本章节主要讲了windows验证的几种类型

1-本地验证

2-NTLM网络验证

3-域验证Kerberos

4-跨域kerberos验证

5-协商机制

6-kerberos delegation

**Part1-Window本地验证**

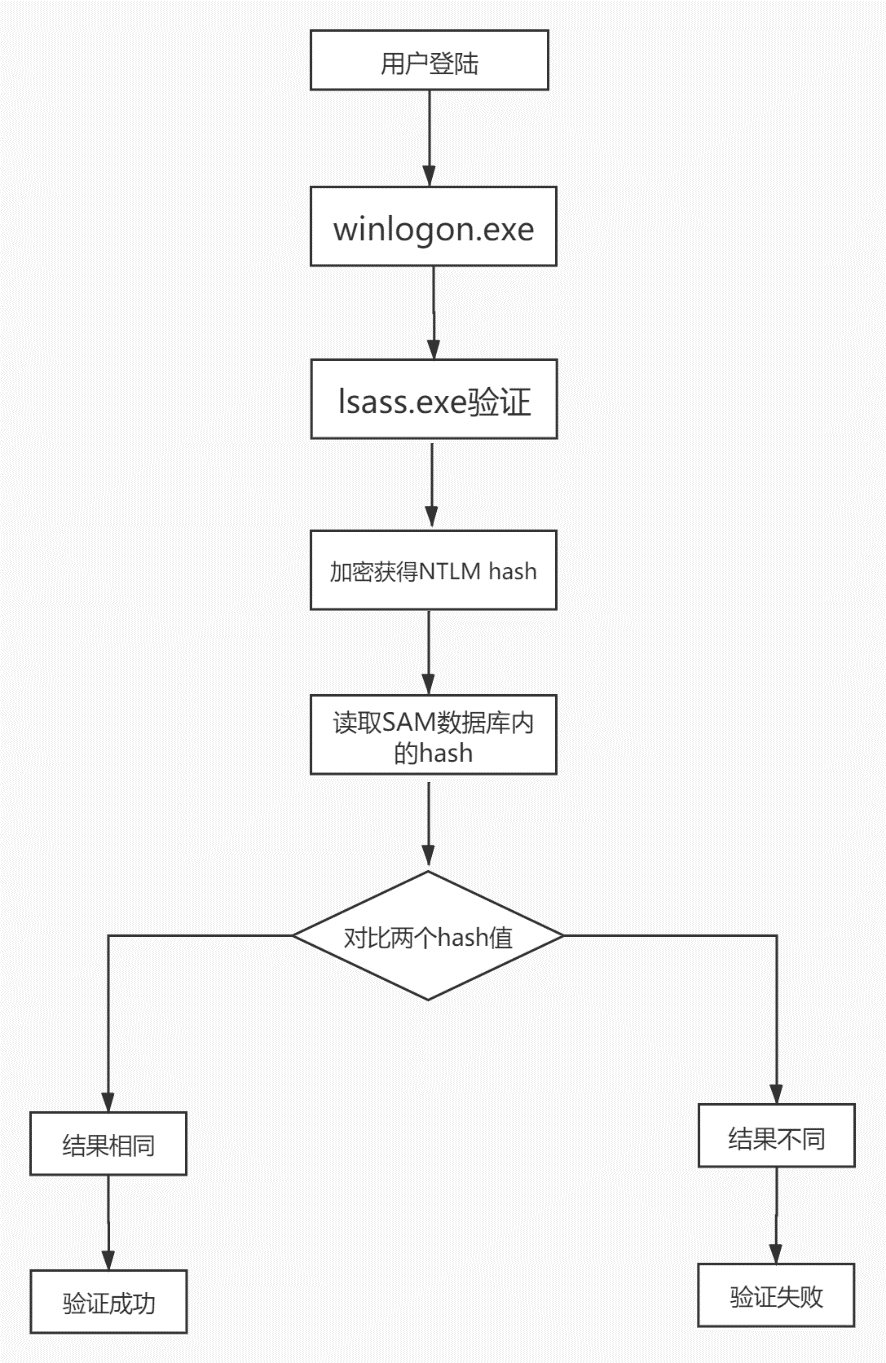
用户登陆计算机需要验证密码，密码储存在windows/system32/config/SAM数据库

登陆采用NT LAN manager HASH来验证，这个hash是一个32位字符串

登陆时用户输入密码，转换成NTLM hash 与SAM数据库文件的加密密码对比

算法：把用户密码转换成16进制编码，再通过UNICODE编码和MD4摘要算法获得32位hash。

以下是验证流程图解



**Part2- NTLM网络验证**

1. LM Hash验证

特点：a)将小写字符转换成大写字符

b)密码转换成16进制，分两组，填充为14字符，空余位用0填补

加密过程：使用DES加密，用KGS!@#￥%作为加密key

由于这个Key是固定的，所以容易导致明文简单，和通过LMhash就可以判断出密码的位数

所以是有缺陷不安全的

1. 在LM Hash的基础上产生了NTLM hash，分为V1和V2两个版本

NTLM验证由SMB服务发展而来，端口是445，SMB协议支持共享文件资源

NTLM V1和V2

采用challenge/response机制

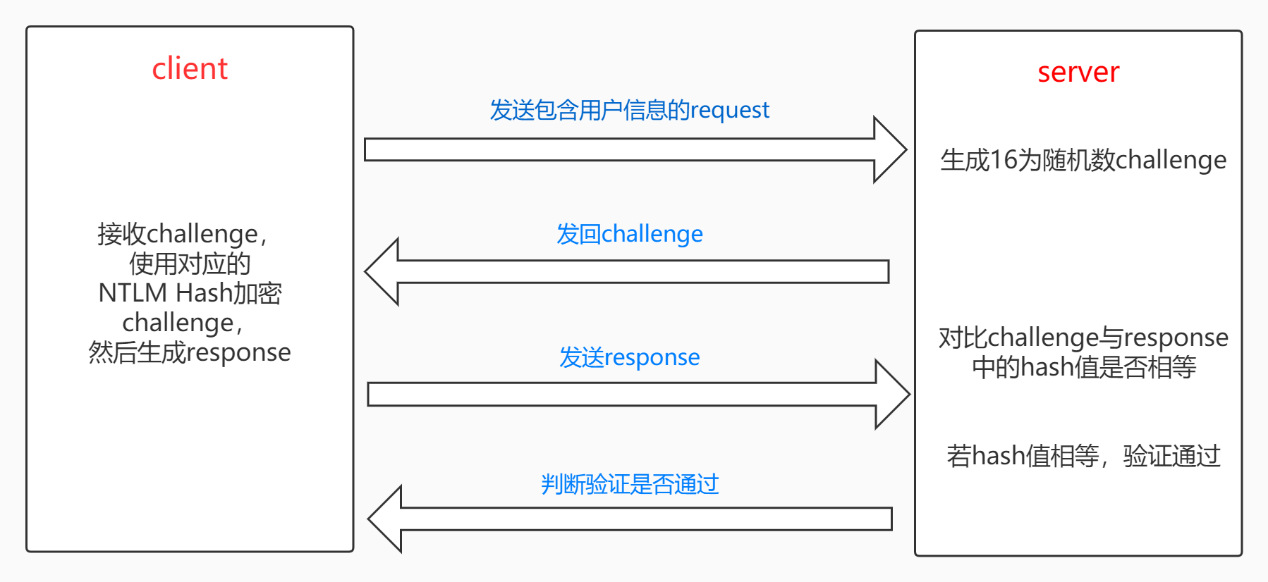
主要有以下三个步骤

协商——质询——验证

协商的主要目的是判断决定使用V1版本协议还是V2版本协议，不同的协议功能和兼容性不同

质询和验证的主要步骤是challenge/response的工作机制

如下图



NTLM V1和V2

区别：challenge和加密算法不同

共同点：加密都使用NTLM Hash

|  |  |  |
| --- | --- | --- |
| 区别 | V1 | V2 |
| Challenge | 8位 | 16位 |
| NTLM hash算法 | DES | HMAC-MD5 |

Windows默认使用 NTLM V2协议验证

1. 关于域环境下的HASH传递

Hash传递允许用户不需输入密码，直接将NTLM response发送给server

实现hash传递需要的条件

* 具有一个server
* 具有一个被传递的用户名
* 具有认证用户的NTLM hash

哈希传递就是使用用户名对应的NTLM Hash将服务器给出的challenge加密，生成一个response

**Part3- 域Kerberos验证**

域Kerberos认证特点：

* 提供密钥体系
* 具有信托机构
* 避免中间人攻击
* 不依赖于主机系统认证
* 通过传统密码技术来执行认证服务

参与角色：client——server——KDC

KDC角色和服务

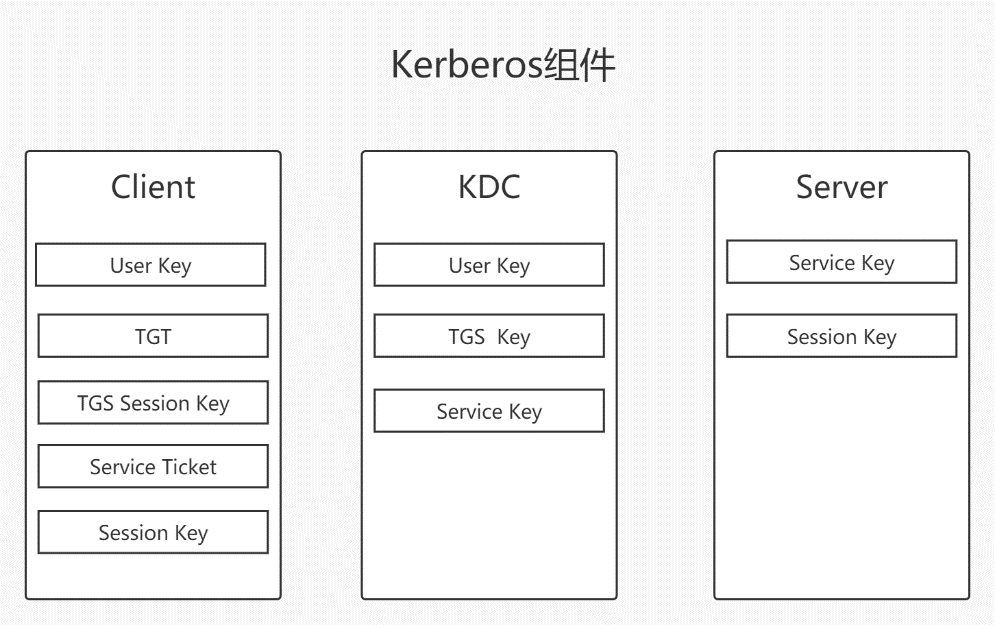
AD DATABASE：用户数据储存的位置

AS：用于生成TGT

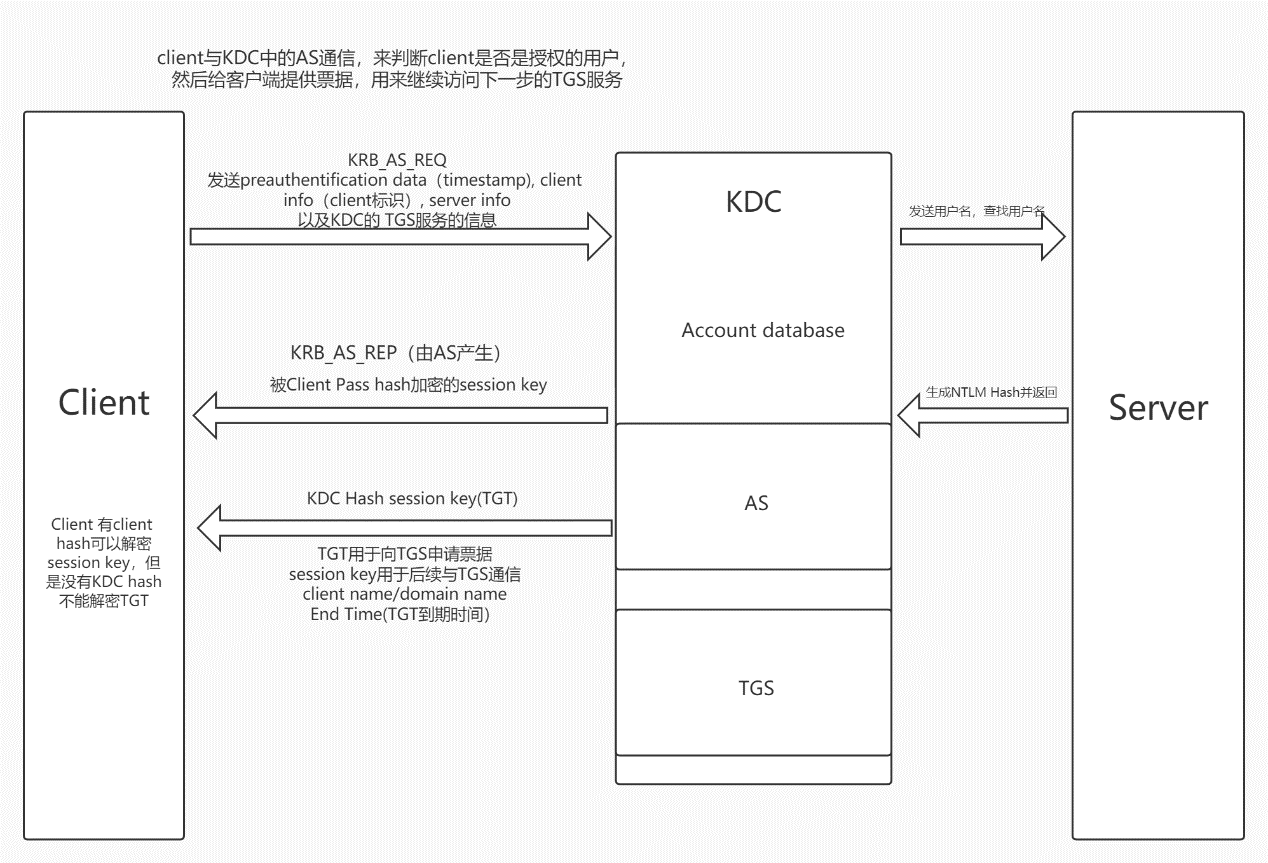
TGT：用于验证的票据

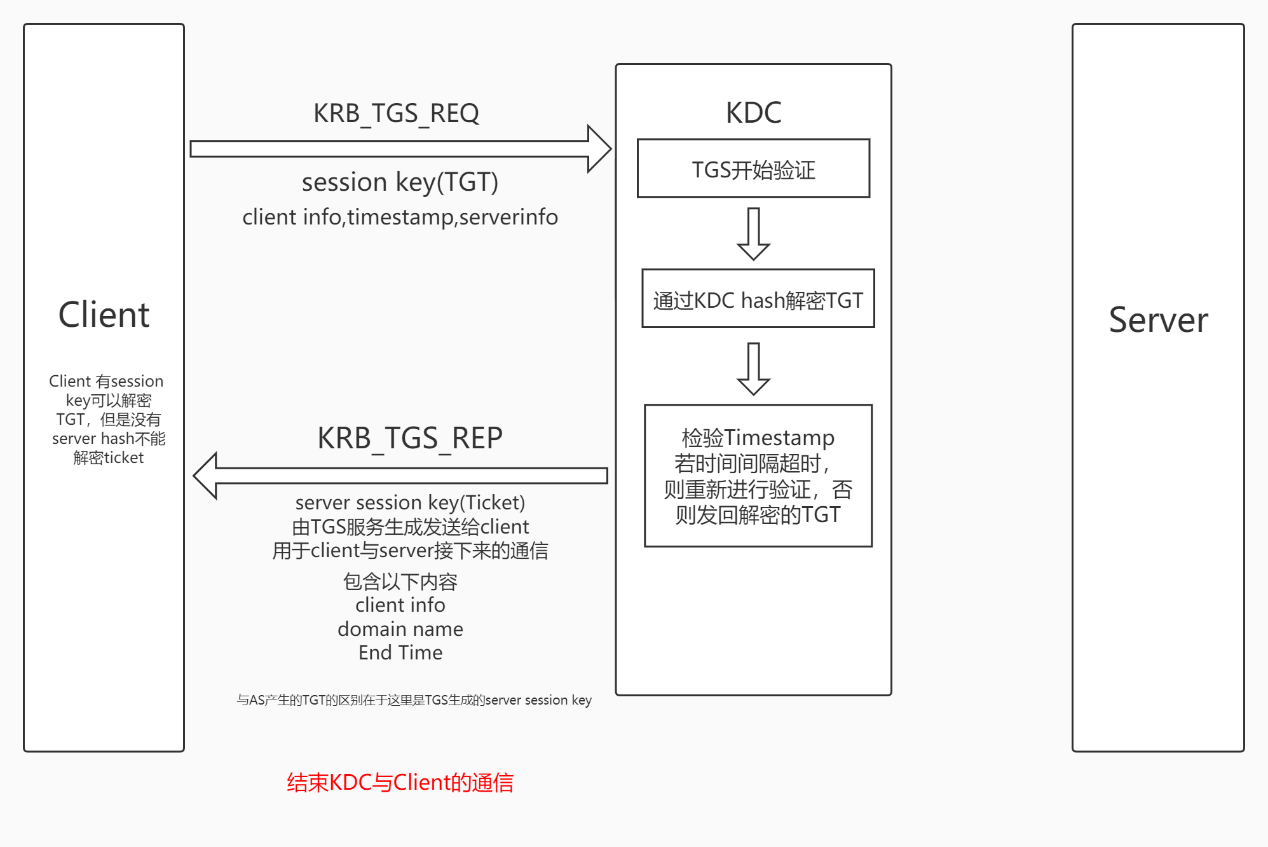
TGS：为客户端生成服务的票据ticket

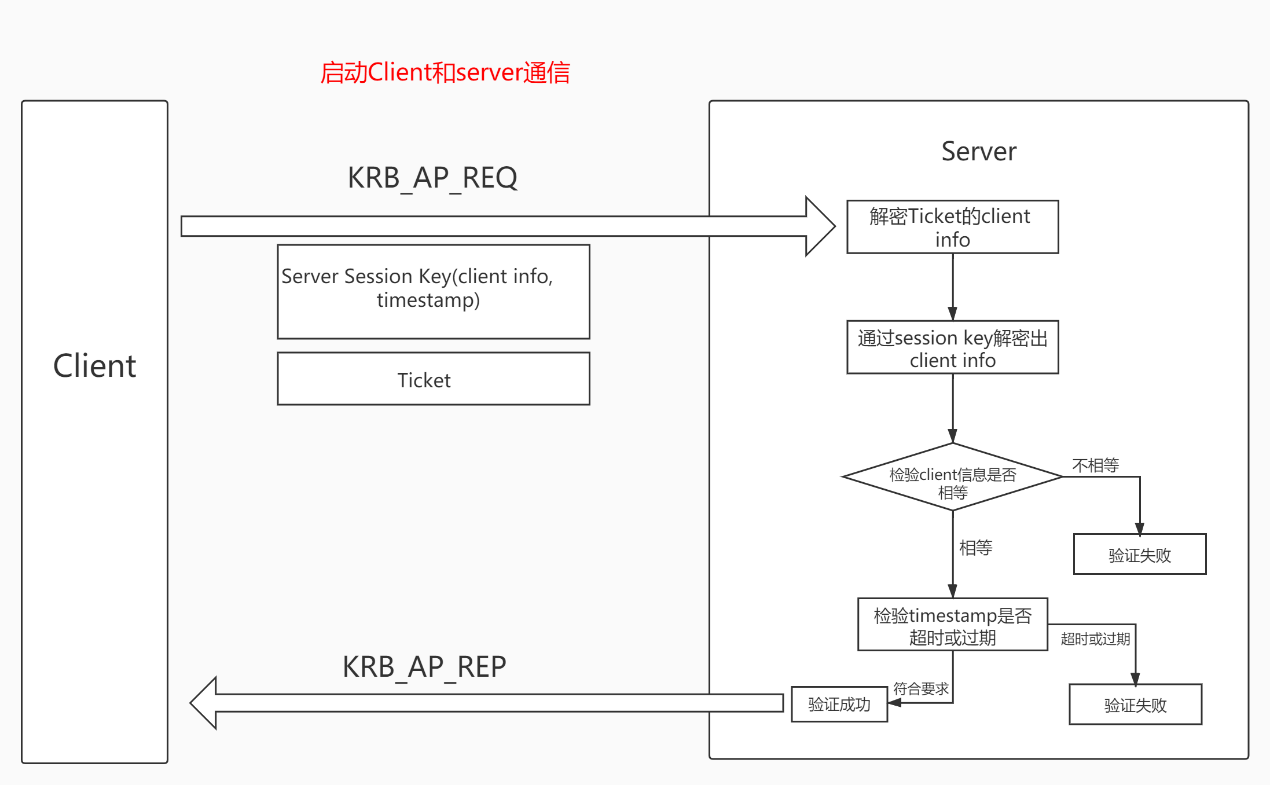
Kerberos 组件



Kerberos验证流程







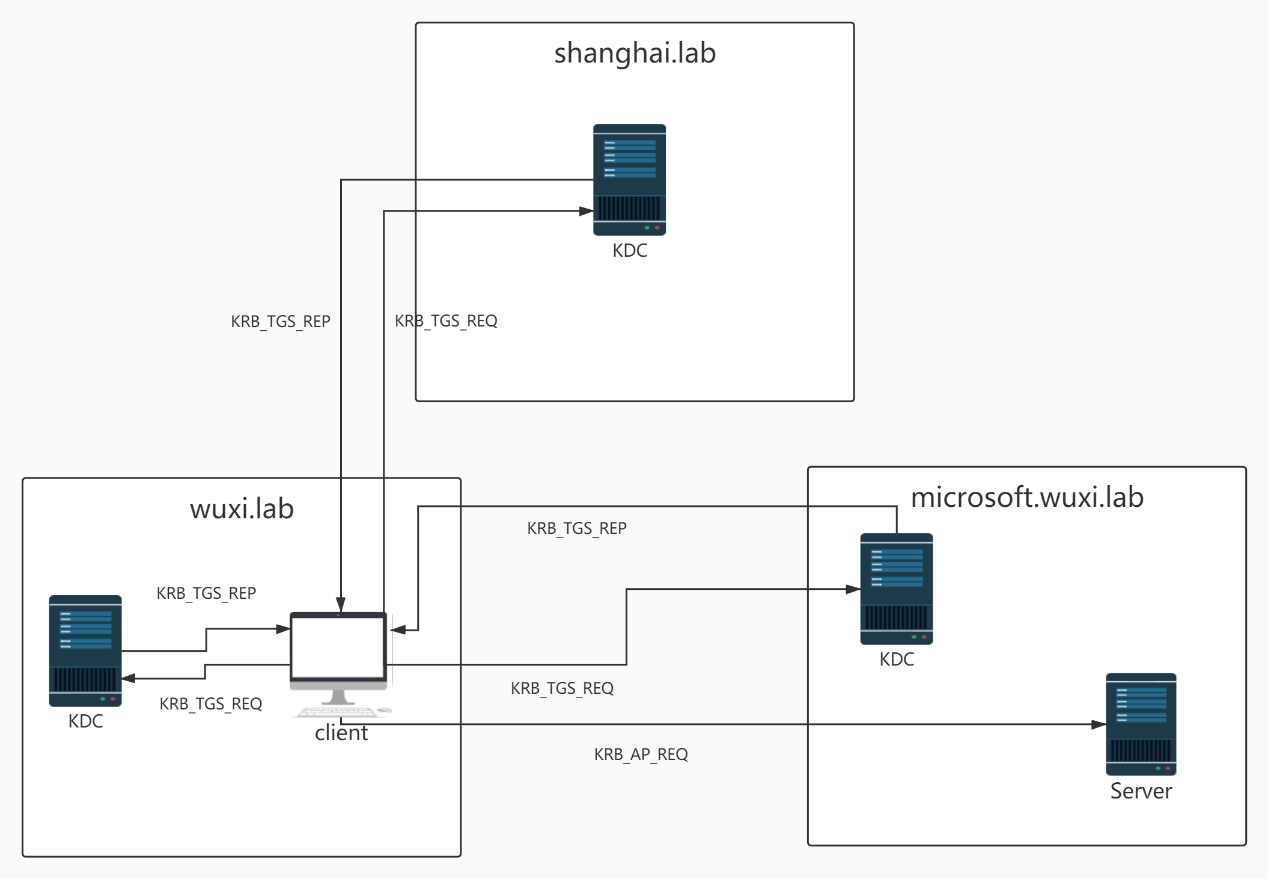
**Part4- 跨域Kerberos验证**

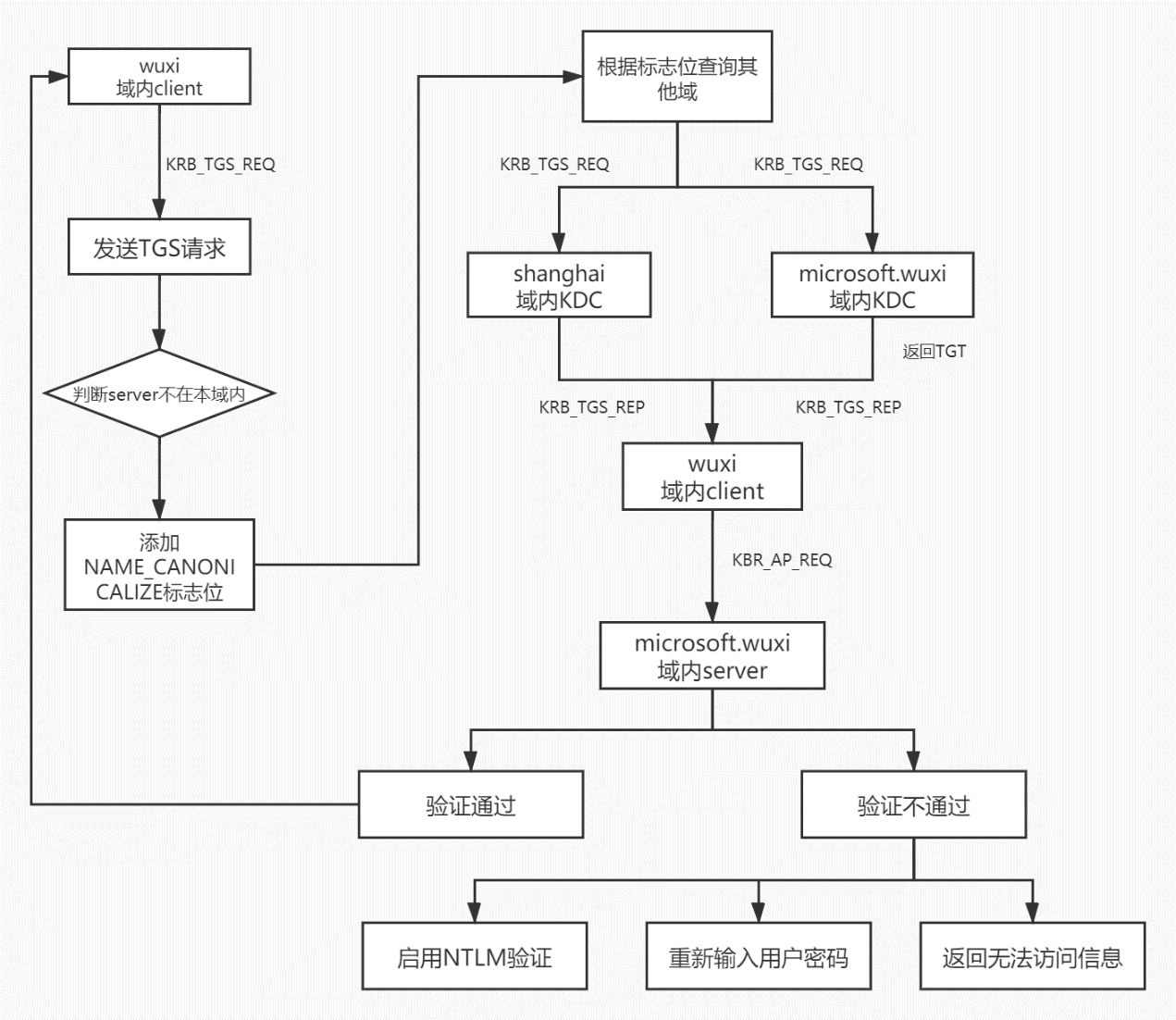
跨域验证流程的基本过程还是kerberos验证，大致流程差不多，以下以我的实验环境举例并图解说明

首先我有三个域：shanghai.lab, wuxi.lab, microsoft.lab

每个域都有KDC，假设wuxi.lab 内的client要访问microsoft.lab内的server，

流程图解如下





**Part5- Negotiation协商机制**

Microsoft negotiation是一个安全支持提供者(SSP)，它充当安全支持提供者接口(SSPI)和其他SSP之间的应用层。

Microsoft negotiation用来分析请求，然后根据配置的安全策略选择最佳的SSP（kerberos或者NTLM）

协商机制

为了允许协商选择Kerberos SSP，客户机应用程序必须提供服务主体名称(SPN)、用户主体名称(UPN)或NetBIOS帐户名称作为目标名称。否则，始终选择NTLM SSP。

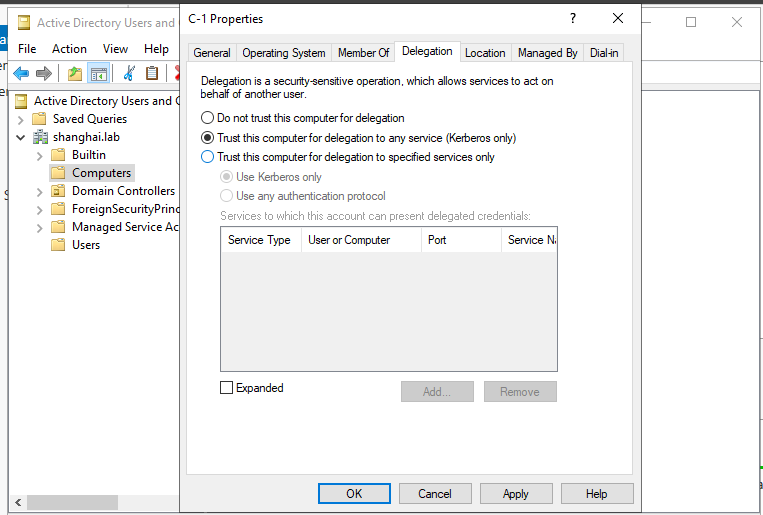
作用

* 允许系统使用最强（最安全）的可用协议。
* 确保您的应用程序的向前兼容性。
* 确保您的应用程序表现出符合客户设置的安全策略的行为。

**Part6- Kerberos委派机制**

Kerberos委派的实际用法是允许用户通过一个被委派的账户，实现对某个不开放给所有人的服务进行访问的权限。

对于计算机账户的委派设置



委派的种类

TRUSTED\_FOR\_DELEGATION 标志

1-不受限制的委托

当对象通过配置了无限制委托的主机进行身份验证时，该帐户的票证授予票证（TGT）将存储在内存中。这样一来，配置了无限制委派的主机可以在以后根据需要模拟为该用户。

默认情况下，域控制器配置有不受约束的委托。

2-约束委派

约束委派配置可以将帐户委派给的服务，但是不能跨林工作。

* 使用“ TRUSTED\_TO\_AUTHENTICATE\_FOR\_DELEGATION”标志更新对象的userAccountControl属性
* 使用在“委派”选项卡上配置的SPN填充msDS-AllowedToDelegateTo属性

### 3-基于资源的约束委派

托管服务的资源没有指定可以委托给哪个对象，而是指定可以委托给哪个服务的对象。

通过使用允许委派给目标资源的对象的SID填充目标资源上的msDS-AllowedToActOnBehalfOfOtherIdentity属性来配置基于资源的约束委派。

**Authentification章节实验报告**

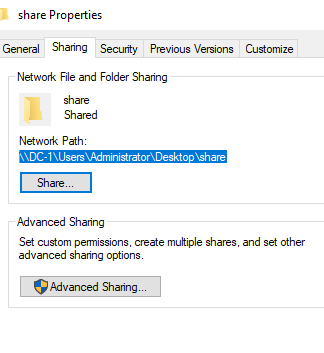
**Kerberos实验部分**

**Exercise 1-实验环境**

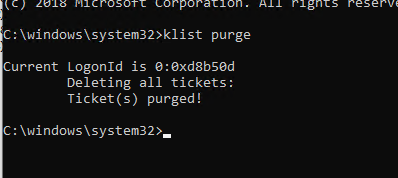
|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | IP | OS | 角色 |
| DC-1 | 192.168.2.68 | Windows 2019 | DC&DNS |
| C-1 | 192.168.2.59 | Windows 2010 | Client |
| DC-2 | 192.168.2.67 | Windows 2019 | Server |

**Exercise 2-抓取正常访问共享网络包，查看Kerberos验证流程**

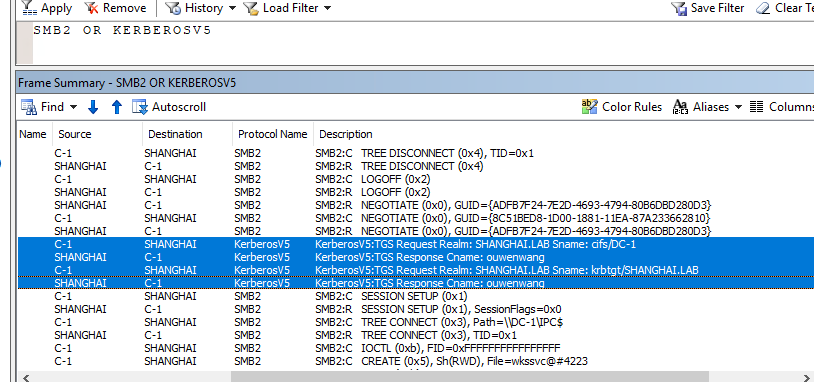
新建共享文件夹share



清除票据



过滤访问共享文件的查询 filter： smb2 or kerberosv5



**Exercise 3-抓取登陆网络包，查看Kerberos验证流程**

**流程：登陆adminnistrator, 开启网络抓包**

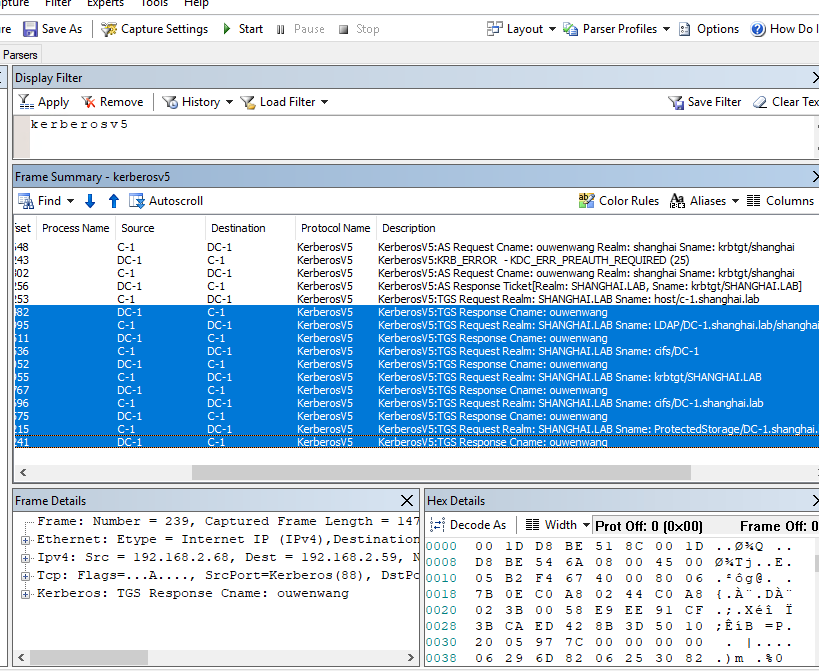
**Switch user，登陆ouwen wang账户，**

**登陆成功后，切换回到adminnistrator,**

**停止抓包，检查抓包流量记录**

**Filter: Kerberosv5**

**结果如下**

****

其中第一次AS Request返回KDC\_ERR\_PREAUTH\_REQUIRED(25)是由于第一次提供的请求中没有提供PaData: PA-ENC-TIMESTAMP ，即加密的客户端时间，因此DC要求再次提供。之后可以看到申请了host/c-1.shanghai.lab的服务票据（登陆）以及其他组策略需要的票据。

**Exercise 4重复的SPN以及排错**

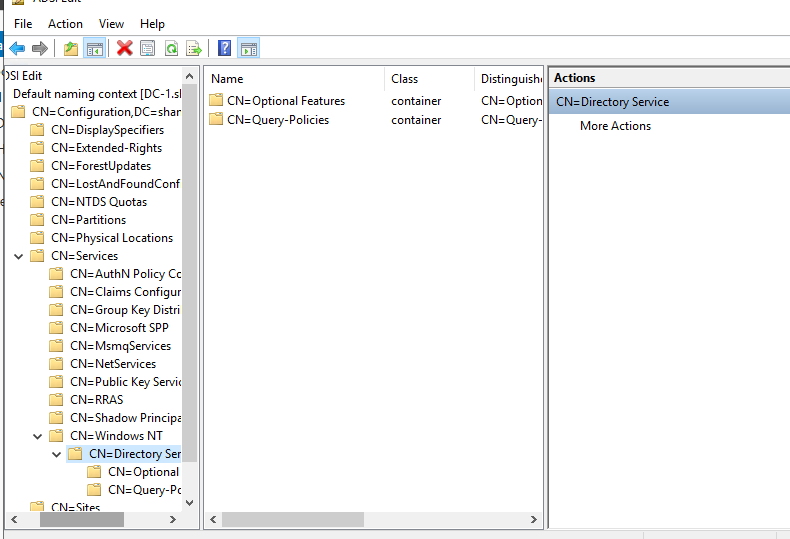
Disable SPN duplicate check

由于在2012服务器版本以上是默认不允许duplicate SPN存在，所以我们要手动把这个默认设置改成启用

方法如下

打开ADSI

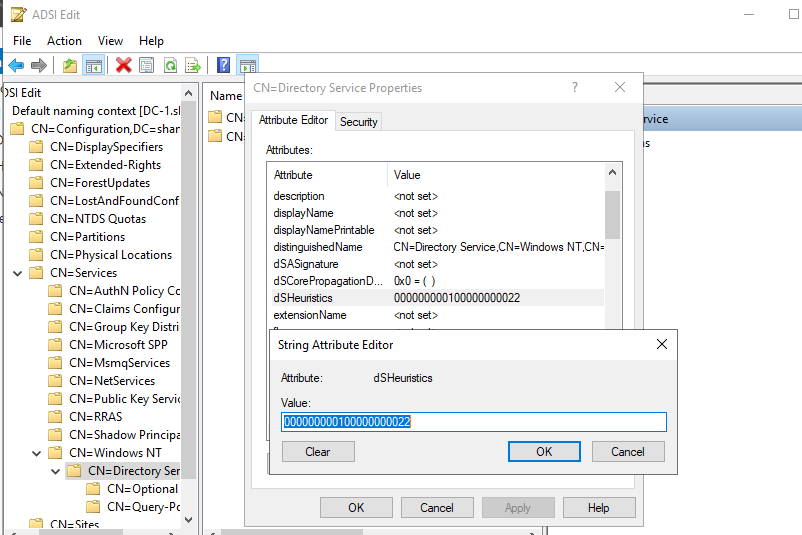
在shanghai.lab的configuration分区中找到以下路径



右击属性，编辑修改“**dSHeuristic**”属性

属性数值为**000000000100000000022**

**禁用了duplicate SPN check**

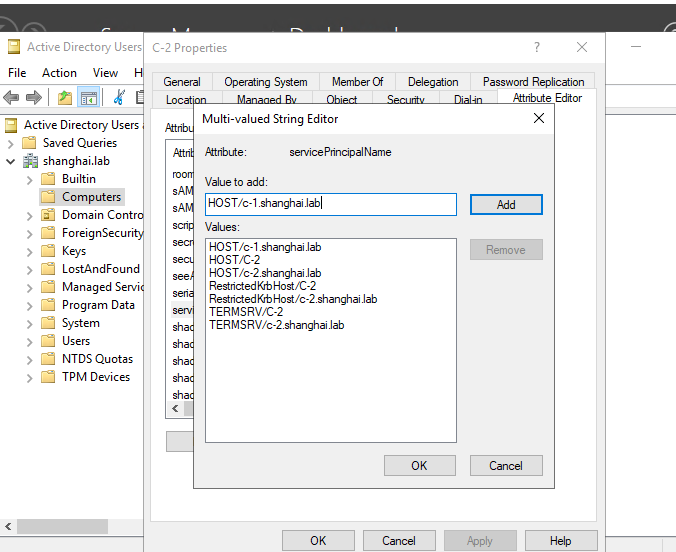


在ADUC内找到计算机c-2

开启advanced选项查看计算机属性

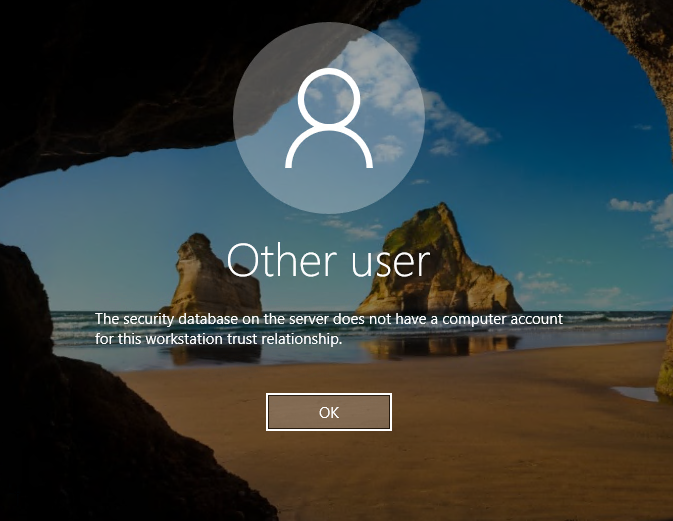
找到servicePrincipleName

添加值：HOST/c-1.shanghai.lab



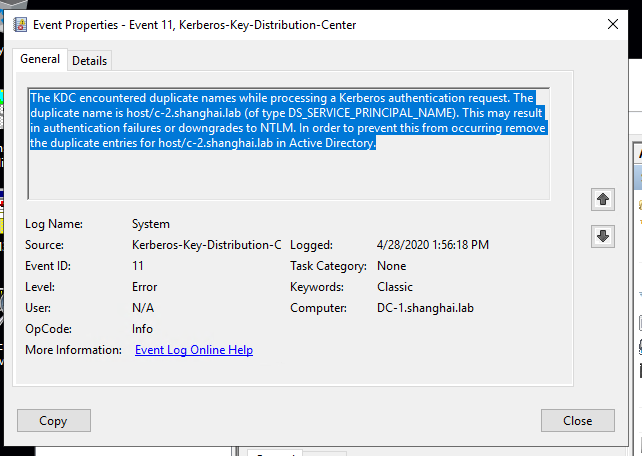
**按照切换登陆的方式登陆账户ouwen wang**

**结果如下图**

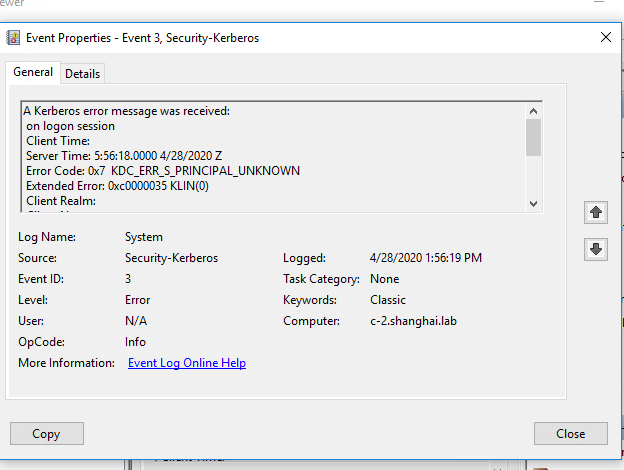


SPN duplicate错误日志记录

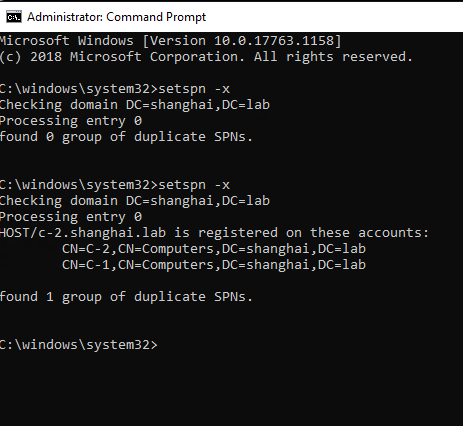
DC上



客户端



采用命令setspn -x查找重复的SPN

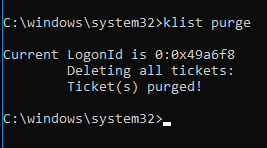


查询前后SPN重复的对比

**Exercise -设置SPN在错误的账号上，查看共享访问并排查**

1-把C-1的host/c-1.shanghai.lab删除并添加到其他计算机账号c-2上，此时域中只有c-2拥有该SPN。

Klist purge清楚缓存

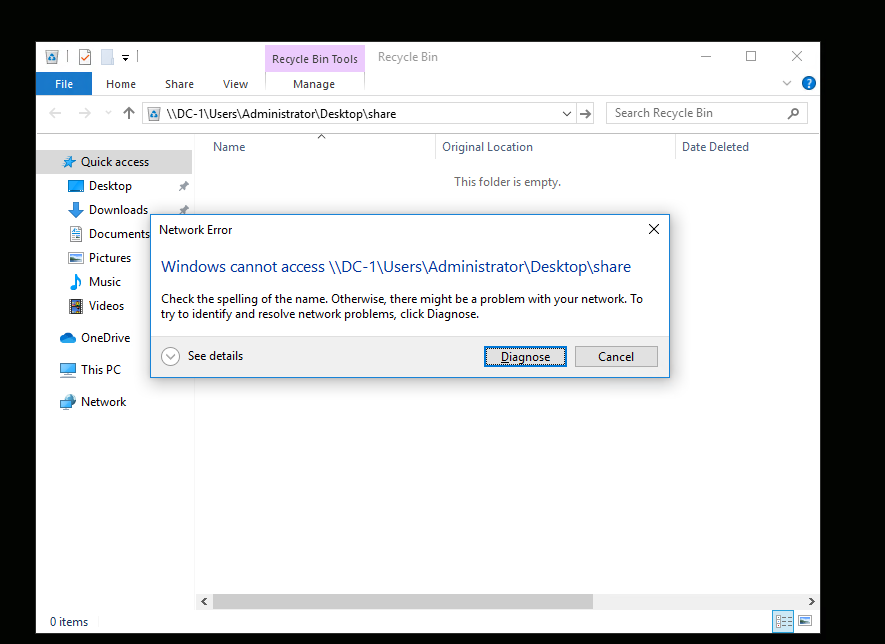


登陆Administrator,开启抓包

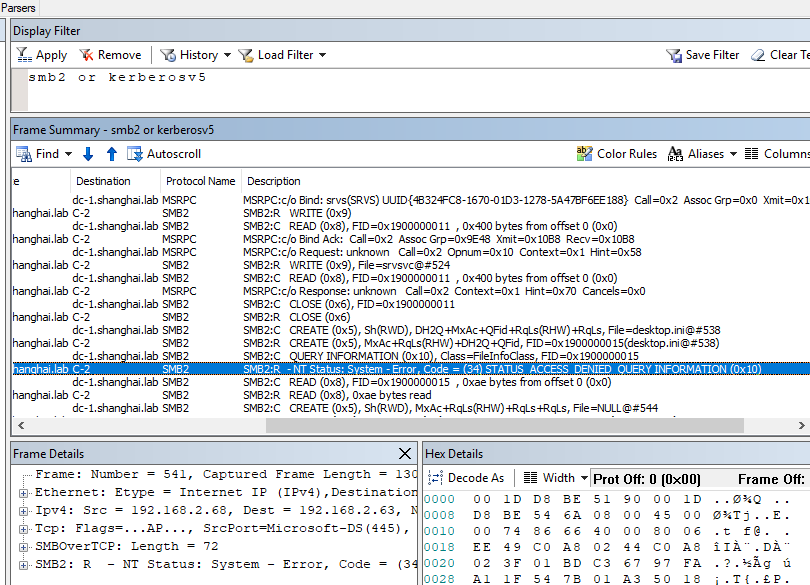
切换到ouwen wang账户

访问**[\\DC-1\Users\Administrator\Desktop\share](\\\\DC-1\\Users\\Administrator\\Desktop\\share)**

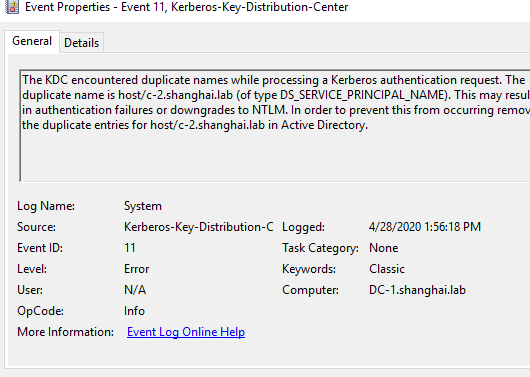
**显示无法访问，如图**

****

**切换回到ADMIN，结束抓包，筛选smb2 or kerberosv5, 获得结果如下:**



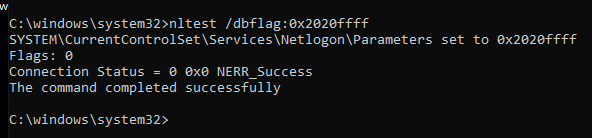
查看系统日志



**NTLM实验部分**

**Exercise1-抓取正常访问共享网络包，查看NTLM验证流程**

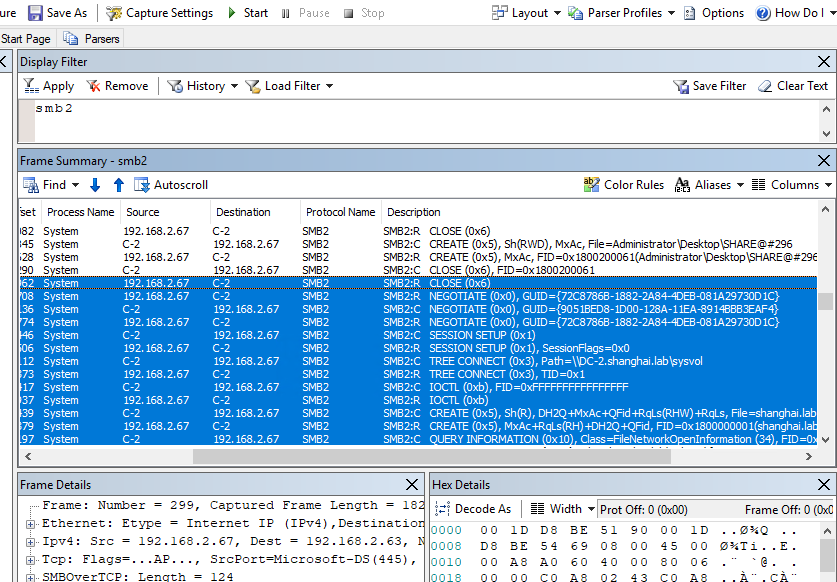
1. 开启netlogon debug日志：nltest /dbflag:0x2080ffff

****

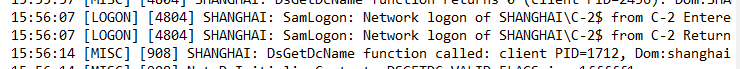
**Exercise2-在DC-2上开启网络抓包，，再从客户端C-1上访问DC-1上**

[**\\DC-2\Users\Administrator\Desktop\share**](file:///\\DC-2\Users\Administrator\Desktop\share)

**获得如下结果**

****

在DC-2上查看c:\windows\debug\netlogon.log，可以观察到如下记录

****

**Exercise3-从Hyper-V主机访问虚拟机共享，查看网络包**

1. 在Hyper-V主机上采用[\\192.168.1.11\share](file:///\\\\192.168.1.11\\share)访问虚拟机共享并抓包

此时注意到弹出验证框，输入虚拟机 的域和域名，即可访问。

此时可以发现不同域的机器可以相互访问，即使没有trust。（Hyper-V主机在公司的fareast域，而虚机在自己创建的域中。）

因此，在NTLM验证中，系统是不会检查域名，只会验证用户名和密码，即使客户端提供了域名。

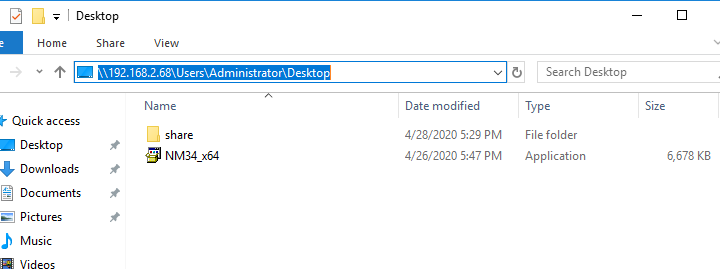
**由于连不上HYPER-V主机，所以直接跳过，**

**因为整个实验的目的是验证NTLM是基于网络验证，而不一定要基于域环境。**

**所以改成在VMAS中增加2台未加域的客户端来做实验**

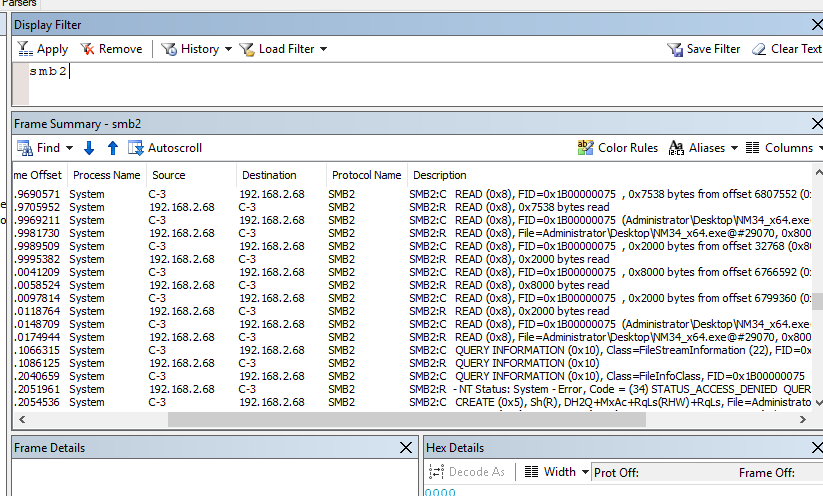
**加入c-3和c-4客户端**

**在c-3上输入**[**\\192.168.2.68**](file:///\\192.168.2.68)**实现在未加域但在同一子网工作组下的网络验证访问共享文件夹**

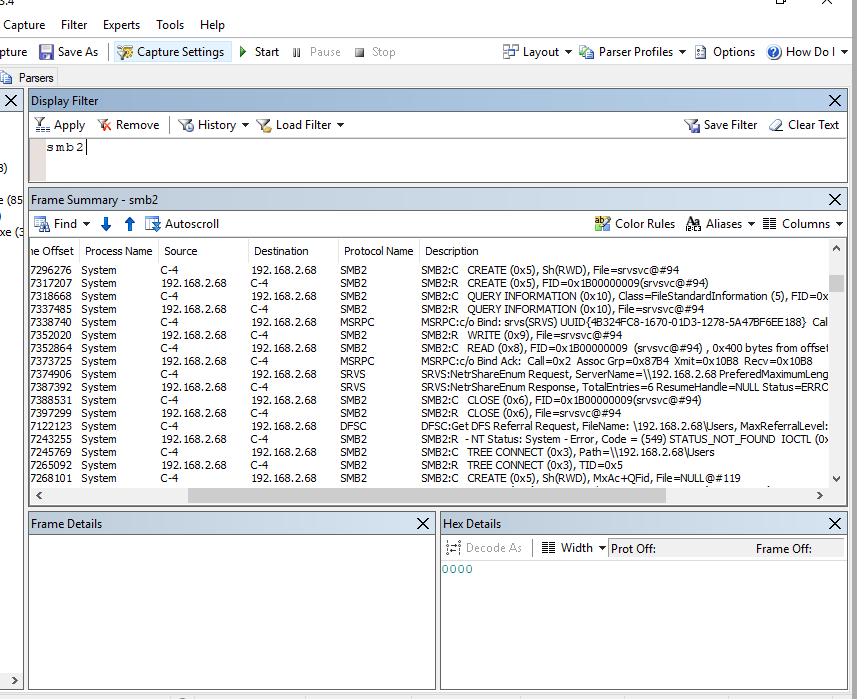
****

**日志抓包如下:filter smb2**

**c-3上抓包**

****

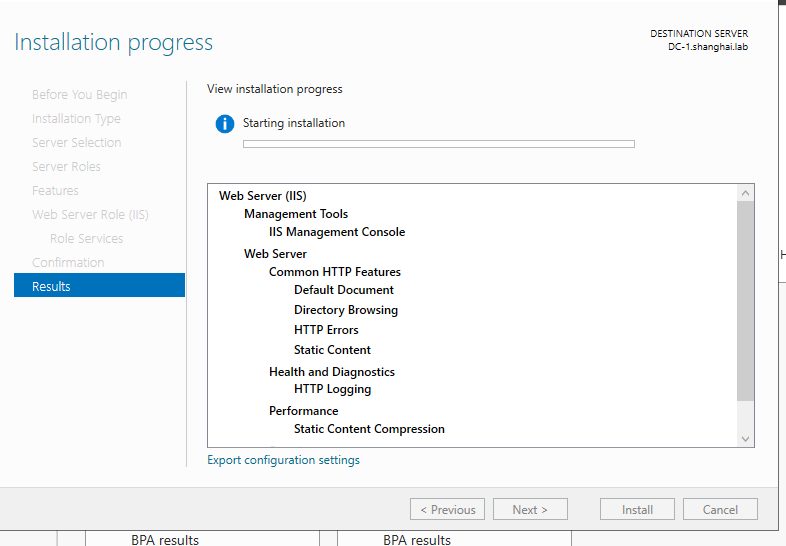
**c-4上抓包**

****

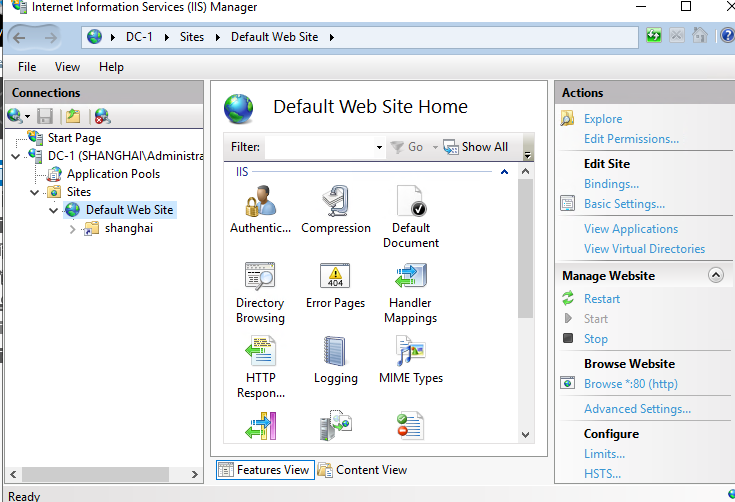
**Kerberos delegation实验部分**

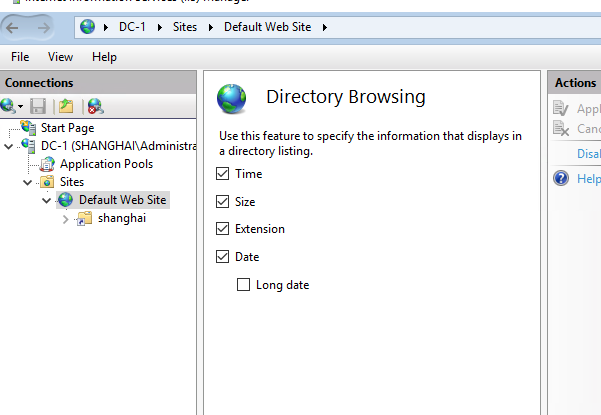
**准备工作：**

1. **安装IIS**



1. 创建virtual directory

****

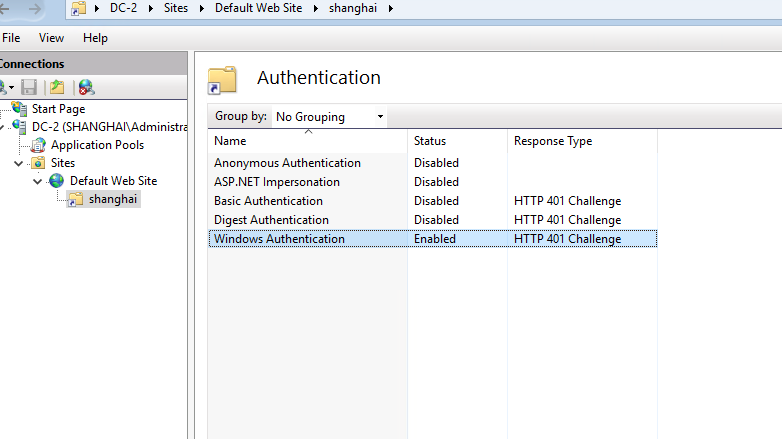
****

进行到这一步，发现我的IIS和KME材料上的界面稍有不同，我没有ASP feature，查看Web.config也没有system.web section.

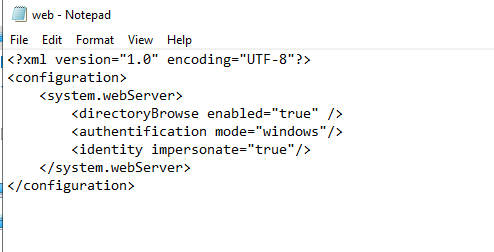
检查安装组件是否有遗漏

发现没有安装windows authentification ,勾选后重新安装。

重新进行配置



修改web.config文件



关于实验指导并没有指导如何设置账户委派，来实现通过委派某个账户，来集中管理对于某个关键服务，比如数据库的访问，前端其他用户可以通过这个被委派的账户来实现访问数据库服务。想学习一下关于这一点该如何设置