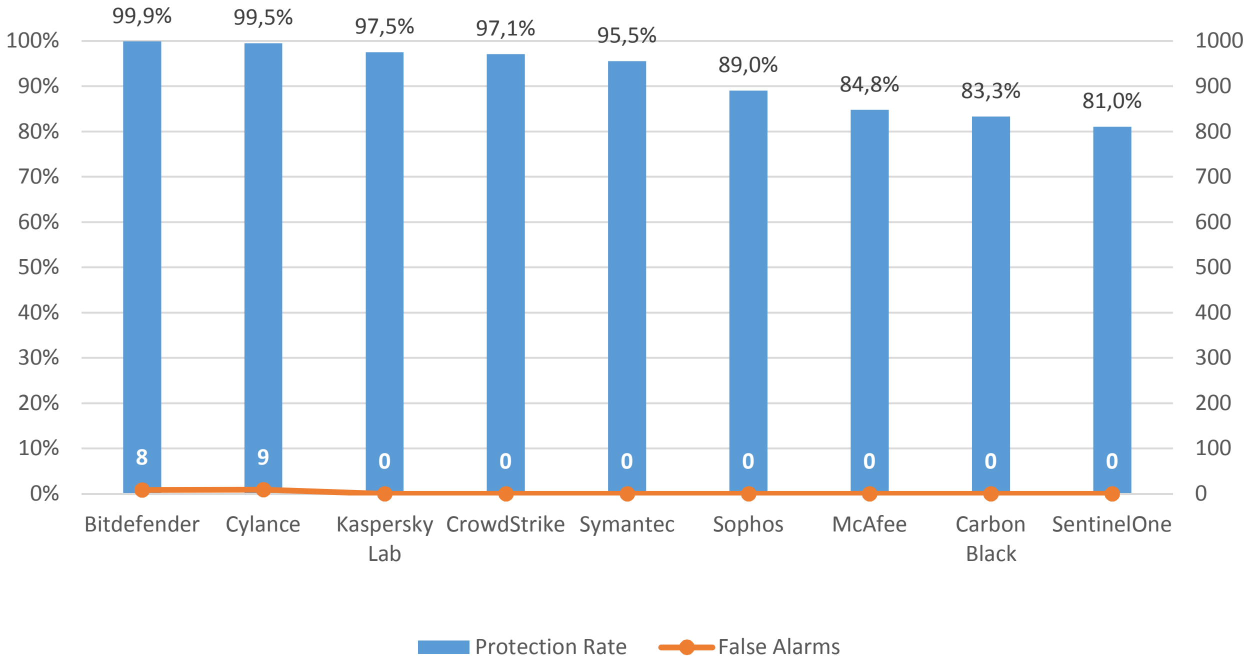
Proactive Protection Test

Bitdefender[2] asked us to run a proactive protection test of the type we used to do (https://www.av-comparatives.org/retrospective-test/), to measure how the products protect against as-yet unknown malware in environments which are disconnected from the Internet.

For this test we froze the products' malware definitions on the 14th of November 2017, and tested the products against 1,000 new malware samples which appeared in the week after the freeze date. All samples were verified by telemetry data as being unknown before the freeze date.

In order to verify that the products do not simply block everything by using paranoid detections, a false alarm test with 1,000 clean files was performed.

	Protection Rate	False Alarms
1. Bitdefender	99.9%	8
2. Cylance	99.5%	9
3. Kaspersky Lab	97.5%	0
4. CrowdStrike	97.1%	0
5. Symantec	95.5%	0
6. Sophos	89.0%	0
7. McAfee	84.8%	0
8. Carbon Black	83.3%	0
9. SentinelOne	81.0%	0



[2] A bug in the sandbox driver was discovered in the first product build of Bitdefender – the bug has been fixed and results corrected.

PowerShell-based File-less Attacks and File-based Exploits Test

File-based malware and ransomware such as VBS, JS or MS Office macros can install a backdoor on victims' systems and create a control channel (C2) to the attacker, who is usually in a different physical location, even in a different country. Apart from these well-known scenarios, it is possible to deliver file-less malware using exploits, remote calls (PSexec, wmic), task scheduler, registry entries, Arduino hardware (USB RubberDucky) and WMI calls. This can be done with built-in Windows tools like PowerShell. These commands load the actual malware directly from the Internet into the victim's memory and continue to expand further into the local network with native OS tools, or may even become persistent on machines in this way.

In pen-tests we see that in such file-less scenarios, many popular AV products still provide insufficient protection. Some newer business security products seem to be focusing on this problematic area now, and are closing the gap in some scenarios.

In the following tests, we focused on several different command-line stacks, CMD/PS commands, which download malware from the network directly into RAM (staged) or base64 encoded calls. Thus, we completely avoid disk access, which is usually (well) guarded by AV products. We sometimes use simple concealment measures or change the method of the stager call as well.

Once the malware has loaded its 2nd stage, an http/https connection to the attacker will be established. This inside-out mechanism has the advantage of establishing a C2 channel beyond the majority of NAT and firewall products to the attacker (http/https). Once the C2 tunnel has been established, the attacker can use all known functions of the common C2 products (Meterpreter, PowerShell Empire). These include e.g. file uploads/downloads, screenshots, keylogging, Windows shell, and webcam snapshots.

If we could establish such a C2 channel in these test scenarios, we rated "✗". If the AV software prevented this infection, we gave a "✓". If the AV client generally blocks remote access via WMI or PSexec on the client firewall, we assign an "🛡".

Our C2 receivers listened on the same IP address but different ports with different products on these ports.

All the tools used are freely available. Their source code is open and created for research purposes. Bad guys, however, often abuse these tools for criminal purposes. As shown here, only inadequate protection by security products is expected.

Results

Below are the results for the 25 scenarios described on the next pages.

	Test scenarios																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Score
1. Bitdefender	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	23
Kaspersky Lab	✓	✓	🛡	✓	🛡	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	23
2. CrowdStrike	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	18
3. Sophos	✗	✗	🛡	🛡	🛡	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	✗	12
4. McAfee	✗	✗	🛡	✓	🛡	✗	✗	✗	✗	✓	✗	✓	✓	✗	✓	✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	11
Symantec	✗	✗	✗	✗	🛡	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	11
5. Carbon Black	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	🛡	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	5
6. SentinelOne	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	2

✓ malware alerted / system protected - no session [1 point]

🛡 no alerts, but no C2-session (due to firewall, etc.) [1 point]

✗ malware not detected - C2-session [0 points]

Cylance had as default policy the „global script blocking“ activated, due to which most cases (except test scenario 14) got blocked (score: 24). Please note that by having "global script blocking" activated, also all clean / innocent scripts are blocked by default (we tested it with 25 clean scripts from Microsoft – only Cylance blocked all of them). As this is not making a distinction between clean and malicious scripts, they are not included in the table above.

Test scenarios used

Below the 25 test scenarios used in the PowerShell-based File-less Attacks and File-based Exploits Test are shortly described.

1.) Manual start mshta Meterpreter

Mshta launches an HTML application, in this case JavaScript code, which then tries to run PowerShell using the Windows Scripting Host (WScript). The PowerShell-Command downloads another PowerShell-Stager for Meterpreter from a network-host and loads it into memory.

2.) Manual start PowerShell > staged and not staged

Regardless of which process/exploit triggers the PowerShell command in a real-world attack (Office script, PDF exploits, called by other script interpreters, Rubber Ducky Keyboard Input, etc.), all attack vectors have one thing in common: they execute a PS command at some point. We test the command directly on command line, which gets a base64 encoded version of a Meterpreter-stager. This stager downloads the second stage from our C2-server and builds a C2 connection to the attacker. Additionally, we tried the same scenario but also staged, i.e. we also load the first stage from the website to keep the triggered payload as short as possible.

3.) Remote WMI Meterpreter

Remote access to a Windows machine via WMIC calls and payload as used in staged scenario 2.

4.) Remote psexec Meterpreter

Using the psexec interface, remote call the PowerShell payload used in staged scenario 2.

5.) Remote WMI pse

Another remote WMI payload similar to the one in Scenario 3. Here, however, a tunnel to a PowerShell Empire Server is initiated, instead of using the Metasploit framework.

6.) PowerLurk (WMI Event)

PowerLurk is a free tool to create WMI events, which in turn initiate the PowerShell Payload Stager. This persistence option is very hard to discover but only works with local admin privileges. Here we create a WMI event that triggers the launch of a process called "notepad.exe" and, for example, the payload from staged Scenario 2.

7.) PowerSploit Tasksched Persistence

Task scheduler jobs are also targets of popular attacks. Payloads hereby only need minimal user rights to persist on systems. This is similar to the registry autorun call.

8.) WMI subscription (PowerSploit)

In this test, a batch file launches a PowerShell command which downloads the first stage of the test, which is a PowerShell command to elevate privileges for the second stage (which is also a PowerShell script), containing the main test code itself. Invoke-WmiCommand executes a PowerShell ScriptBlock on a target computer using WMI. It does this by using the StdRegProv WMI registry provider methods to store a payload into a registry value. The command is then executed on the victim system and the output is stored in another registry value that is then retrieved.

9.) WMI subscription (PowerLurk)

In this test, a batch file launches a PowerShell command which downloads the first stage of the test, which is a PowerShell command to elevate privileges for the second stage (which is also a PowerShell script), containing the main test code itself (which is PowerLurk). It takes a command as the action, and an event for triggering this action, then creates the WMI element so as to have a fully functional Permanent WMI Event Subscription.

10.) Scheduled task (manually)

In this test, we combined task scheduler and Pupy. Pupy is an open source, cross-platform (Windows, Linux, OSX, Android) remote administration and post-exploitation tool mainly written in Python.

11.) Scheduled task (delivering with Firefox exploit)

This test is very similar to the previous one, the only difference being that the command execution is caused by an exploit in Firefox 31.0. To exploit the known vulnerability in this browser we used Metasploit.

12.) PowerShell script (Doc, PowerSploit ReflectivePEinjection)

We created a Microsoft Word doc file with an auto-open macro in it. When the document is opened, a VBS script launches a PowerShell command that downloads the first stage of the test, which is also a PowerShell, command to elevate privileges for the second stage, another PowerShell script, containing the PowerSploit ReflectivePEinjection itself. This script also downloads a base64 encoded DLL (decoded during execution). This DLL was built via Metasploit.

13.) PowerShell script (Doc, Pupy PowerShell)

This test is very similar to the previous one, the only difference is the payload, which is a Pupy stager.

14.) Eternalblue FuzzyBunch

In this test, we attack the test machine with FuzzyBunch (the leaked NSA toolset) through the network (TCP/port 445). After successfully exploiting the test machine, the Peddlecheap.dll component is uploaded and then connected to the newly exploited machine with Peddlecheap.

15.) Manual start: mshta (command exec)

In this test, mshta launches an HTML application - in this case VBS code - which then tries to run PowerShell using the Windows Scripting Host (WScript). The PowerShell-Command gets another PowerShell-Stager for elevated privileges and then executes some malicious commands.

16.) Manual start: LNK (command exec)

A .lnk file contains a PowerShell-Command to gets another PowerShell-Stager for elevated privileges then execute some malicious commands.

17.) Manual start: script (command exec)

A script file (such as JS/VBS) contains a tiny VBS code to execute a PowerShell-Command using the Windows Scripting Host (WScript) to get another PowerShell-Stager for elevate privileges then execute some malicious commands.

18.) Meterpreter (macro)

We created a Microsoft Word .doc file with an auto-open macro in it. When the document is opened, VBS script launches a PowerShell command, which is a Meterpreter stager.

19.) Meterpreter (Firefox 31.0 exploit firefox_proxy_prototype)

To exploit a known vulnerability in Firefox 31.0, we used Metasploit with a payload to connect back to our C2 server.

20.) Meterpreter (Silverlight exploit)

To exploit a known Silverlight ActiveX component vulnerability in IE, we used Metasploit with a payload to connect back to our C2 server.

21.) PowerShell empire (Firefox 31.0 exploit firefox_proxy_prototype)

PowerShell Empire is a pure PowerShell post-exploitation agent commonly used by pentester and red teams, however it can also be used by cyber criminals with the same effect. In this test case, we used the same delivery technique as in the Meterpreter cases, but we replace the payload with a PowerShell empire stager one.

22.) PowerShell empire (batch stager)

We used the batch stager of PowerShell empire (which is a one-line command) to establish a connection with the C2 server and to get further stages of the tool.

23.) PowerShell empire (doc with macro stager)

In this test we used the “doc with macro” stager of PowerShell empire. As a result, we got a Microsoft Word .doc file with an auto-open macro in it. When the document is opened, a VBS script launches and executes the one-line PowerShell command which starts a PowerShell empire session with the C2 server.

24.) PowerShell empire (Firefox encrypted exploit (ECDH IronSquirrel)

In this test, we used the same Firefox exploit as in the previous test, and also used the same PowerShell Empire payload. However, using IronSquirrel in the delivering phrase, we hide the actual exploit we are using against Firefox.

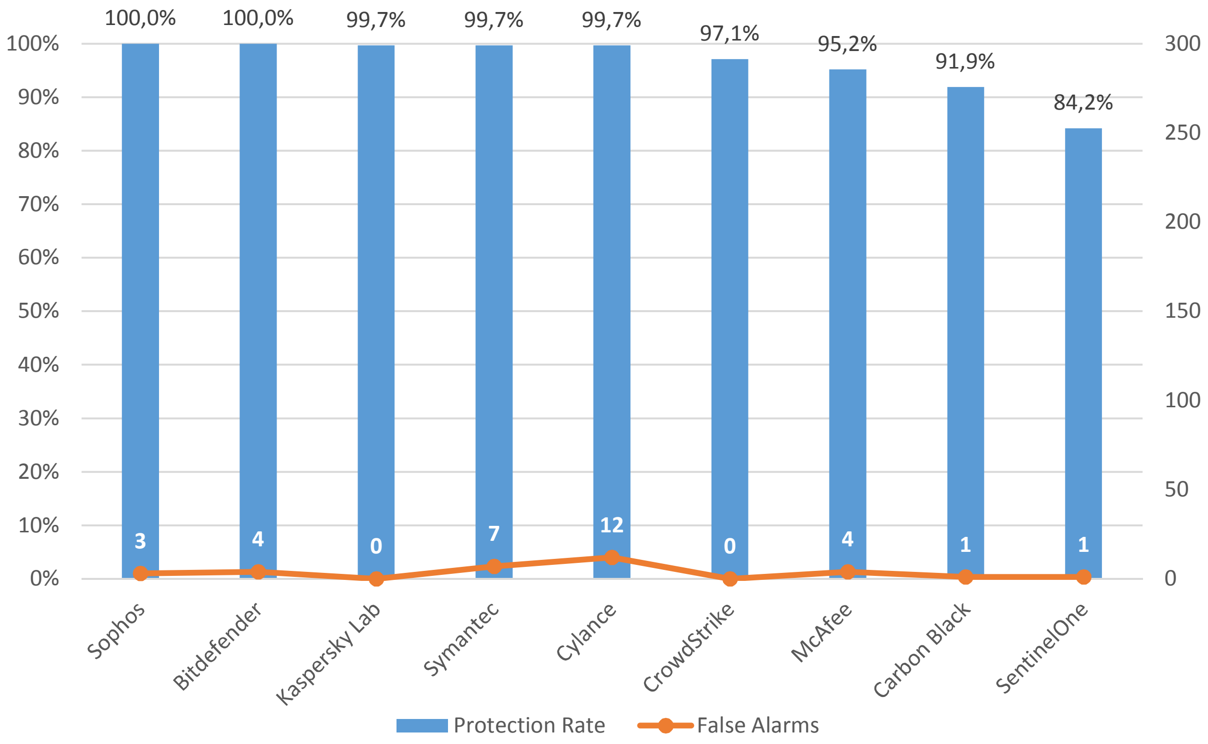
25.) PowerShell empire (SCT file stager)

In this test we used the SCT stager of PowerShell empire. This is a combination of command execution on the local host and abuse of regsvr32.dll and scrobj.dll, which are legitimate components of the Windows OS. The launcher script contains JavaScript code that executes the first stage of PowerShell empire.

Real-World Protection Test

The results below are based on a test set of 300 live test cases, i.e. malicious URLs found in the field, which were tested using our Real-World Testing Framework between the 10th November 2017 and the 30th November 2017. Additionally, 10 malicious test cases came from the email vector (emails with malicious attachments and emails containing links to malicious websites). The same infection vectors were used as a typical user would experience in everyday life. The test cases used cover a wide range of current malicious sites, and provide insights into the protection given by the various products (using all their protection features) while surfing the web. Using the same methodology, a false alarm test with 300 clean test cases was performed.

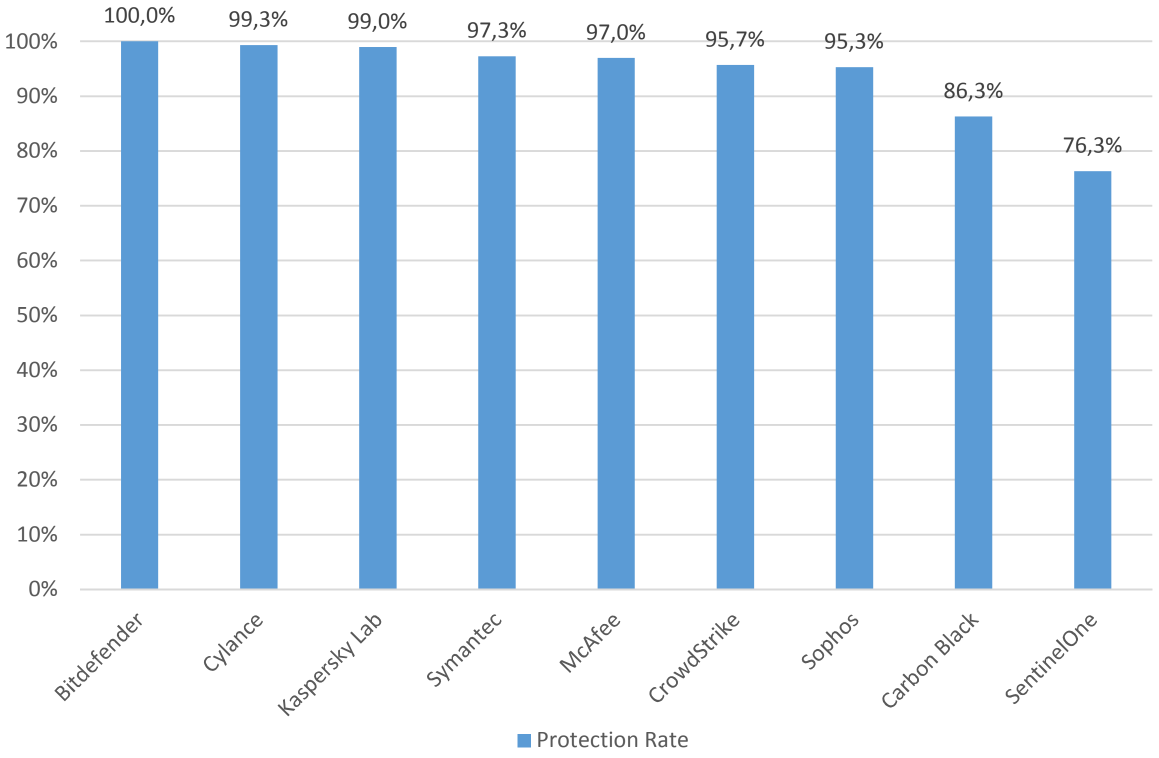
	Protection Rate	False Alarms
1. Sophos	100%	3
2. Bitdefender	100%	4
3. Kaspersky Lab	99.7%	0
4. Symantec	99.7%	7
5. Cylance	99.7%	12
6. CrowdStrike	97.1%	0
7. McAfee	95.2%	4
8. Carbon Black	91.9%	1
9. SentinelOne	84.2%	1



Ransomware Test

In order to measure how good the products are at detecting and blocking ransomware, we tested against a variety of new ransomware samples, in total 300 test cases, between 22nd November 2017 and 16th December 2017.

	Protection Rate
1. Bitdefender	100%
2. Cylance	99.3%
3. Kaspersky Lab	99.0%
4. Symantec	97.3%
5. McAfee	97.0%
6. CrowdStrike	95.7%
7. Sophos	95.3%
8. Carbon Black	86.3%
9. SentinelOne	76.3%



Copyright and Disclaimer

Any use of the results, etc. in whole or in part, is ONLY permitted after the explicit written agreement of the management board of AV-Comparatives, prior to any publication. AV-Comparatives and its testers cannot be held liable for any damage or loss, which might occur as result of, or in connection with, the use of the information provided in this paper. We take every possible care to ensure the correctness of the basic data, but a liability for the correctness of the test results cannot be taken by any representative of AV-Comparatives. We do not give any guarantee of the correctness, completeness, or suitability for a specific purpose of any of the information/content provided at any given time. No one else involved in creating, producing or delivering test results shall be liable for any indirect, special or consequential damage, or loss of profits, arising out of, or related to, the use or inability to use, the services provided by the website, test documents or any related data.

For more information about AV-Comparatives and the testing methodologies, please visit our website.

AV-Comparatives (March 2018)