application control rules for defending against advanced attacks by malware and HAckers

By Gleb Glazkov

<https://twitter.com/Gl3bGl4z> | [glebglaz@protonmail.com](mailto:glebglaz@protonmail.com) | <https://github.com/Gl3bGl4z/>

Supported product: Symantec Endpoint Protection 14+

App control rules: Version: (6 GTA)

Table of contents

[Intro 2](#_Toc14887590)

[Use cases: 3](#_Toc14887591)

[Notes: 3](#_Toc14887592)

[Download: 4](#_Toc14887593)

[Features: 4](#_Toc14887594)

[Recommendations for use: 5](#_Toc14887595)

[Known possible issues: 6](#_Toc14887596)

[Rules: 7](#_Toc14887597)

[Adversary simulation using Endgame RTA 8](#_Toc14887598)

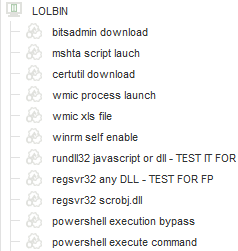
[Testing conditions: 9](#_Toc14887599)

[You Cant fully disable the "Suspicious Behavior Detection" feature but you can change it to "prompt": 10](#_Toc14887600)

[Reference 13](#_Toc14887601)

# Intro

Big changes were made to the new version after a lot of feedbacks that I got from a friend who has implemented the rules in a big organization that tested my last *version number 5 and got lots of false positives.*



The big changes in this version are:

The testing/implementation flexibility by creating a rules for each windows executable.

The use of arguments in case complete (ZERO TRUST) blocking of executables is not an option, which lowers the false positive rates.

*The friend works in a highly discrete defense company so the credit will unfortunately stay untold ;)*

**Thanks friend!**

# Use cases:

* ZERO TRUST methodology implementation in endpoint protection.
* Can be used as a secondary security control for prevention or monitoring.
* For testing EDR/ATP/EPP products against malware samples or RED teaming.
* For stopping/monitoring RED teaming activity.
* For enriching SIEM/MSSP/MDR/SOC logs.

# Notes:

* Added PWSH.exe (PowerShell Core).
* Added Powershell\_ISE.exe (Editor).
* Script types currently include: scr, pif, ps1, bat, vba/vbs.
* The 6 (GA) version will be tested before release against "Atomic Red by Red Canary" or/and "RTA by Endgame" which are adversary simulation frameworks.

<https://github.com/endgameinc/RTA>

<https://atomicredteam.io/>

* The 6 (GA) version will be released after getting community inputs for false positives and other suggestions.
* Version 7 (GA) will be compared to "MITRE ATT&K" matrix. <https://attack.mitre.org/>
* I highly recommend reviewing the new version of the official Symantec "**Hardening Endpoint Protection with an Application and Device Control Policy to increase security**" - <https://support.symantec.com/us/en/article.tech132337.html>

It adds many useful protections.

The most recommended by me are:

**Prevents changes to Windows Shell load points (HIPS) [AC12]**

**Block applications from running out of the recycle bin [AC-25]**

**HIGH FALSE POSITIVE COUNT:**

**Prevents changes to system using browser and office products (HIPS) [AC13]**

**Prevent vulnerable Windows processes from writing code [AC17]​**

# Download:

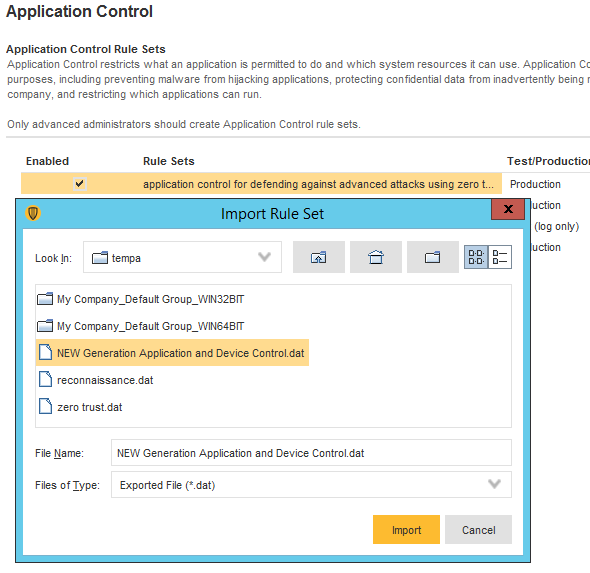
<https://github.com/Gl3bGl4z/SEP_Application_Control>

# Features:

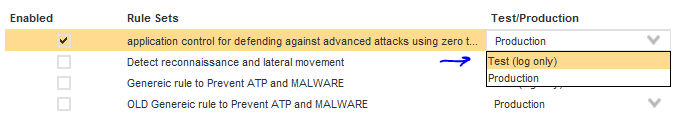
* "ZERO TRUST" methodology – Block high risk LOLBINS(Living-Off=The-Land-Binaries).
* Detect lateral movement.
* Detect reconnaissance commands.
* Prevent OFFICE exploitation for executing commands or scripts.
* Prevent Internet Browser exploitation for executing commands or scripts.
* Prevent popular 3rd party apps exploitation for executing commands or scripts.

# Recommendations for use:

1. Import the rules to SEPM



1. Put in TEST mode



1. Review the "Microsoft recommended blacklist" for executables that may be used in your organizations. <https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/microsoft-recommended-block-rules>
2. Assign the policy to one testing endpoint or a small test group of testing endpoints.

You may want to create a new group to which you will move the test endpoints.

1. Watch over the logs for false positives.
2. In case of false positives. you have the following options:

* Delete a process/folder/file from the policy.
* Add an exclusion.
* Delete the rule.

I personally recommend adding an explicit exclution as much precise as possible. For example: not "myprogram.exe" but rather "%pragramfiles%\myprogram\myprogram.exe".

* Edit the rule "compromised CMD" 🡪 "known legitimate folder or file"
  + If you are **not** using Forescout NACm delete the following entries:

%windir%\Temp\fstmpsc\\*

C:\Users\\*\AppData\Local\Temp\fstmpsc\\*

C:\Users\\*\AppData\Local\Temp\fstmp\\*

%windir%\Temp\fstmp\\*

* + If you are **not** using logon scripts, delete the following entries:

[\\\\DOMAIN\.LOCAL\\[^\]\*](file:///\\\\DOMAIN\.LOCAL\\%5b%5e\%5d*)

[\\DOMAIN.LOCAL\sysvol\DOMAIN.LOCAL\Policies\{????????-????-????-????-????????????}\User\Scripts\Logon\\*.bat](file:///\\DOMAIN.LOCAL\sysvol\DOMAIN.LOCAL\Policies\%7b%3f%3f%3f%3f%3f%3f%3f%3f-%3f%3f%3f%3f-%3f%3f%3f%3f-%3f%3f%3f%3f-%3f%3f%3f%3f%3f%3f%3f%3f%3f%3f%3f%3f%7d\User\Scripts\Logon\*.bat)

[\\DOMAIN.LOCAL\SysVol\DOMAIN.LOCAL\Policies\\*.bat](file:///\\DOMAIN.LOCAL\SysVol\DOMAIN.LOCAL\Policies\*.bat)

* + If you are **not** using VNWARE View, delete the following entries:

C:\Program Files\VMware\VMware View\Agent\DCT\\*.bat

C:\Program Files\VMware\VMware View\Agent\DCT\\*.vbs

1. After no false positives have been seen, move for testing the policies on more endpoints.

The testing endpoints should represent as much versatility as possible to represent the company/organization. That means – operating systems, software user types.

1. After enough endpoints have been tested which represent most of the endpoint types in the organization, you may start moving rules to "block" while taking a grace period between each rule change.
2. Enable sending mail alerts on block events so you can quickly know about a problem that is chased by an application control rule on a new endpoint for different reasons.
3. After all rules have been moved to "block" you may start a slow migration process by adding more endpoints the group in which your new application control rule is assigned.

# Known possible issues:

Many rules use REGEX for finding arguments, it my impact endpoint system resource consumption.

If you see an impact try disabling some rules for troubleshooting.

All rules have logging enabled, which may impact disk IO performance on endpoint and other LOG collection mechanisms.

# Rules:

# Adversary simulation using Endgame RTA

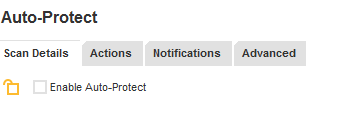
Failed 🡺 attack failed

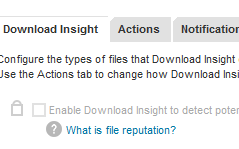
Executed 🡺 attack was successful

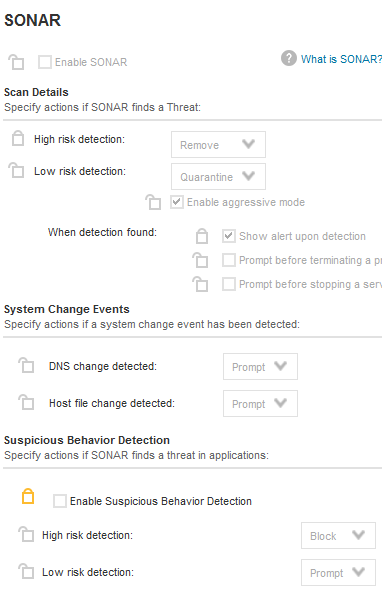
Results:

|  |  |
| --- | --- |
| at\_command | failed |
| certutil\_file\_obfuscation | executed |
| certutil\_webrequest | failed |
| delete\_catalogs | executed |
| delete\_usnjrnl | executed |
| delete\_volume\_shadow | failed |
| disable\_windows\_firewall | executed |
| enum\_commands | failed |
| findstr\_pw\_search | executed |
| installutil\_network | failed |
| Lateral\_commands | failed |
| msbuild\_network | failed |
| mshta\_network | failed |
| msiexec\_http\_installer | executed |
| msxsl\_network | executed |
| net\_user\_ad | failed |
| office\_applicaiton\_startup | failed |
| persistent\_scripts | failed |
| run task | failed |
| powershell\_args | failed |
| process\_extension\_anomalies | partial execution |
| Processname\_masquerade | executed |
| recyclebin\_process | executed, can be blocked by official symantec rule |
| registry hive export | failed |
| registry\_persistence\_create | failed |
| regsvr32\_scrobj.y | failed |
| rundll32+inf\_callback | executed |
| rundll32\_ordinal | failed |
| schtask\_escalation | failed |
| scrobj\_com\_hijack | executed |
| sip\_provider | failed |
| smb\_connection | executed |
| suspicious\_officE\_children | executed but real app will be blocked |
| system\_restore\_process\_evasion | failed |
| trust\_provider | failed |
| uac\_eventviewer | failed |
| uca\_sdclt | failed |
| unusual\_ms\_tool\_network | failed |
| unusual\_parent | executed |
| unusual\_process\_path | failed |
| user\_dir\_escalation | failed |
| wevtutil\_log\_clear | failed |
| wmi\_tool\_execution | failed |

## Testing conditions:

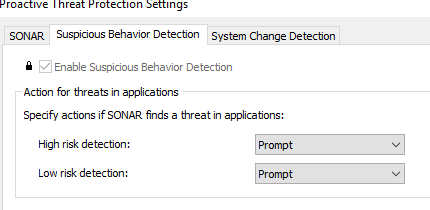


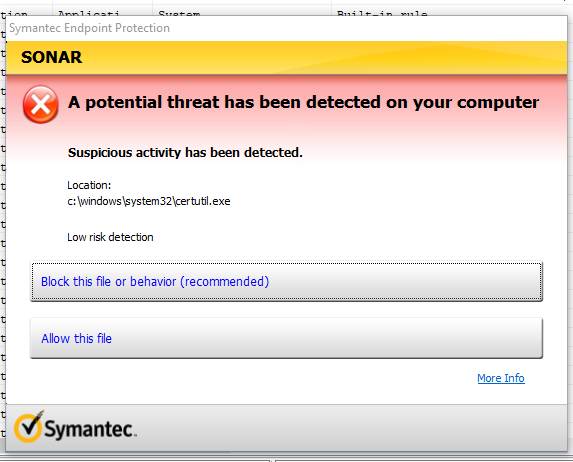


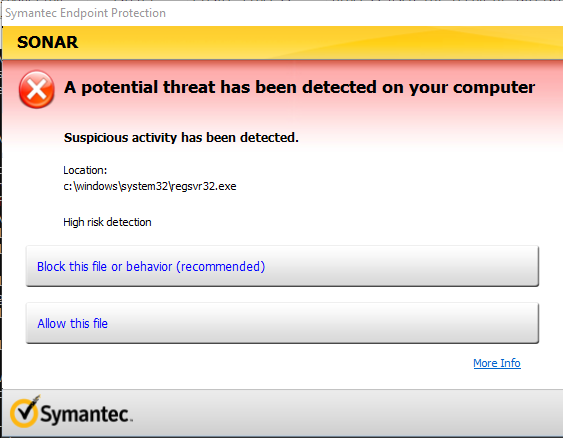


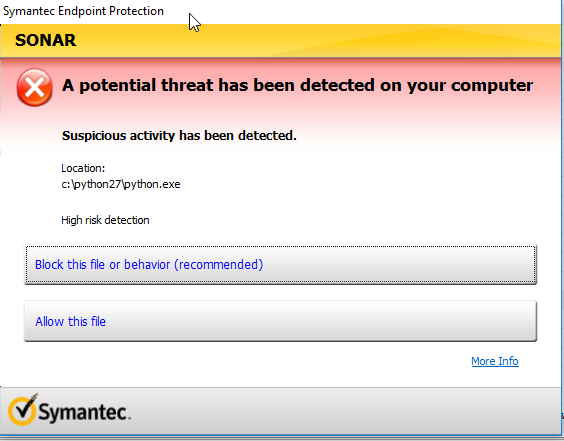


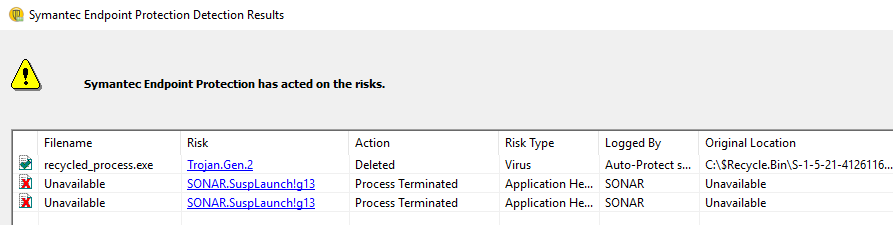
## You Cant fully disable the "Suspicious Behavior Detection" feature but you can change it to "prompt":











# Reference

<https://mgreen27.github.io/posts/2018/02/18/Sharing_my_BITS.html>

<https://bytesoverbombs.io/living-off-the-land-windows-command-line-downloading-without-powershell-6b3a2b8acd97>

<https://github.com/pwndizzle/CodeExecutionOnWindows>

<https://blog.netspi.com/15-ways-to-bypass-the-powershell-execution-policy/>

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Download%20and%20Execute.md>

<https://support.symantec.com/us/en/article.tech188597.html>

<https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/microsoft-recommended-block-rules>

<https://github.com/endgameinc/RTA>

<https://github.com/LOLBAS-Project/LOLBAS>

<https://gtfobins.github.io/>

<https://github.com/xapax/security/blob/master/privilege_escalation_windows.md>

Recommended read:

<https://support.symantec.com/us/en/article.tech132337.html>