ODDVAR MOE'S BLOG

Notes from My adventures with Windows security

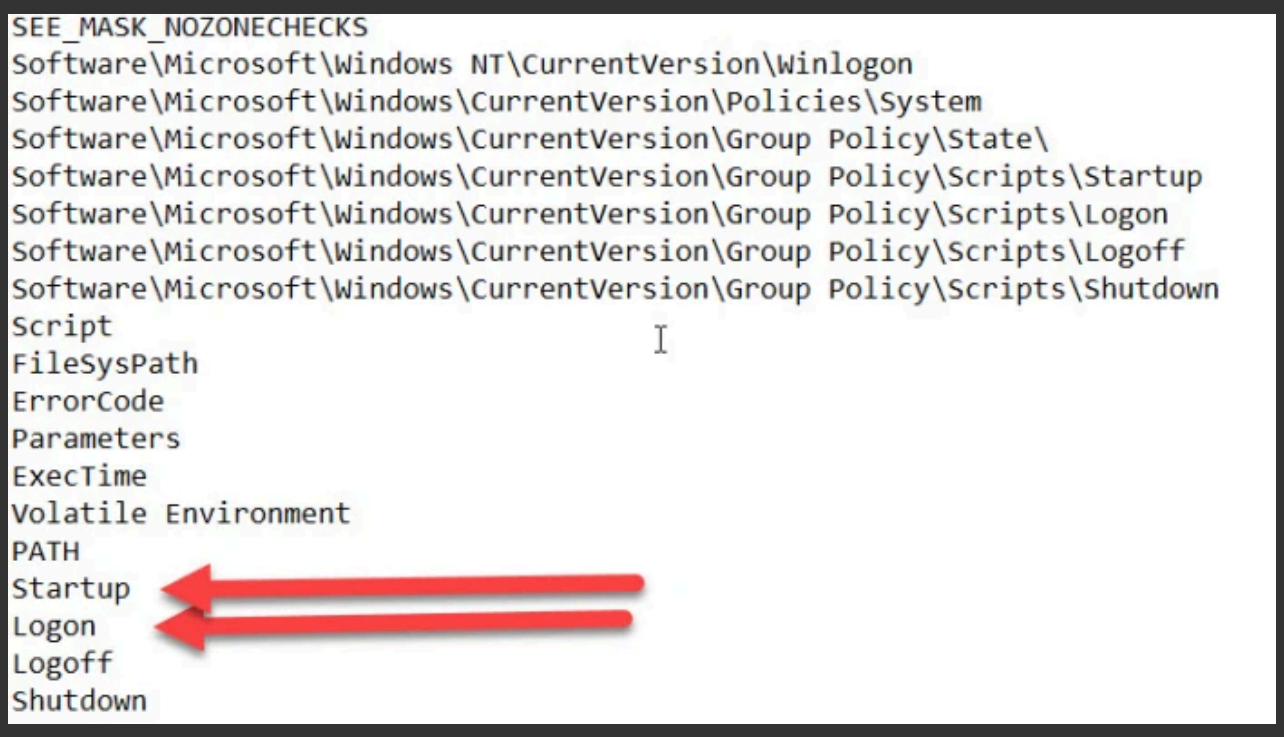
GPSCRIPT.EXE – ANOTHER LOLBIN TO THE LIST

Posted on 27 Apr 2018

TL;DR
– GPO scripts can be defined for user and started with GPScript.exe /Logon
– Logonscripts do not show up in Autoruns.exe

I started to play around with GPscript.exe here the other day and found some interesting stuff and I want to have this documented for the future, so therefor I wrote this blogpost for you to read.

I know from previous experiences that GPscript.exe is responsible for triggering logon scripts when you define them in Group Policy. After thinking a bit about this binary I did a strings on the file to see if could see anything interesting, and sure enough I did.



It turned out that you could supply the following parameters to the binary:

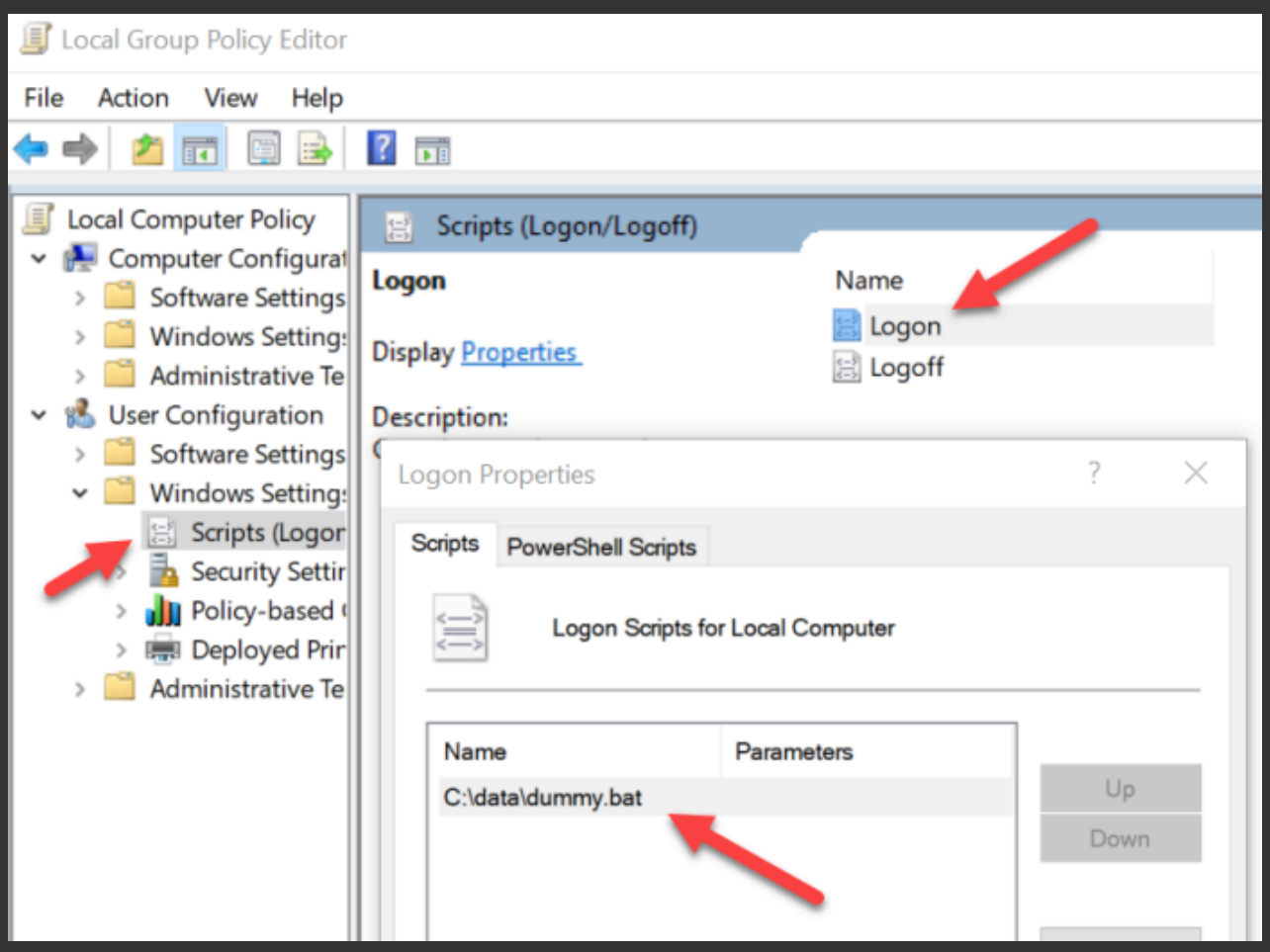
GPScript.exe /logon or GPscript.exe /Startup

If you have anything defined in the Group policy (Local group policy – gpedit.msc) under logon scripts it will execute if you supply /logon to the binary. That means you can execute the defined logon scripts at will with the command:

- GPScript.exe /logon

. When you add a script the group policy editor writes to the following registry key location:

- HKCU\Software\Microsoft\Windows\CurrentVersion\Group Policy\Scripts



In my example I defined a simple batch file as a logon script (C:\data\Dummy.bat).
When this is defined it adds these registry keys (exported in .reg format):

```
Windows Registry Editor Version 5.00
[HKEY_USERS\S-1-5-21-1848305745-3675528341-1622750934-1001\Software\Microsc

[HKEY_USERS\S-1-5-21-1848305745-3675528341-1622750934-1001\Software\Microsc
"GPO-ID"="LocalGPO"
"SOM-ID"="Local"
"FileSysPath"="C:\\Windows\\System32\\GroupPolicy\\User"
"DisplayName"="Lokal gruppepolicy"
"GPOName"="Lokal gruppepolicy"
"PSScriptOrder"=dword:00000001

[HKEY_USERS\S-1-5-21-1848305745-3675528341-1622750934-1001\Software\Microsc
"Script"="C:\\data\\dummy.bat"
"Parameters"=""
"IsPowershell"=dword:00000000
"ExecTime"=hex(b):00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
```

It also writes some info to the scripts.ini file
under C:\Windows\System32\GroupPolicy\User\Scripts and to the gpt.ini
under C:\Windows\System32\GroupPolicy
The content of that scripts.ini file looks like this:

```
[Logon]
0CmdLine=C:\data\dummy.bat
0Parameters=
```

The content of my GPT.ini file looks like this:

```
[General]
gPCMachineExtensionNames=[{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{62C1845D
Version=2020567
gPCUserExtensionNames=[{42B5FAAE-6536-11D2-AE5A-0000F87571E3}{40B66650-49
```

A lot of the different GUIDs can be found here.

Remember that the testing I did was against local group policy and not a defined Domain Group Policy. It is also important to understand that you need local administrator permissions to conduct these operations I am explaining in this blogpost.

After struggling a while with getting the registry keys and files in place for an attack I ended hitting my head against the wall. I reached out to the awesome Darren Mar-Elia aka grouppolicyguy on the Bloodhound slack. After some discussions he taught me something incredibly interesting. You don't need to add those registry keys at all. Thanks again Darren!

All you need to make this work is:

1. Add the Scripts.ini file in the correct place with the correct content (0CMDLine… See above)
2. Add the CSE guid (gPCUserExtensionNames=[{42B5FAAE-6536-11D2-AE5A-0000F87571E3}{40B66650-4972-11D1-A7CA-0000F87571E3}] to the GPT.ini
3. Increase the Version number of GPT.ini to something that is higher than what is currently in the file

4. Run gpupdate (populates all the needed reg keys)
5. Run Gpscript /logon and the script executes!!

Another cool thing that I discovered is that this technique **this does not show up in autoruns.**
You heard me right, user logon scripts do not show up in autoruns. (Mind blown)

I have not tested this if the computer is joined to a domain, but I am assuming that this persistence trick will work there as well, unless there is a domain gpo that overruns the setting.

If you want to go with computer startup scripts, you must know that it shows up in autoruns. You can define a startupscript for the computer as part of the group policy and get GPScript.exe to fire the script. The only "stupid" thing is that you need to get GPScript.exe started as system and supply the /Startup parameter. That means you already need to run GPScript.exe as system for this to work.

If you want to execute a Powershell script instead, you need to create a file called psscript.ini instead of script.ini and place it in the folder – C:\Windows\System32\GroupPolicy\User\Scripts

In my example I have a script called dummy.ps1.

```
[Logon]
0CmdLine=C:\data\dummy.ps1
0Parameters=
```

Detection (Blue team):

I would monitor for changes to or new Scripts.ini files.

 # "Responsible disclosure" #

I tried to reach out to Mark R. on Twitter about this a while back and I also wrote him an email. I have not gotten any response. I therefor decided to post this, since the technique is already known and publicly available in Hexacorns blog here:
http://www.hexacorn.com/blog/2017/01/07/beyond-good-ol-run-key-part-52/

It was not until after the intial discovery of the persistence technique that I figured out that it was already discovered by Adam – @hexacorn. He has written an excellent blogpost about this here .

 # Update #

Darren Mar-Elia has reached out and got contact with Mark Russinovich and this issue will be fixed. An update for Autoruns will likely be available within the next few days. 🙂

Loading...

RELATED

Another way to get to a system shell – Assistive Technology
23 Jul 2018
In "Security"

Persistence using Universal Windows Platform apps (APPX)
6 Sep 2018
In "Security"

Real whitelisting attempt using AppLocker
14 May 2018
In "Security"

PREVIOUS POST
Putting data in Alternate data streams and how to execute it – part 2

NEXT POST
Real whitelisting attempt using AppLocker

LEAVE A COMMENT