KAZ HACK STAN 2025

SEPTEMBER 17–19
ALMATY

TSARKA · DIGITAL & SPACE MINISTRY · FREEDOM HOLDING CORP. · TREND MICRO · astana hub

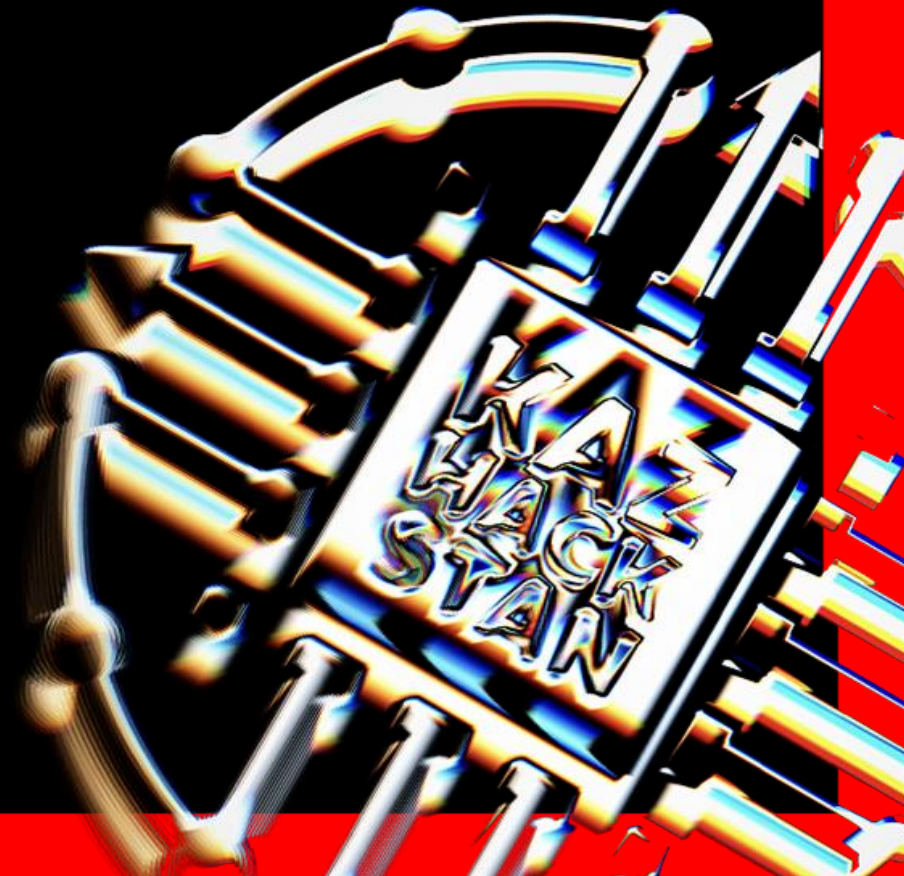
HACKDAY

Will It Run?

Fooling EDRs with command lines using empirical data

SPEAKER

Wietze Beukema



Wietze Beukema

Lead Threat Detection & Response Engineer

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

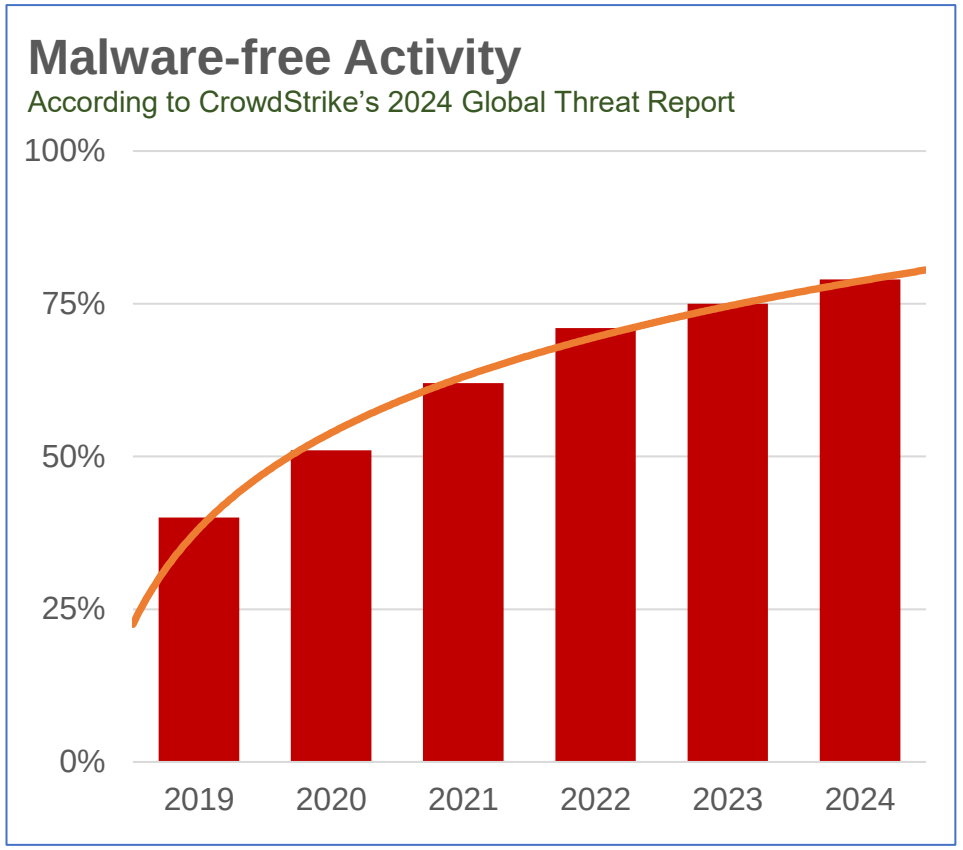
01

Command-line obfuscation

Changing landscape

Increased use of legitimate tools
- Built-in scripting tools
- LOLBAS
- Legitimate 3rd party tools

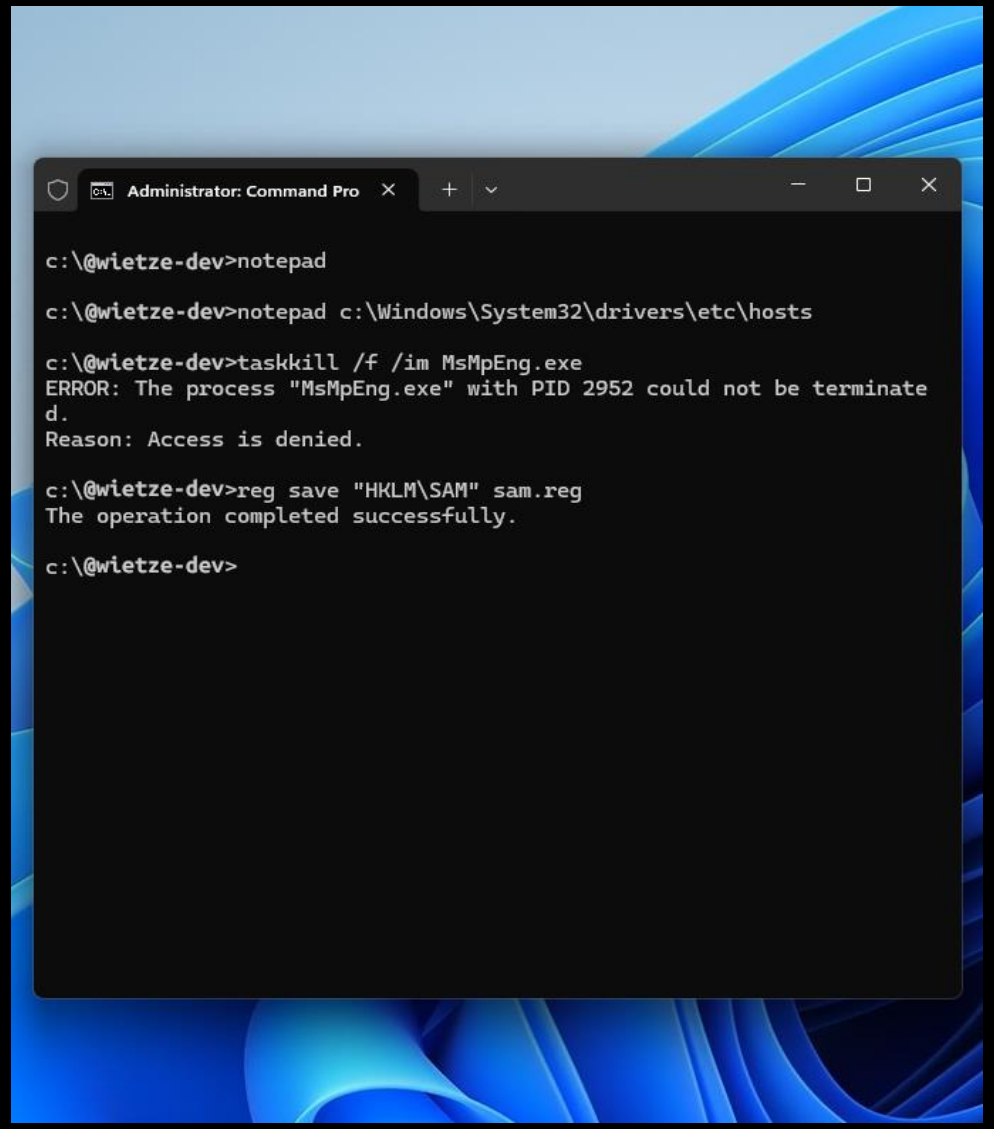
Challenges
- Blending in with normal use
- Not setting off detections (?)



HACKDAY

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

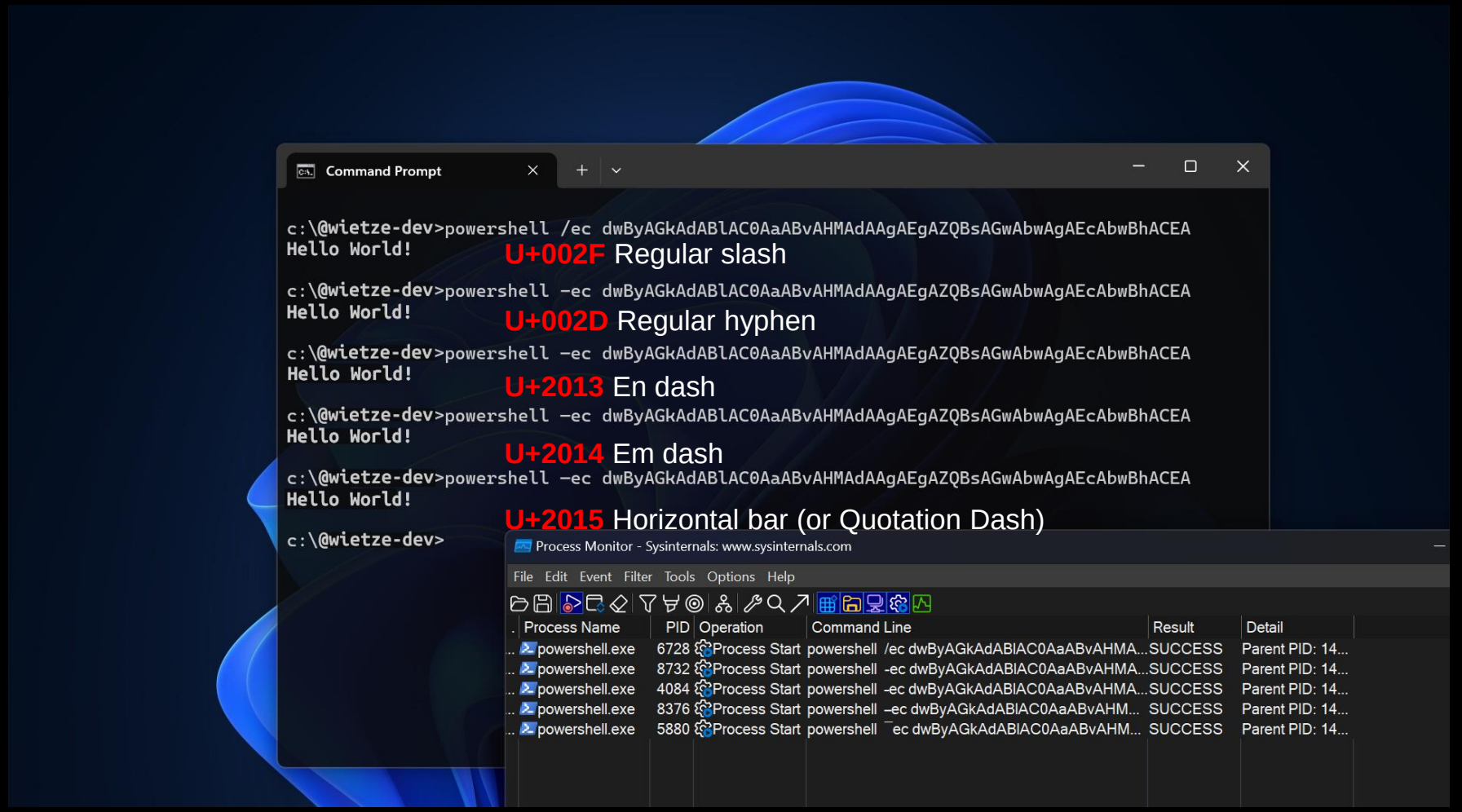


Command-Line Obfuscation is the masquerading of the true intention of a command you are trying to run.

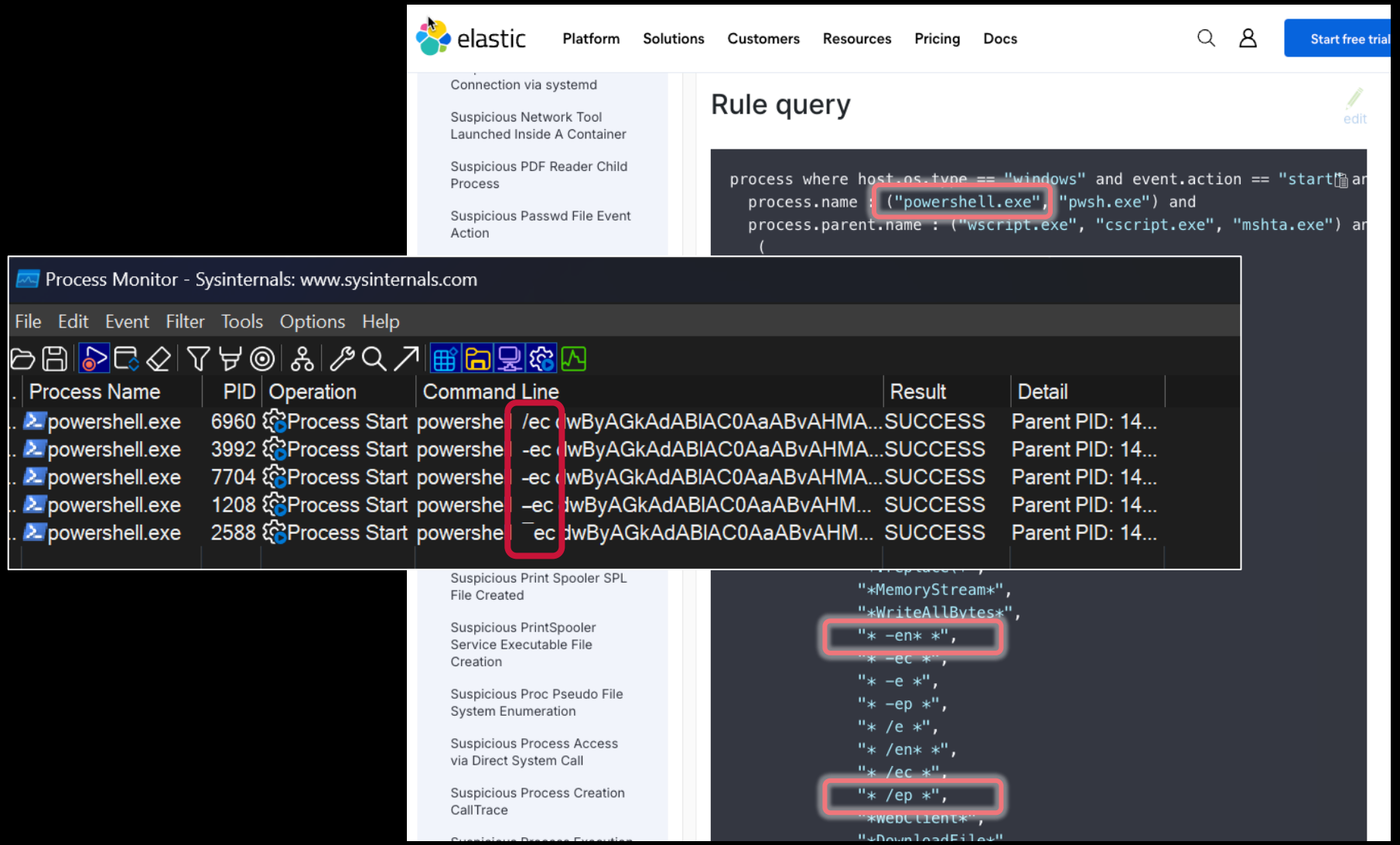
Command-Line Obfuscation variants

Option Char Substitution

HACKDAY



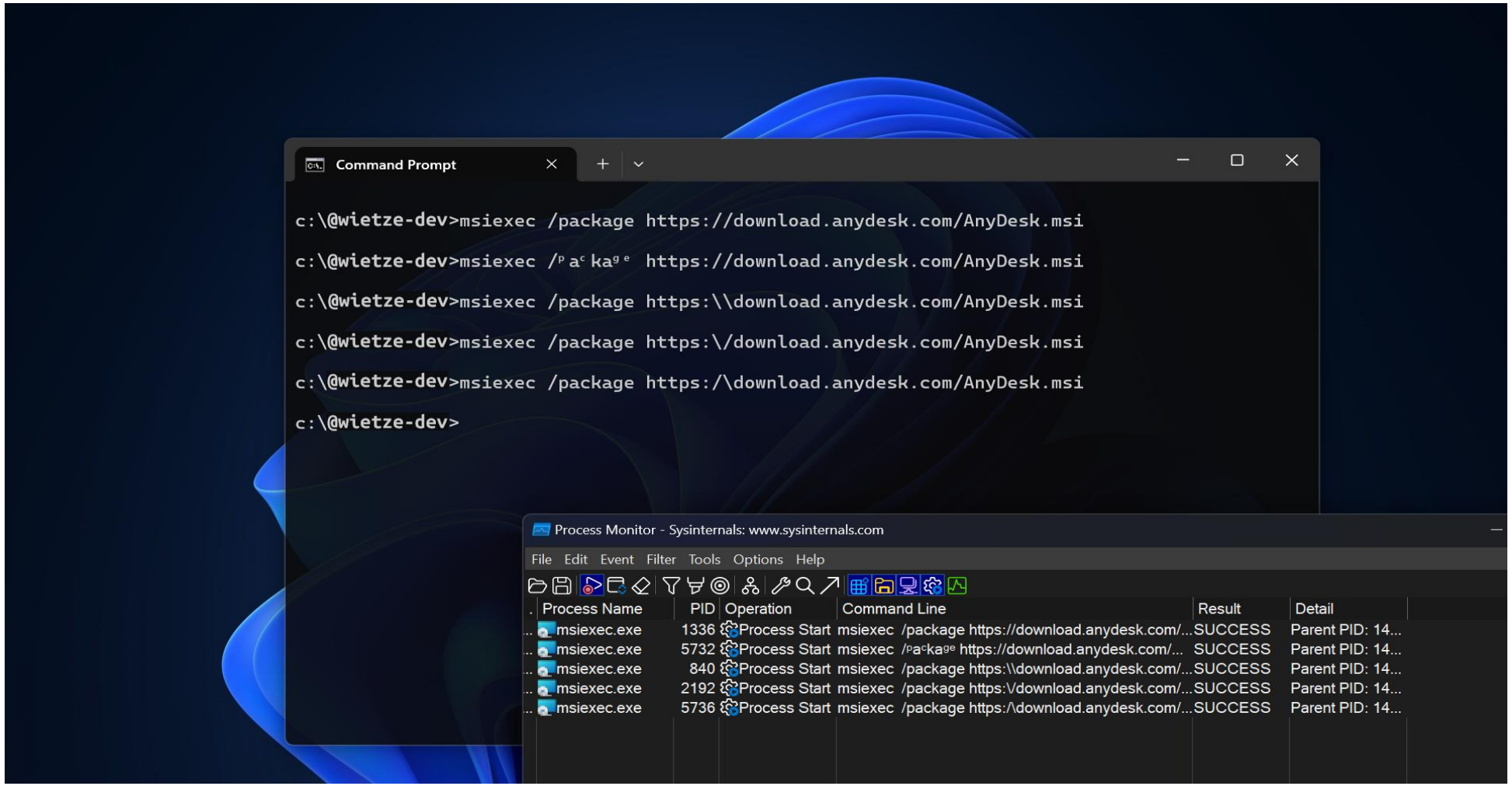
HACKDAY


elastic
Platform
Solutions
Customers
Resources
Pricing
Docs
Start free trial
Connection via systemd
Suspicious Network Tool Launched Inside A Container
Suspicious PDF Reader Child Process
Suspicious Passwd File Event Action
Rule query
edit
process where host.os.type == "windows" and event.action == "start
process.name : ("powershell.exe", "pwsh.exe") and
process.parent.name : ("wscript.exe", "cscript.exe", "mshta.exe") a
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 powershe /ec wByAGkAdABlAC0AaABvAHMA... SUCCESS Parent PID: 14...
powershell.exe 3992 Process Start powershe -ec wByAGkAdABlAC0AaABvAHMA... SUCCESS Parent PID: 14...
powershell.exe 7704 Process Start powershe -ec wByAGkAdABlAC0AaABvAHMA... SUCCESS Parent PID: 14...
powershell.exe 1208 Process Start powershe –ec dwByAGkAdABlAC0AaABvAHM... SUCCESS Parent PID: 14...
powershell.exe 2588 Process Start powershe ¯ec dwByAGkAdABlAC0AaABvAHM... 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 *",

Command-Line Obfuscation variants

HACKDAY

Character Substitution



HACKDAY

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.

Known False Positives

Associated Analytic Story

Risk Based Analytics (RBA)

References

Detection Testing

Search

```
SPL
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`
```

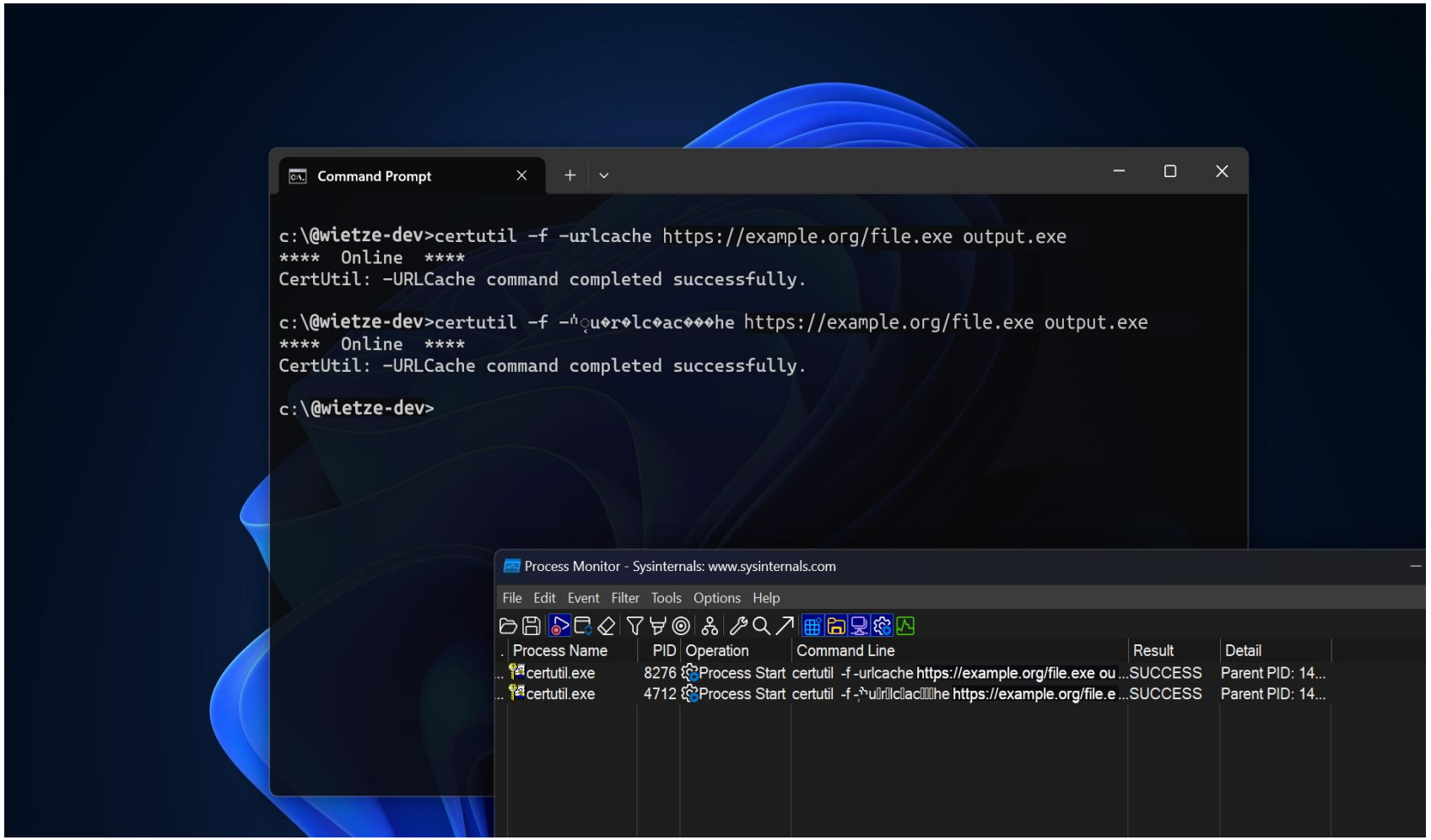
Data Source

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

HACKDAY

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 Filter Tools Options Help

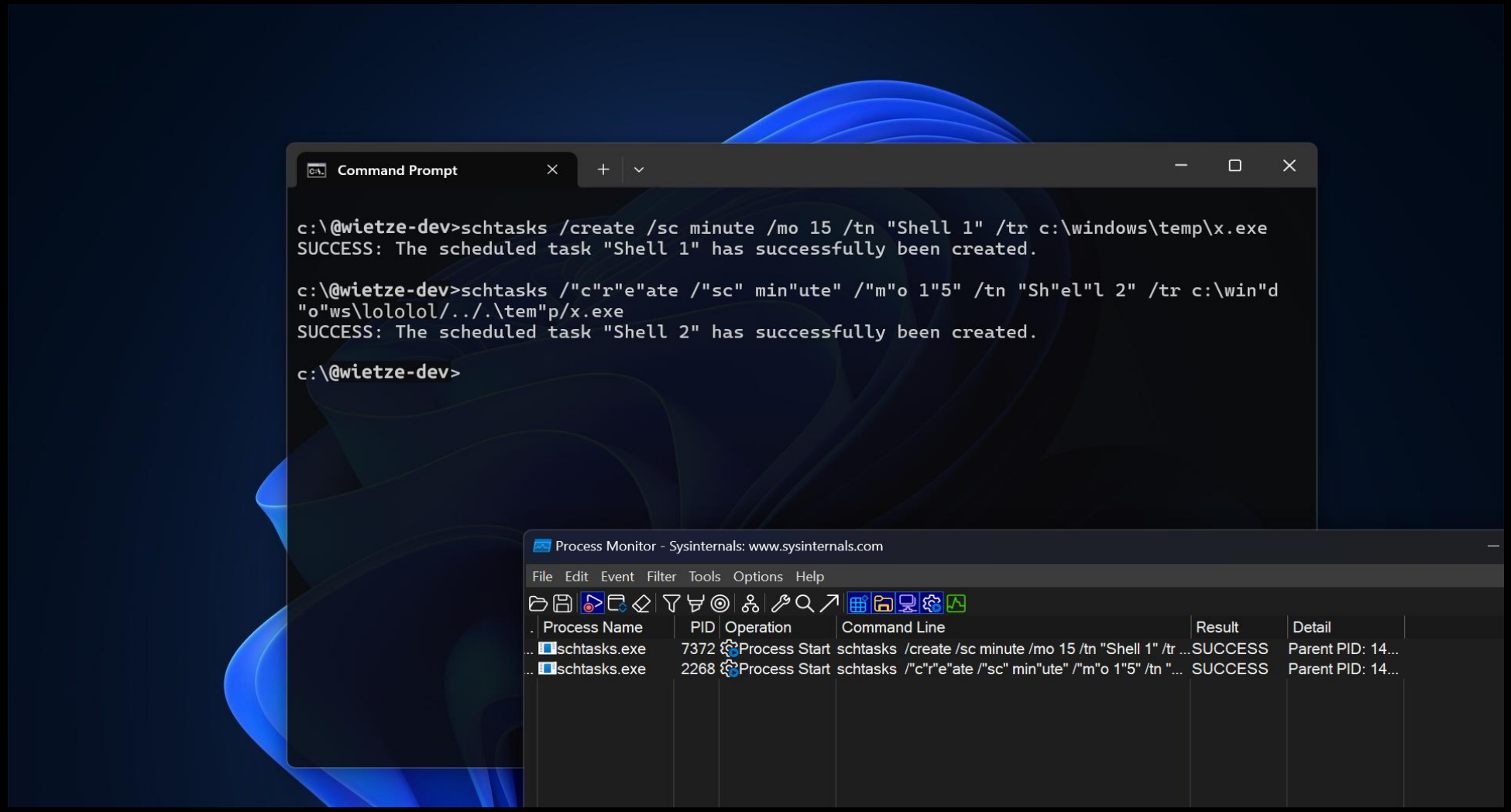
Process Name	PID	Operation	Command Line	Result	Detail
certutil.exe	8276	Process Start	certutil -f -urlcache https://example.org/file.exe ou ...	SUCCESS	Parent PID: 14...
certutil.exe	4712	Process Start	certutil -f -urlcache https://example.org/file.e ...	SUCCESS	Parent PID: 14...

```
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                - 'verifyctl '
30        selection_http:
31            CommandLine|contains: 'http'
32        condition: all of selection_*
33    falsepositives:
34        - Unknown
35    level: medium
```

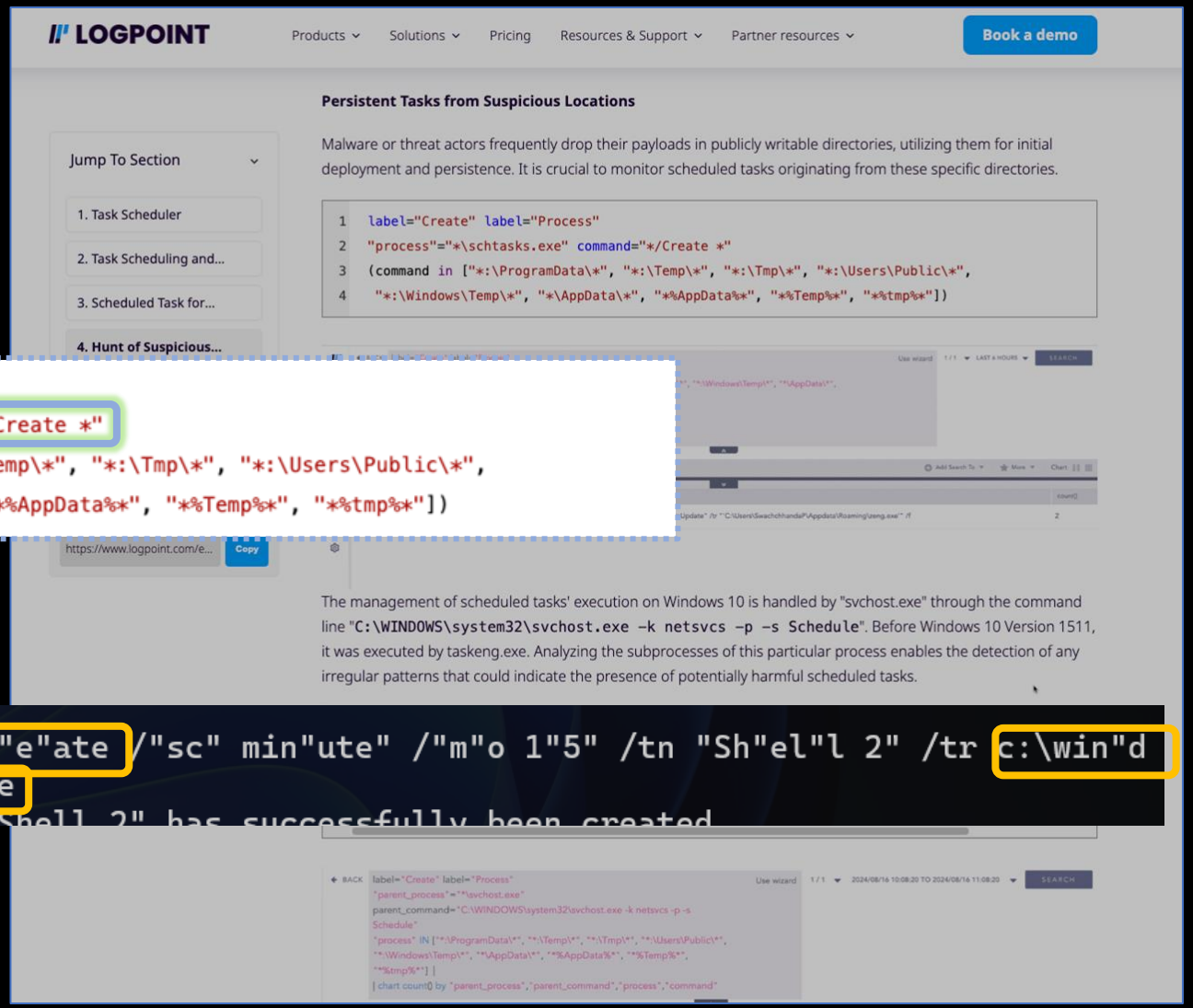
Command-Line Obfuscation variants

Keyword Obstruction

HACKDAY



HACKDAY

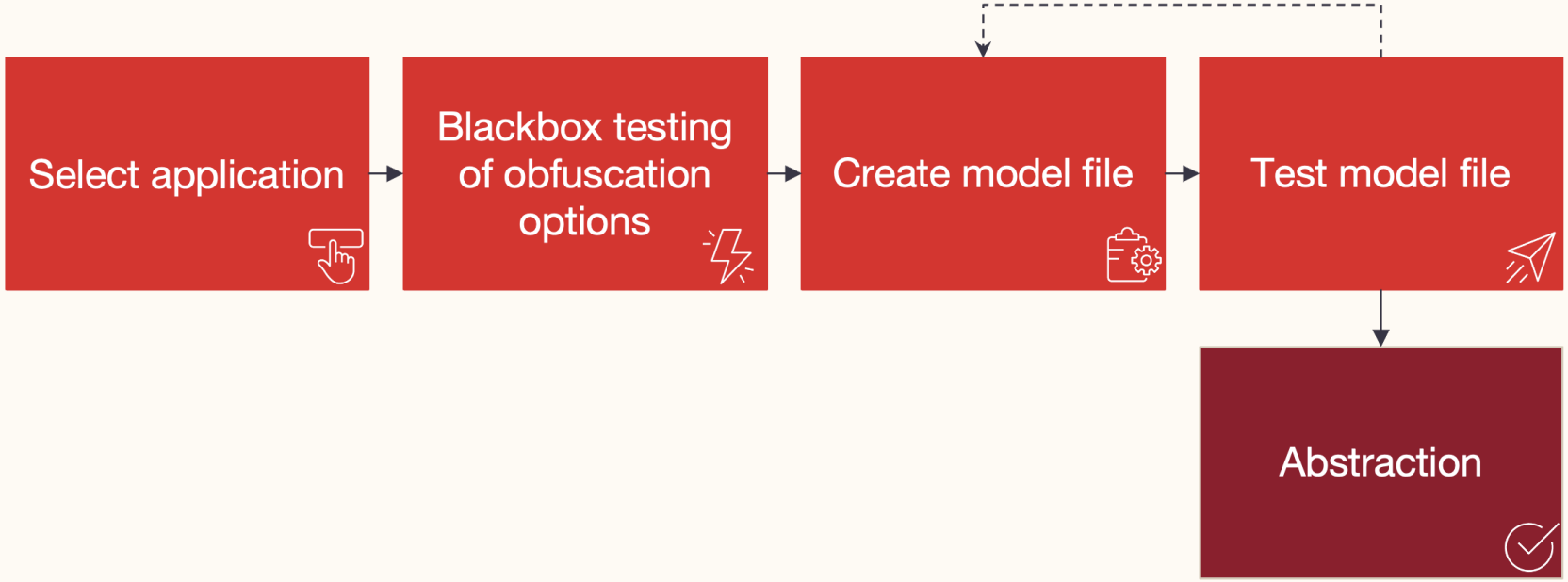



```
c:\@wietze-dev>schtasks /"c"r"e"ate /"sc" min"ute" /"m"o 1"5" /tn "Sh"el"l 2" /tr c:\win"d
"o"ws\lololol/../.\tem"p/x.exe
```

Cool, but…

it’s all anecdotal

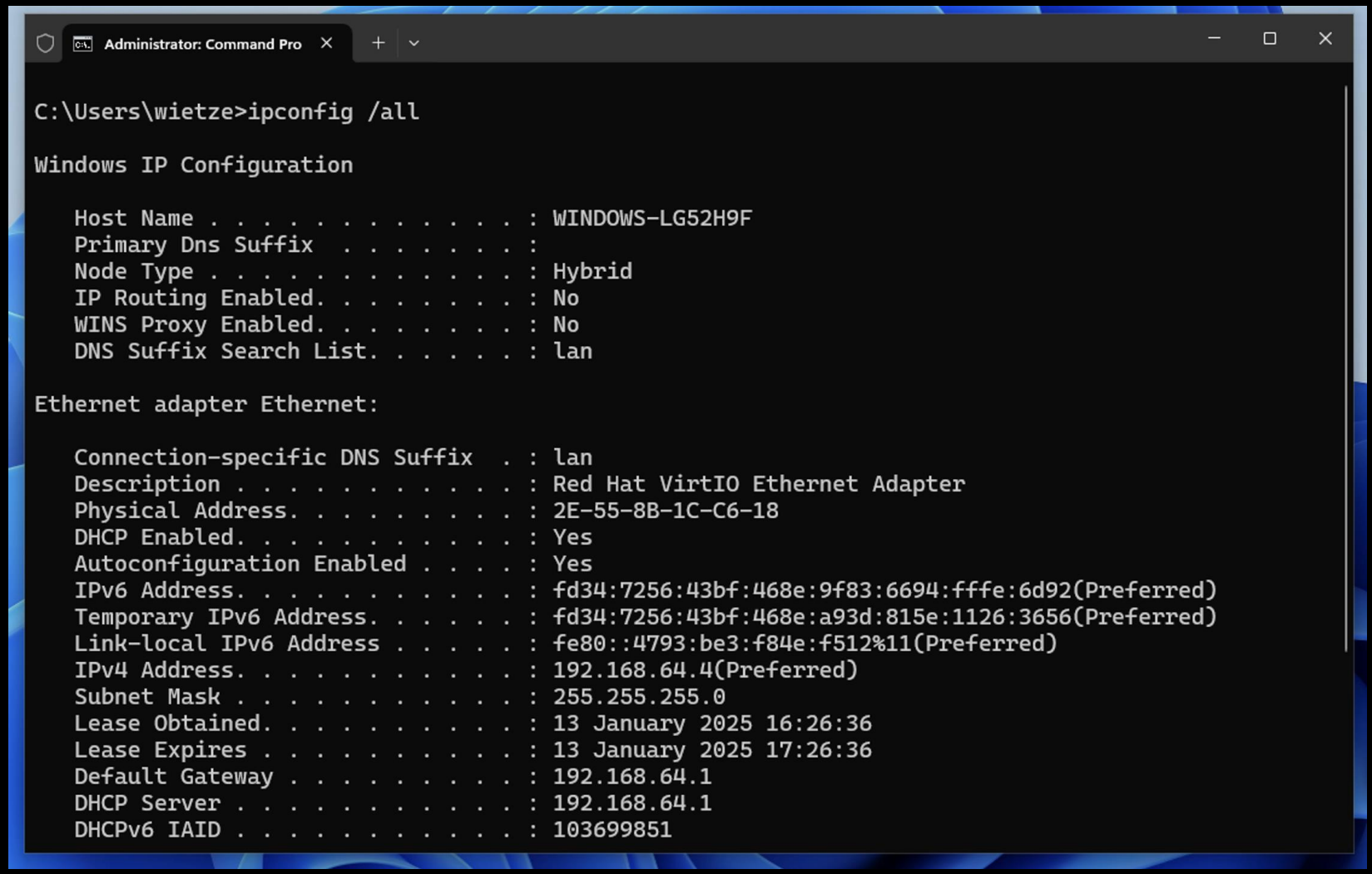
Process


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

Step 1:

Select an application

HACKDAY



Step 2:

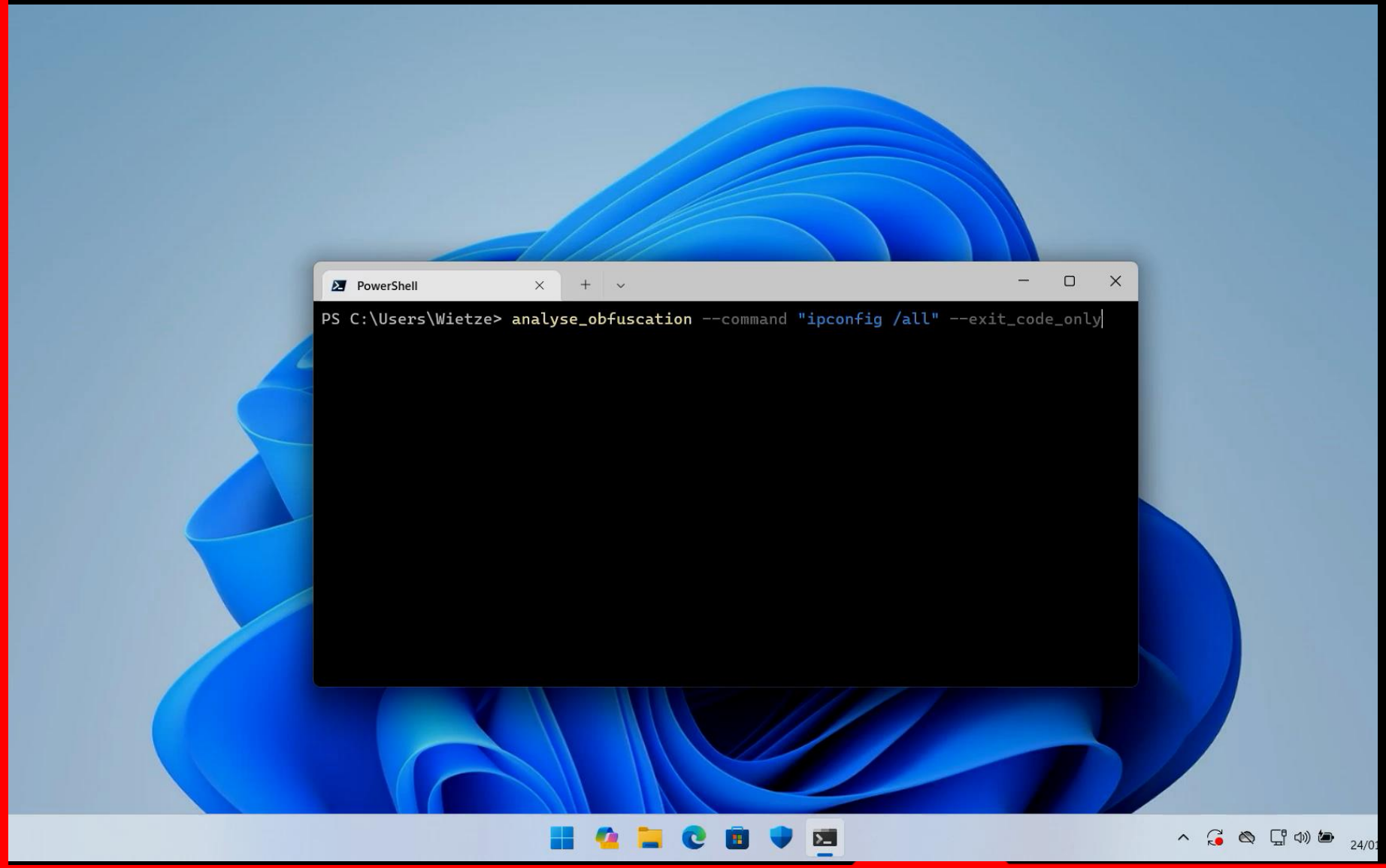
Blackbox test obfuscation options

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:

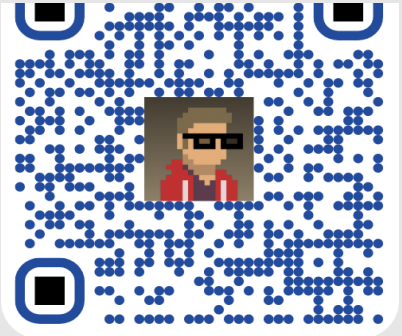
Blackbox test obfuscation options

HACKDAY



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

Introducing **Invoke-ArgFuscator**:

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

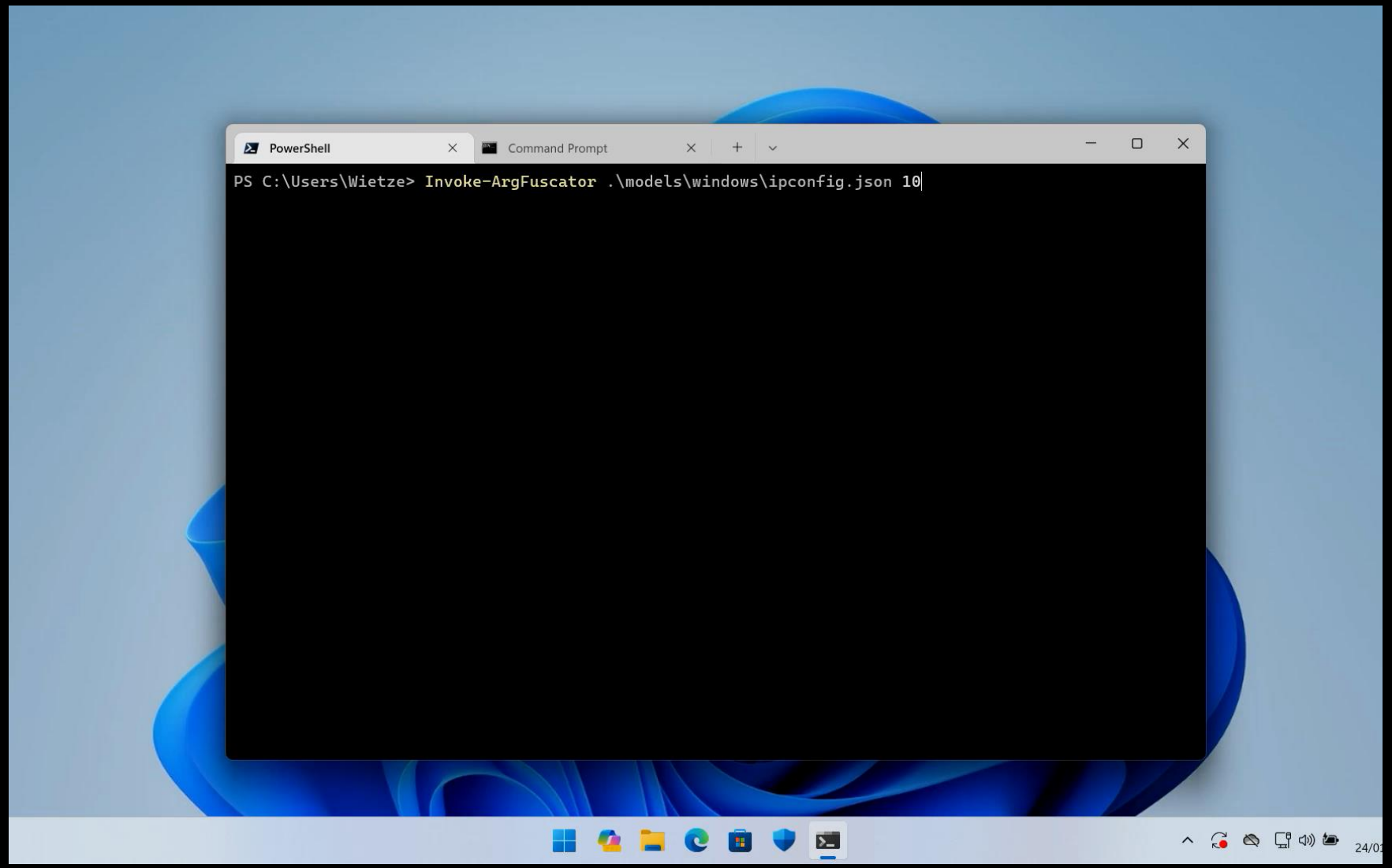
```
> Invoke-ArgFuscator "ipconfig.json" 3

< iPcoNfiG -Al⚽ ‾ᵇl
 >ipCoNFiG -ᵃL
< IpCoNFig /"Aˡ L"
```

But now…

Will It Run?

Step 4:
Test Model File

HACKDAY

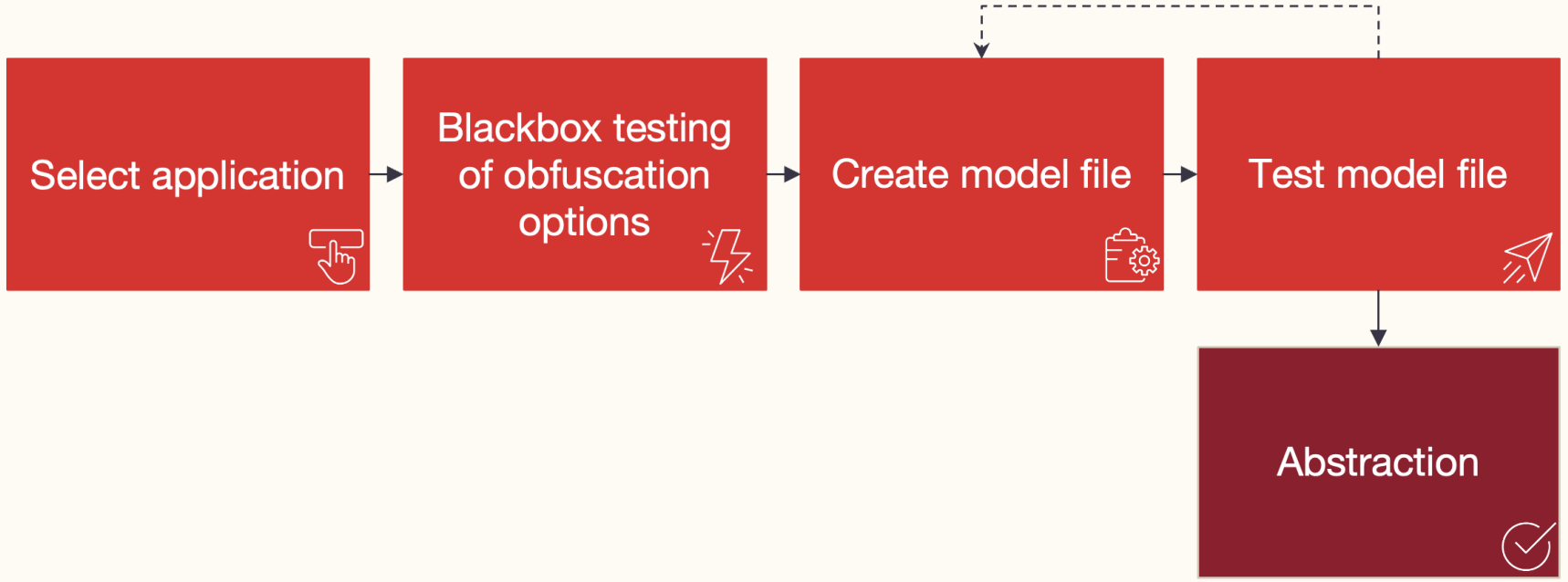


Invoke-ArgFuscator

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

HACKDAY

An open-source project

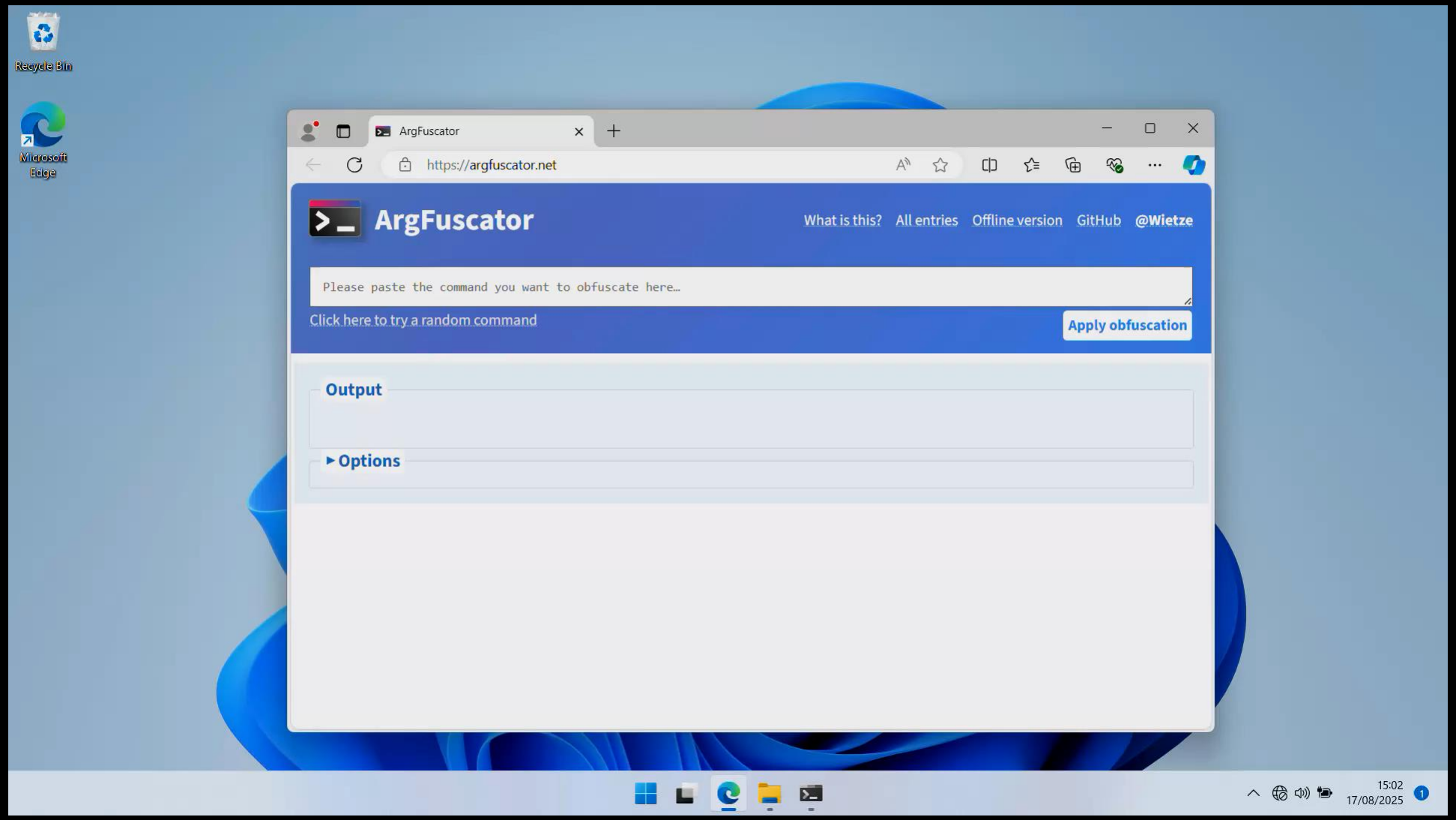
documenting and generating

command-line obfuscation opportunities

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

ArgFuscator.net
Platform

https://argfuscator.net


Recycle Bin
Microsoft Edge
ArgFuscator
https://argfuscator.net
ArgFuscator
What is this?
All entries
Offline version
GitHub
@Wietze
Please paste the command you want to obfuscate here…
Click here to try a random command
Apply obfuscation
Output
▸ Options
15:02
17/08/2025

Impact
Defenders

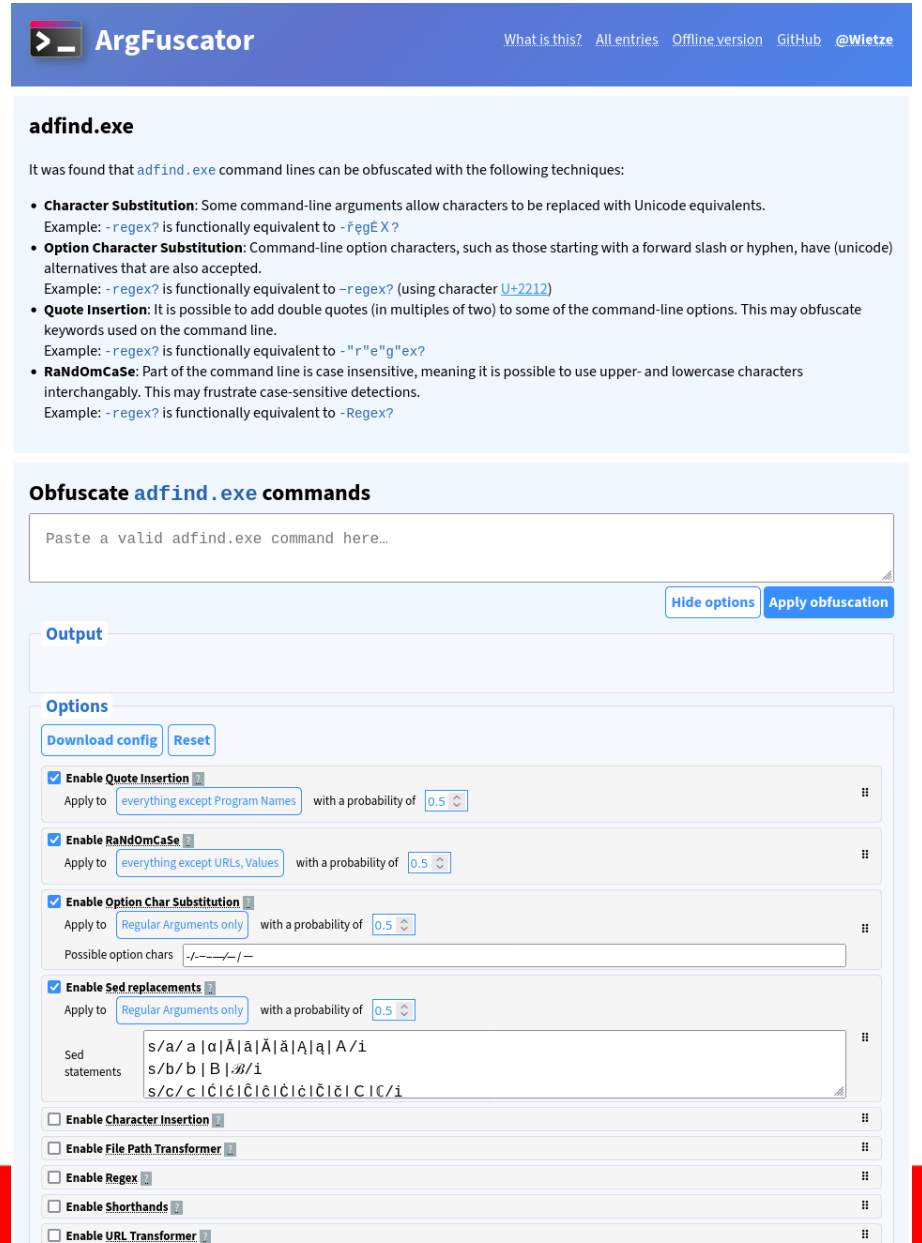
HACKDAY

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***



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

…

ArgFuscator

What is this? All entries Offline version GitHub @Wietze

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	✅		✅	✅				✅
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

HACKDAY

𝕏 @wietze
@wietzebeukema.nl
@wietze@infosec.exchange

/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