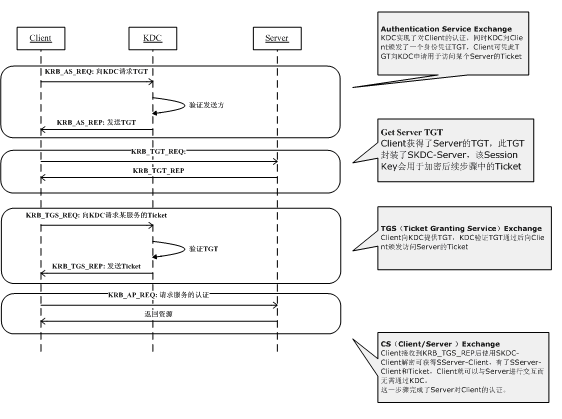
最近公司在做单点登录的选型，看了看Kerberos的技术实现。感觉网上Kerberos的相关资料不多，自己做了总结和大家分享，因为是在选型阶段，没做太深入的学习，学习了Kerberos的单点登录过程。   
Kerberos基本原理   
  
****先转几个链接****   
1.关于理解Kerberos原理的经典对话，MIT的人为了帮助人们理解Kerberos的原理而写的一篇对话集，google “Kerberos的原理”，满篇都是。   
2．谈谈基于Kerberos的Windows Network Authentication - Part I、II、III，挺好挺详细的文章，一共三篇，值得一看。   
[http://www.cnblogs.com/artech/archive/2007/07/05/807492.html](http://www.cnblogs.com/artech/archive/2007/07/05/807492.html" \t "F:/%E5%AF%86%E9%92%A5%E7%AE%A1%E7%90%86%E5%BC%80%E5%8F%91/_blank)   
3. 单点登录(SSO)的核心－－kerberos身份认证协议技术参考（一），也是一系列三篇文章   
[http://chnking.cnblogs.com/archive/2006/03/07/344506.html](http://chnking.cnblogs.com/archive/2006/03/07/344506.html" \t "F:/%E5%AF%86%E9%92%A5%E7%AE%A1%E7%90%86%E5%BC%80%E5%8F%91/_blank)   
  
*下面是自己的原创整理   
Kerberos登录过程*   
  
****名字解释****   
****KDC****：Kerberos Distribution Center 认证中心   
****TGT****：Ticket Granting Ticket 身份凭证   
****Master Key****：真正的密钥的Hash，Client、KDC、Server均有自己的Master Key   
****Session Key****：这种Key只在一段时间有效，用于加密双方传输的部分数据包   
****SKDC-Server****：Server和KDC间的Session Key   
****SKDC-Client****：Client和KDC间的Session Key   
  


****1.      Authentication Service Exchange**  **这一步骤KDC实现了对Client的认证，同时KDC向Client颁发了一个身份凭证TGT，Client可凭此TGT向KDC申请用于访问某个Server的Ticket。****   
1.1.     Client向KDC的Authentication Service服务发送KRB\_AS\_REQ，目的是申请获取一个用以访问所有Server的凭证（TGT）；   
KRB\_AS\_REQ包含：

* Pre-authentication data：一个被Client Master Key加密的Timestamp，用于KDC验证客户端身份；
* Client Info：Client标识，KDC以此查找Client的Master Key；
* Server Info：此处并不是Client要访问Server的Server Name，实际上是KDC的Ticket Granting Service

1.2.     AS（Authentication Service服务）根据Client Info从KDC中提取Client Master Key，解密Pre-authentication data，如果Timestamp合法，则客户端通过验证，因为这个Master Key仅Client和KDC知道；   
1.3.     AS将KRB\_AS\_REP发送给客户端   
KRB\_AS\_REP包含：

* 被Client Master Key加密过的SKDC-Client
* 被KDC Key加密过的TGT，TGT（SKDC-Client + Client Info + End Time），此TGT记做Client的TGT

****2.      KRB\_TGT**  **这一步骤Client获得了Server的TGT，此TGT封装了SKDC-Server，该Session Key会用于加密后续步骤中的Ticket****   
2.1.     Client接收到KRB\_AS\_REP后，使用Client Master Key对第一部分进行解密，可获得SKDC-Client，之后向Server申请获取Server的TGT，此TGT中封装了SKDC-Server；   
2.2.     Server将KRB\_AP\_REP返回给Client；   
KRB\_AP\_REP包括：

* 被KDC Key加密过的TGT，TGT（SKDC-Server + Server Info + End Time），此TGT记做Server的TGT

注：Server的TGT可如同步骤1由Server向KDC申请，并将申请到的TGT缓存到Server中。Client向Server获取时可直接将其返回，否则需通过AS Exchange由Server向KDC获取。   
****3.      TGS（Ticket Granting Service）Exchange**  **这一步骤Client向KDC提供TGT，KDC验证TGT通过后向Client颁发访问Server的Ticket****   
3.1.     Client获得Server TGT后，向KDC中的TGS（Ticket Granting Service）发送KRB\_TGS\_REQ，申请用于访问Server 的Ticket；   
KRB\_TGS\_REQ包含：

* Client的TGT：在步骤1处生成，由KDC Master Key加密
* Server的TGT：在步骤2处生成，由KDC Master Key加密
* Authenticator（Client Info +Timestamp）：用来证明Client TGT是自己所拥有的，所以需要用SKDC—Client加密
* Client Info：Client标识
* Server Info: Client试图访问的那个Server

3.2.     KDC先用KDC Master Key解密Client的TGT，获得SKDC-Client；   
3.3.     使用SKDC—Client解密Authenticator，通过比对3.2和3.3步骤中分别获得的Client Info可验证发送者是否是Client TGT的真正拥有者；   
3.4.     KDC再用KDC Master Key解密Server的TGT获得SKDC-Server；   
3.5.     验证通过后，向Client发送KRB\_TGS\_REP   
KRB\_TGS\_REP 包含：

* 使用SKDC-Client加密的SServer-Client
* 使用SKDC-Server加密的Session Ticket，该Ticket中包括SServer-Client、Client name、End Time（Ticket到期时间）

****4.      CS（Client/Server ）Exchange**  **Client接收到KRB\_TGS\_REP 后，使用SKDC-Client解密可获得SServer-Client，有了SServer-Client和Ticket，Client就可以与Server进行交互而无需通过KDC。**  **这一步骤完成了Server对Client的认证。****   
4.1.     Client接收到KRB\_TGS\_REP后，使用SKDC-Client对第一部分解密可获得SServer-Client，随后创建Authenticator并使用SServer-Client加密   
4.2.     Client向Server发送KRB\_AP\_REQ   
KRB\_AP\_REQ包含内容：

* Session Ticket（Client Info + SServer-Client+ End Time）
* Authenticator （Client Info + Timestamp）
* 是否需要进行双向验证的Flag

4.3.     Server使用SKDC-Server解密Ticket获得 SServer-Client；   
4.4.     使用SServer-Client解密Authenticator，通过比较Authenticator中的Client Info和Ticket中的Client Info实现对Client的验证。