

Introduction to Sigma



Chris Peacock – Principal Detection Engineer

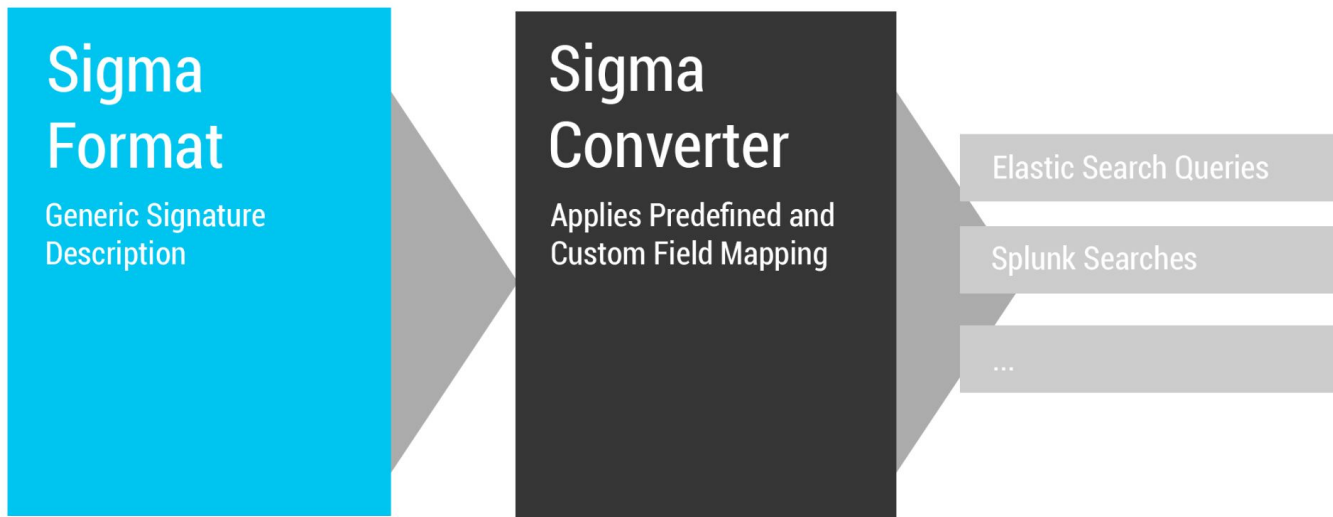


- Network Engineer
- SOC Analyst
- Threat Hunter
- Detection Engineer
- CTI Analyst
- Incident Responder
- Purple Team Lead
- GCTI, GCFA, GCED
- MITRE ATT&CK Contributor
- Sigma Contributor
- LOLBAS Contributor



What is Sigma?

- “Sigma is for log files what Snort is for network traffic and YARA is for files.”

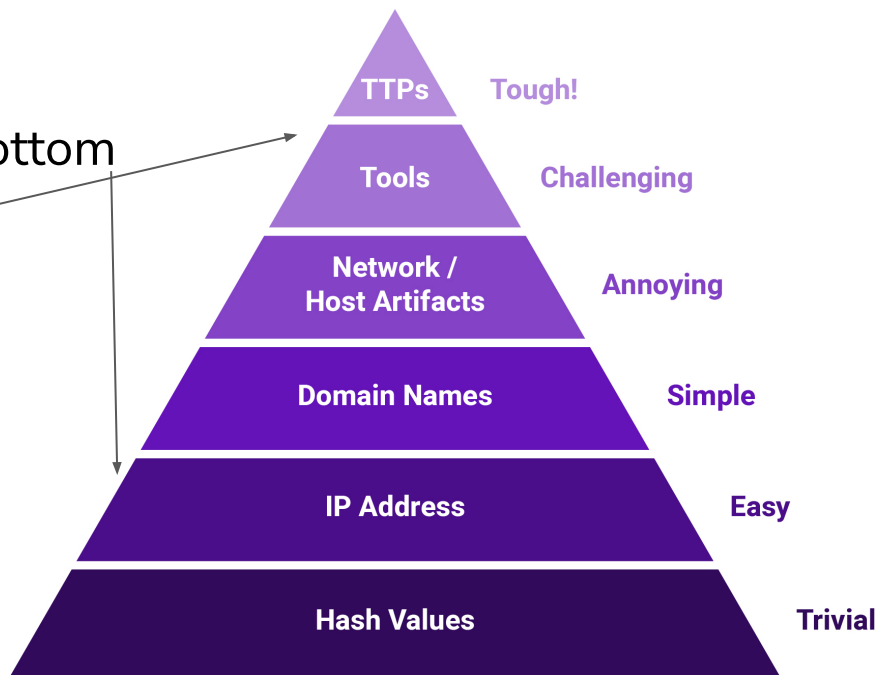


<https://github.com/SigmaHQ/sigma>

Why Sigma?

David Bianco: <http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html>

Started at the bottom
now I'm here



Why Sigma Example

IcedID Initial Discovery			
	Procedure	Alert	Alert Level & Notes
1	ipconfig /all	✗	<ul style="list-style-type: none">• No Alert• One Sigma Recommendation
2	systeminfo	✗	<ul style="list-style-type: none">• No Alert• One Sigma Recommendation
3	whoami /groups	✓	<ul style="list-style-type: none">• Low Alert• Tune if needed & Raise Alert Level• Two Sigma Recommendations
4	net config workstation	✗	<ul style="list-style-type: none">• No Alert• One Sigma Recommendation
5	net use	✗	<ul style="list-style-type: none">• No Alert• One Sigma Recommendation



Sigma Solves This

Internal Reconnaissance

In the Internal Reconnaissance stage, the intruder collects information about the victim environment. Like most APT (and non-APT) intruders, APT1 primarily uses built-in operating system commands to explore a compromised system and its networked environment. Although they usually simply type these commands into a command shell, sometimes intruders may use batch scripts to speed up the process. Figure 18 below shows the contents of a batch script that APT1 used on at least four victim networks.

```
@echo off
ipconfig /all>>"C:\WINNT\Debug\1.txt"
net start>>"C:\WINNT\Debug\1.txt"
tasklist /v>>"C:\WINNT\Debug\1.txt"
net user >>"C:\WINNT\Debug\1.txt"
net localgroup administrators>>"C:\WINNT\Debug\1.txt"
netstat -ano>>"C:\WINNT\Debug\1.txt"
net use>>"C:\WINNT\Debug\1.txt"
net view>>"C:\WINNT\Debug\1.txt"
net view /domain>>"C:\WINNT\Debug\1.txt"
net group /domain>>"C:\WINNT\Debug\1.txt"
net group "domain users" /domain>>"C:\WINNT\Debug\1.txt"
net group "domain admins" /domain>>"C:\WINNT\Debug\1.txt"
net group "domain controllers" /domain>>"C:\WINNT\Debug\1.txt"
net group "exchange domain servers" /domain>>"C:\WINNT\Debug\1.txt"
net group "exchange servers" /domain>>"C:\WINNT\Debug\1.txt"
net group "domain computers" /domain>>"C:\WINNT\Debug\1.txt"
```

FIGURE 18: An APT1 batch script that automates reconnaissance

1.5 . 2 . net domain_controllers < ===== this command will show the ip addresses of domain controllers

1.6 . shell net localgroup administrators <===== local administrators

1.7 . shell net group / domain "Domain Admins" <===== domain administrators

1.8 . shell net group "Enterprise Admins" / domain <===== enterprise administrators


1.9 . the shell net group "the Domain Computers has" / domain <===== total number - in the PC in the domain

1.10 . net computers < ===== ping all hosts with the output of ip addresses.



Parts of Sigma

sigma / rules / windows / process_creation / proc_creation_win_regsvr32_remote_share.yml

 nasbench feat: multiple fixes and updates

63888f7 · 8 months ago

Code Blame 25 lines (25 loc) · 787 Bytes

Raw

```
1 title: Suspicious Regsvr32 Execution From Remote Share
2 id: 88a87a10-384b-4ad7-8871-2f9bf9259ce5
3 status: experimental
4 description: Detects REGSVR32.exe to execute DLL hosted on remote shares
5 references:
6   - https://thefirreport.com/2022/10/31/follina-exploit-leads-to-domain-compromise/
7 author: Nasreddine Bencherchali (Nextron Systems)
8 date: 2022/10/31
9 tags:
10   - attack.defense_evasion
11   - attack.t1218.010
12 logsource:
13   category: process_creation
14   product: windows
15 detection:
16   selection_img:
17     - Image|endswith: '\regsvr32.exe'
18     - OriginalFileName: '\REGSVR32.EXE'
19   selection_cli:
20     CommandLine|contains: ' \\\\'
21   condition: all of selection_*
22 falsepositives:
23   - Unknown
24 # Decrease to medium if this is something common in your org
25 level: high
```

https://github.com/SigmaHQ/sigma/blob/4b277429c16c21c0191792147bf21169c902b79/rules/windows/process_creation/proc_creation_win_regsvr32_remote_share.yml



Need a GUID?

← → ↻ guidgenerator.com/online-guid-generator.aspx

Online GUID / UUID Generator

How many GUIDs do you want (1-2000):

Format: ☐ Uppercase ☐ {} Braces ☒ Hyphens

Encoding: ☐ Base64 ☐ RFC 7515 ☐ URL encode

Generate some GUIDs!

Results:

```
6bb08bbc-d0ba-45f7-a582-ebb41746f5e2
```

Use these GUIDs at your own risk! No guarantee of their uniqueness or suitability is given or implied.

Permalink is your friend!

sigma / rules / windows / process_creation / proc_creation_win_regsvr32_remote_share.yml

nasbench feat: multiple fixes and updates

Code Blame 25 lines (25 loc) · 787 Bytes

```
1 title: Suspicious Regsvr32 Execution From Remote Share
2 id: 88a87a10-384b-4ad7-8871-2f9bf9259ce5
3 status: experimental
4 description: Detects REGSVR32.exe to execute DLL hosted on remote shares
5 references:
6   - https://thedfirreport.com/2022/10/31/follina-exploit-leads-to-domain-comprom
7   author: Nasreddine Bencherchali (Nextron Systems)
8   date: 2022/10/31
9   tags:
10    - attack.defense_evasion
11    - attack.t1218.010
12 logsource:
13   category: process_creation
14   product: windows
15 detection:
16   selection_img:
17     - Image|endswith: '\\regsvr32.exe'
18     - OriginalFileName: '\\REGSVR32.EXE'
19   selection_cli:
20     CommandLine|contains: ' \\\\'
21   condition: all of selection_*
22 falsepositives:
23   - Unknown
24 # Decrease to medium if this is something common in your org
25 level: high
```

Raw file content

Download ⌘ shift s

Jump to line 1

Copy path ⌘ shift .

Copy permalink ⌘ shift ,

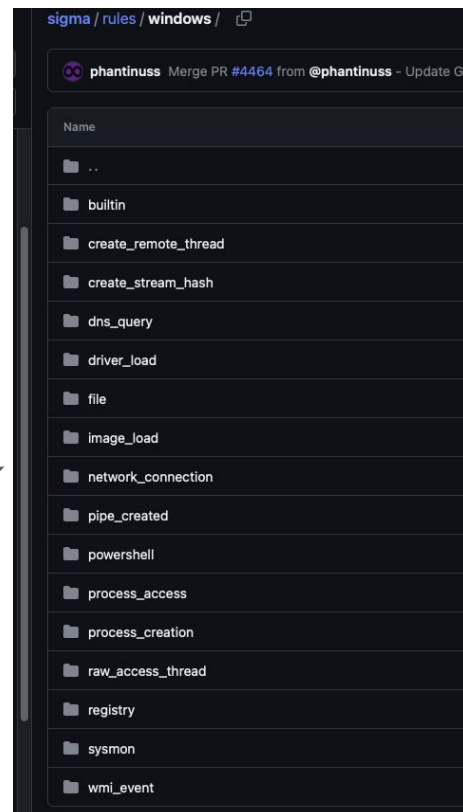
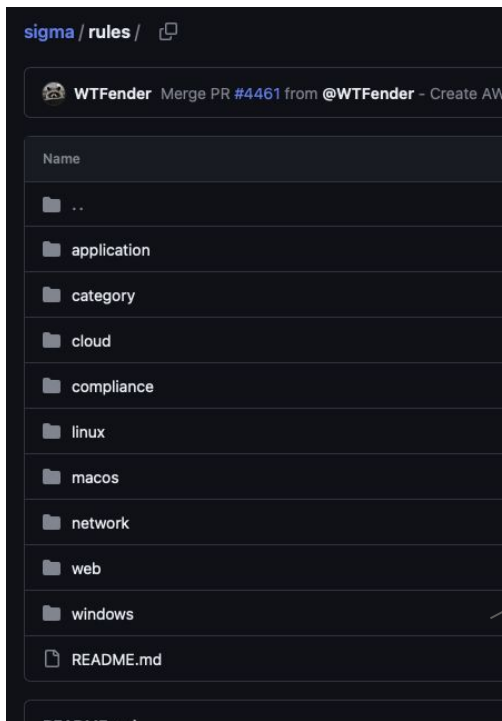
View options

- ✓ Show code folding buttons
- Wrap lines
- Center content
- ✓ Open symbols on click

https://github.com/SigmaHQ/sigma/blob/43277f26fc1c81fc98fc79147b711189e901b757/rules/windows/process_creation/proc_creation_win_regsvr32_remote_share.yml



Buckets on Buckets



<https://github.com/SigmaHQ/>

IcedID Continued & Placeholder Discovery

Procedure		Alert	Level & Sigma
6	cmd /c echo %userdomain%	✗	<ul style="list-style-type: none">• No Alert• Engineer custom alerts for<ul style="list-style-type: none">◦ "echo <my_domain_name_here>"◦ "/c"
7	nltest /domain_trusts	✗	<ul style="list-style-type: none">• No Alert• One Sigma Recommendation
8	nltest /domain_trusts /all_trusts	✗	<ul style="list-style-type: none">• No Alert• Two Sigma Recommendations
9	net view /all /domain	✓	<ul style="list-style-type: none">• Low Alert• Change to High/Critical• Two Sigma Recommendations
10	net view /all	✓	<ul style="list-style-type: none">• Low Alert• Change to High/Critical• Two Sigma Recommendations



Placeholder Rules

sign /rules-placeholder / windows / process_creation / proc_creation_win_userdomain_variable_enumeration.yml

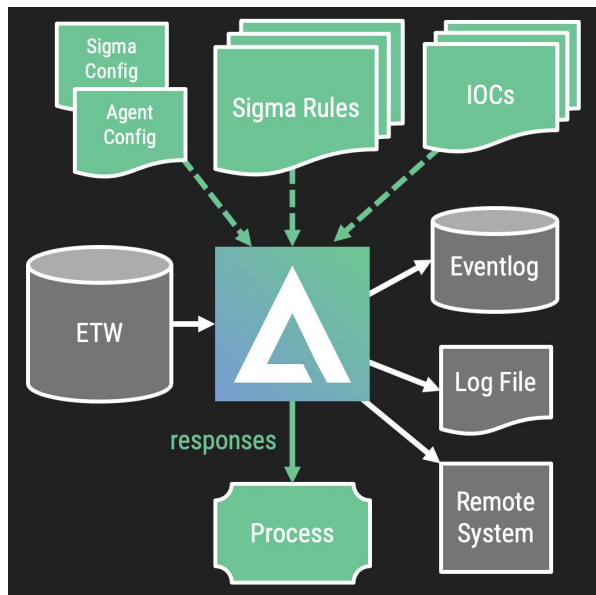
nasbench fix: some stylistic issues ✓ 6623dec · 8 months ago History

Code Blame 24 lines (24 loc) · 745 Bytes Raw Copy Download

```
1 title: Userdomain Variable Enumeration
2 id: 43311e65-84d8-42a5-b3d4-c94d9b67038f
3 status: experimental
4 description: Detects suspicious enumeration of the domain the user is associated with.
5 references:
6   - https://www.arxiv-vanity.com/papers/2008.04676/
7   - https://thedfirreport.com/2022/11/14/bumblebee-zeros-in-on-meterpreter/
8 author: 'Christopher Peacock @SecurePeacock, SCYTHE @scythe_io'
9 date: 2023/02/09
10 tags:
11   - attack.discovery
12   - attack.t1016
13 logsource:
14   category: process_creation
15   product: windows
16 detection:
17   selection:
18     CommandLine|contains|all:
19     - 'echo '
20     - '%userdomain%'
21   condition: selection
22 falsepositives:
23   - Certain scripts or applications may leverage this.
24 level: low
```

Is there a Sigma for that?

- Ask Aurora. “The AURORA Agent is a lightweight and customisable EDR agent based on Sigma.” -Nextron Systems



<https://www.nextron-systems.com/aurora/>

Aurora Dashboard

The screenshot displays the Aurora Dashboard interface. On the left is a sidebar with navigation links: Dashboard, Status, Settings, Documentation, and About Aurora. The main content area shows a table of findings under the 'Overview' tab. The table has columns for Level, Message, Title, Description, Match, Image, and Author. Three findings are listed, all with a 'warning' level. A notification window for 'AuroraNotifier.exe' is overlaid on the bottom right, displaying the details of the first finding.

Level	Message	Title	Description	Match	Image	Author
warning	Sigma match found	Run Whoami Showing Privileges	Detects a whoami.exe executed with the /priv command line flag instructing the tool to show all current user privileges. This is often used after a privilege escalation attempt.	/priv in CommandLine \\whoami.exe in Image whoami.exe in OriginalFileName	C:\Windows\SysWOW64\whoami.exe	Florian Roth
notice	Sigma match found	Whoami Execution	Detects the execution of whoami, which is often used by attackers after exploitation / privilege escalation but rarely used by administrators	\\whoami.exe in Image whoami.exe in OriginalFileName	C:\Windows\SysWOW64\whoami.exe	Florian Roth
notice	Sigma match found	Whoami Execution	Detects the execution of whoami, which is often used by attackers after exploitation / privilege escalation but rarely used by administrators	\\whoami.exe in Image whoami.exe in OriginalFileName		

AuroraNotifier.exe

! Sigma: Run Whoami Showing Privileges
Level: warning
Description: whoami - displays logged on user information

Detection Rule License

Detection Rule License (DRL) 1.1

Permission is hereby granted, free of charge, to any person obtaining a copy of this rule set and associated documentation files (the "Rules"), to deal in the Rules without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Rules, and to permit persons to whom the Rules are furnished to do so, subject to the following conditions:

If you share the Rules (including in modified form), you must retain the following if it is supplied within the Rules:

1. identification of the authors(s) ("author" field) of the Rule and any others designated to receive attribution, in any reasonable manner requested by the Rule author (including by pseudonym if designated).
2. a URI or hyperlink to the Rule set or explicit Rule to the extent reasonably practicable
3. indicate the Rules are licensed under this Detection Rule License, and include the text of, or the URI or hyperlink to, this Detection Rule License to the extent reasonably practicable

If you use the Rules (including in modified form) on data, messages based on matches with the Rules must retain the following if it is supplied within the Rules:

1. identification of the authors(s) ("author" field) of the Rule and any others designated to receive attribution, in any reasonable manner requested by the Rule author (including by pseudonym if designated).

THE RULES ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE RULES OR THE USE OR OTHER DEALINGS IN THE RULES.

<https://github.com/SigmaHQ/sigma/blob/43277f26fc1c81fc98fc79147b711189e901b757/LICENSE.Detection.Rules.md>

Converting

Rule Usage


- Use [Sigma CLI](#) to convert your rules into queries.
- Use [pySigma](#) to integrate Sigma in your own toolchain or product.
- Check out the [legacy sigmatools](#) and [sigmac](#) if your target query language is not yet supported by the new toolchain. Please be aware that the legacy sigmatools are not maintained anymore and some of the backends don't generate correct queries.

Converting

Rule Usage

- Use [Sigma CLI](#) to convert your rules into queries.
- Use [pySigma](#) to integrate Sigma in your own toolchain or product.
- Check out the [legacy sigmatools](#) and [sigmac](#) if your target query language is not yet supported by the new toolchain. Please be aware that the legacy sigmatools are not maintained anymore and some of the backends don't generate correct queries.

Converting – Sigconverter.io

 sigconverter.io - sigma rule converter

Select target:
sentinelone

Select output format:
default

Select pipeline(s):

```
sigma convert --without-pipeline  
-t sentinelone -f default  
rule.yml
```

Copy

```
title: Suspicious SYSTEM User Process Creation  
id: 2617e7ed-adb7-40ba-b0f3-8f9945fe6c09  
status: test  
description: Detects a suspicious process creation as  
SYSTEM user (suspicious program or command line parameter)  
references:  
  - Internal Research  
  - https://tools.thehacker.recipes/mimikatz/modules  
author: Florian Roth (rule), David ANDRE (additional  
keywords)  
date: 2021/12/20  
modified: 2022/04/27  
logsource:  
  category: process_creation  
  product: windows  
detection:  
  selection:  
    IntegrityLevel: System  
    User|contains: # covers many language settings  
      - 'AUTHORI'  
      - 'AUTORI'  
  selection_special:  
    - Image|endswith:  
      - '\calc.exe'
```

```
EventType = "Process Creation" AND (EndpointOS = "windows"  
AND ((TgtProcIntegrityLevel = "System" AND (TgtProcUser  
containsCIS "AUTHORI" OR TgtProcUser containsCIS "AUTORI"))  
AND ((TgtProcImagePath endswithCIS "\calc.exe" OR  
TgtProcImagePath endswithCIS "\wscript.exe" OR  
TgtProcImagePath endswithCIS "\cscript.exe" OR  
TgtProcImagePath endswithCIS "\hh.exe" OR TgtProcImagePath  
endswithCIS "\mshta.exe" OR TgtProcImagePath endswithCIS  
"\forfiles.exe" OR TgtProcImagePath endswithCIS  
"\ping.exe") OR (TgtProcCmdLine containsCIS " -NoP " OR  
TgtProcCmdLine containsCIS " -W Hidden " OR TgtProcCmdLine  
containsCIS " -decode " OR TgtProcCmdLine containsCIS "  
/decode " OR TgtProcCmdLine containsCIS " /urlcache " OR  
TgtProcCmdLine containsCIS " -urlcache " OR TgtProcCmdLine  
= "*" -e* JAB*" OR TgtProcCmdLine = "*" -e* SUVYI*" OR  
TgtProcCmdLine = "*" -e* SQBFAFgA*" OR TgtProcCmdLine = "*" -  
e* aMV4I*" OR TgtProcCmdLine = "*" -e* IAB*" OR  
TgtProcCmdLine = "*" -e* PAA*" OR TgtProcCmdLine = "*" -e*  
aQBLAHgA*" OR TgtProcCmdLine containsCIS "vssadmin delete  
shadows" OR TgtProcCmdLine containsCIS "reg SAVE HKLM" OR  
TgtProcCmdLine containsCIS " -ma " OR TgtProcCmdLine  
containsCIS "Microsoft\Windows\CurrentVersion\Run" OR  
TgtProcCmdLine containsCIS ".downloadstring(" OR  
TgtProcCmdLine containsCIS ".downloadfile(" OR  
TgtProcCmdLine containsCIS " /find:" OR TgtProcCmdLine
```

They're in LOLBAS!

[/Rundll32.exe](#) ☆ Star 5,998

Execute Alternate data streams

Used by Windows to execute dll files

Paths:

C:\Windows\System32\rundll32.exe
C:\Windows\SysWOW64\rundll32.exe

Resources:

- <https://pentestlab.blog/2017/05/23/applocker-bypass-rundll32/>
- https://evi1cg.me/archives/AppLocker_Bypass_Techniques.html#menu_index_7
- <https://oddvar.moe/2017/12/13/applocker-case-study-how-insecure-is-it-really-part-1/>
- <https://oddvar.moe/2018/01/14/putting-data-in-alternate-data-streams-and-how-to-execute-it/>
- <https://bohops.com/2018/06/28/abusing-com-registry-structure-clsid-localserver32-inprocserver32/>
- <https://github.com/sailay1996/expl-bin/blob/master/obfus.md>
- <https://github.com/sailay1996/misc-bin/blob/master/rundll32.md>
- <https://nasbench.medium.com/a-deep-dive-into-rundll32-exe-642344b41e90>
- <https://www.cybereason.com/blog/rundll32-the-infamous-proxy-for-executing-malicious-code>

Acknowledgements:

- Casey Smith (@sublee)
- Oddvar Moe (@oddvarmoe)
- Jimmy (@bohops)
- Sailay (@404death)
- Martin Ingesen (@Mrtin)

Detection:

- Sigma: [sysmon_rundll32_net_connections.yml](#)
- Sigma: [win_susp_rundll32_activity.yml](#)
- Elastic: [defense_evasion_unusual_network_connection_via_rundll32.toml](#)
- IOC: Outbound Internet/network connections made from rundll32
- IOC: Suspicious use of cmdline flags such as -sta

Execute

AllTheThingsx64 would be a .DLL file and EntryPoint would be the name of the entry point in the .DLL file to execute.

```
rundll32.exe AllTheThingsx64,EntryPoint
```

<https://lolbas-project.github.io/lolbas/Binaries/Rundll32/>

Other Resources

- Rule Creation Guide:

<https://github.com/SigmaHQ/sigma/wiki/Rule-Creation-Guide>

- How to Write Sigma Rules:

<https://www.nextron-systems.com/2018/02/10/write-sigma-rules/>

Happy Huntin'

