Kerberos

1. Kerberos 协议

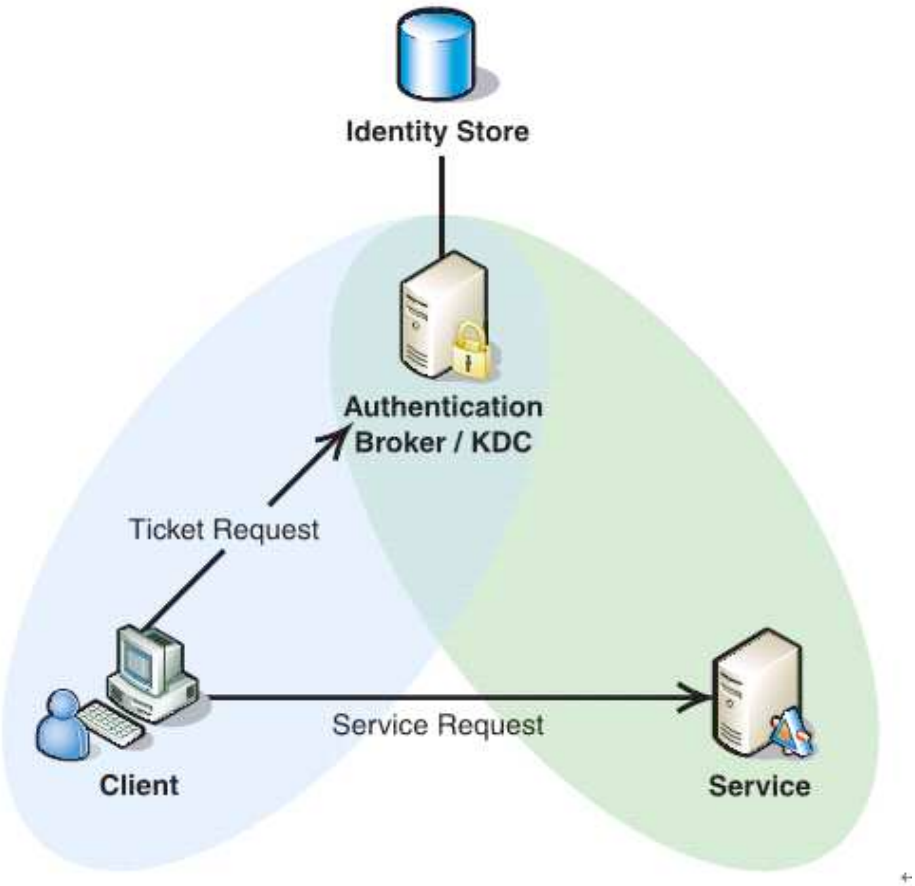
Kerberos 协议主要用于计算机网络的身份鉴别(Authentication), 其特点是用户只需输入一次身份验证信息就可以凭借此验证获得的票据(ticket-granting ticket)访问多个服务，即 SSO(Single Sign On)。由于在每个 Client 和 Service 之间建立了共享密钥，使得该协议具有相当的安全性。

2. 角色

Kerbores 中有三种角色：
KDC：负责分发密钥的密钥分配中心
Client：需要使用 kerbores 服务的客户端
Service：提供具体服务的服务端

3.条件

如下图所示，Client 与 KDC， KDC 与 Service 在协议工作前已经有了各自的共享密钥，并且由于协议中的消息无法穿透防火墙，这些条件就限制了 Kerberos 协议往往用于一个组织的内部， 使其应用场景不同于 X.509 PKI。

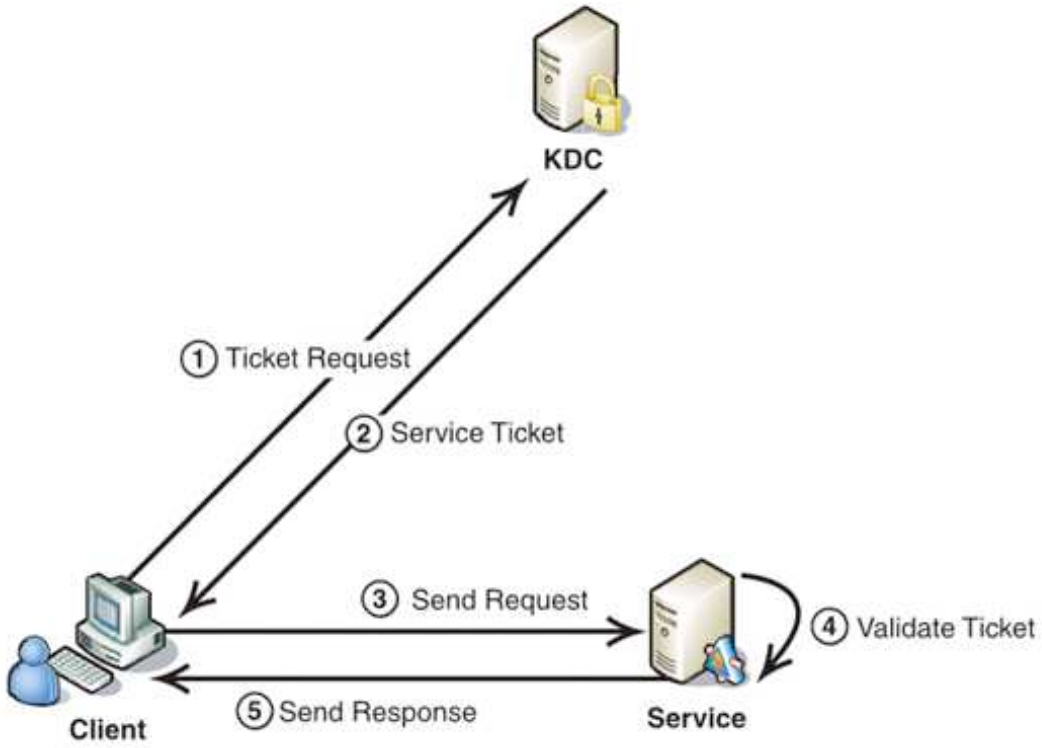


4.过程

Kerberos 协议获取原始票据过程：

1) Client 向 KDC 发送自己的身份信息；
2) KDC 从 Ticket Granting Service 得到 TGT(ticket-granting ticket)；
3) KDC 用 Client 与 KDC 之间的密钥将 TGT 加密；
4) KDC 将加密后的 TGT 发送给 Client；
5) Client 利用 Client 与 KDC 之间的密钥将加密的 TGT 解密，从而获得真正的 TGT；

Kerberos 协议获取服务票据过程：



1) Client 将之前获得 TGT 和要请求的服务信息(服务名等)发送给 KDC；
2) KDC 中的 Ticket Granting Service 将为 Client 和 Service 之间生成一个 Session Key 用于 Service 对 Client 的身份鉴别。
3) 然后 KDC 将这个 Session Key 和用户名，用户地址（IP），服务名，有效期, 时间戳一起包装成一个 Service Ticket，用于 Service 对 Client 的身份鉴别；
4) KDC 用 KDC 与 Service 之间的密钥将 Service Ticket 加密，这个 Ticket 是要给 Service 的，不能让 Client 看到；并且 KDC 用 Client 与它之间的密钥将 Session Key 加密；
5) KDC 将加密的 Service Ticket 和加密的 Session Key 发送给 Client ；
6) Client 解密 Session Key，然后将自己的用户名，用户地址（IP）打包 Session Key 加密也发送给 Service；Client 将收到的加密的 Service Ticket 转发到 Service；
7) Service 收到 Ticket 后，用它与 KDC 之间的密钥将 Service Ticket 解密，从而获得 Session Key 和用户名，用户地址（IP），服务名，有效期。再用 Session Key 将 Authenticator 解密从而获得用户名，用户地址 (IP) 将其与之前 Ticket 中解密出来的用户名，用户地址（IP）做比较从而验证 Client 的身份。
8) 如果 Service 有返回结果，将其返回给 Client。

5.keytab

keytab（密钥表）是“key table（密钥表）”的缩写，提供服务的每台主机都必须包含称为 keytab（密钥表）的本地文件。密钥表包含相应服务的主体，称为服务密钥。服务使用服务密钥向 KDC 进行自我验证，并且只有 Kerberos 和服务本身知道服务密钥。例如，如果您有基于 Kerberos 的 NFS 服务器，则该服务器必须具有包含其 nfs 服务主体的密钥表文件。

要将服务密钥添加至密钥表文件，应使用 kadmin 的 ktadd 命令，将相应的服务主体添加至主机的密钥表文件。由于要将服务主体添加至密钥表文件，因此该主体必须已存在于 Kerberos 数据库中，以便 kadmin 可验证其存在。

6.总结

概括起来说 Kerberos 协议主要做了两件事

1) Ticket 的安全传递。
2) Session Key 的安全发布。

再加上时间戳的使用就很大程度上的保证了用户鉴别的安全性。并且利用 Session Key，在通过鉴别之后 Client 和 Service 之间传递的消息也可以获得 Confidentiality(机密性), Integrity(完整性)的保证。不过由于没有使用非对称密钥自然也就无法具有抗否认性，这也限制了它的应用。不过相对而言它比 X.509 PKI 的身份鉴别方式实施起来要简单多了。

KDC 为什么不直接将包发送给 Client 和 Server？

- 由于一个 Server 会面对若干不同的 Client，而每个 Client 都具有一个不同的 Session Key。那么 Server 就会为所有的 Client 维护这样一个 Session Key 的列表，这样做对于 Server 来说是比较麻烦而低效的。
- 由于网络传输的不确定性，可能出现这样一种情况：Client 很快获得 Session Key，并将这个 Session Key 作为 Credential 随同访问请求发送到 Server，但是用于 Server 的 Session Key 确还没有收到，并且很有可能承载这个 Session Key 的永远也到不了 Server 端，Client 将永远得不到认证。