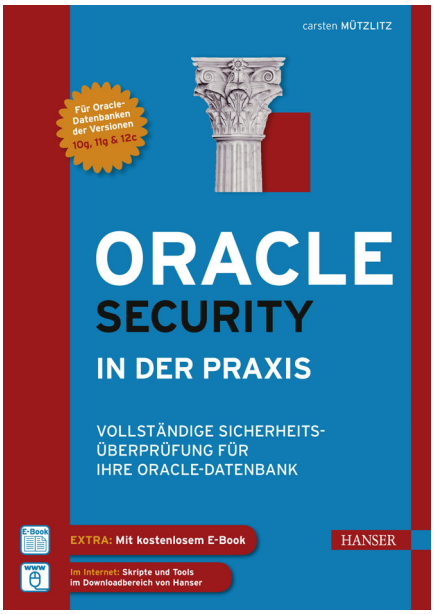
HANSER




# Leseprobe

zu

# „Oracle Security in der Praxis“

von Carsten Mützlitz

ISBN (Buch): 978-3-446-43869-9
ISBN (E-Book): 978-3-446-43923-8

Weitere Informationen und Bestellungen unter
http://www.hanser-fachbuch.de/978-3-446-43869-9
sowie im Buchhandel

# 5 Sicherheit einer Oracle-Datenbank prüfen

Die Einführung in die Best Practices und die neuen 12c-Features führt uns jetzt zum Hauptthema: die Sicherheitsüberprüfung einer Oracle-Datenbank.

Jede Datenbankuntersuchung konzentriert sich auf drei wesentliche Grundbedrohungen: Verlust der Integrität, Verlust der Vertraulichkeit sowie Verlust der Verfügbarkeit. Diese drei Verlustarten nutzt auch das standardisierte Bewertungssystem für Schwachstellen, kurz CVSS[1] (Common Vulnerability Scoring System), zur Bewertung eben dieser Schwachstellen. Und auch Oracle nutzt diese Grundbedrohungen und Verlustarten, um in den Risikomatrizes[2], bekannt aus den Oracle-Veröffentlichungen zu Security Alerts und kritischen Patches (siehe *http://www.oracle.com/technetwork/topics/security/alerts-086861.html*), die Auswirkungen auf die Systeme klarzustellen.

Innerhalb dieser Grundbedrohungen werden in meiner Datenbanküberprüfung fünf wesentliche Kategorien untersucht, die zusammen ein Bild über den Sicherheitszustand einer Datenbank ergeben. Diese Kategorien sind allgemeingültig und gelten auch für andere Datenbankhersteller.

- **Konfiguration der Datenbank:**
  Ist die Datenbank sicherheitstechnisch optimal konfiguriert?
- **Überwachung der Datenbank:**
  Wurden wesentliche Policies aktiviert, um die wichtigen Datenbankaktivitäten zu protokollieren? Sind die Objekte mit den unternehmenskritischen Daten in die Überwachung eingebunden? Existiert ein Reporting für die einfache Auswertung der Protokolle bzw. Tools zur Verbesserung der Nachhaltigkeit?
- **Verfügbarkeit der Datenbank:**
  Reicht das Setup aus, um die Datenbank vor dem geforderten Verlust der Verfügbarkeit zu schützen? Wenn beispielsweise ein Datenverlust von maximal 30 Minuten gefordert ist, kann diese Anforderung erfüllt werden?

---

[1] Erklärung und Beschreibung: *http://www.first.org/cvss*; Kalkulator für Datenbankschwachstellen nach CVSS: *http://www.security-database.com/cvss.php*

[2] Beispiel: Critical Patch Update (CPU) April 2013, *http://www.oracle.com/technetwork/topics/security/cpuapr2013-1899555.html* und nach unten blättern, bis die Risikomatrizes erscheinen.

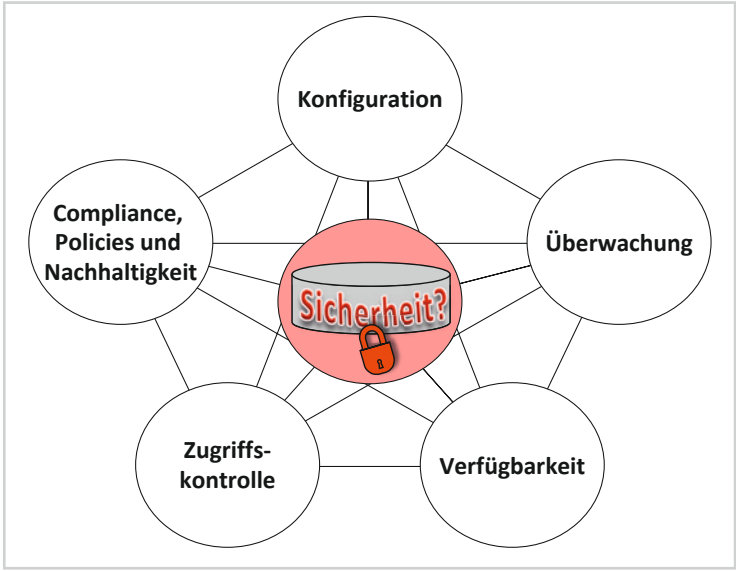



**Bild 5.1** Untersuchte Kategorien

- **Zugriffskontrolle der Datenbank**:
  Wer greift auf meine Datenbank zu und welche Konzepte sind implementiert, die eine Policy durchsetzen nach dem Prinzip „Jeder bekommt die Rechte, die er benötigt, nicht mehr und nicht weniger (Least Privilege)“?
- **Compliance/Nachhaltigkeit der Datenbank**:
  Die Datenbank muss als wesentlicher Teil einer Anwendung auch in die Betrachtung der Compliance-Anforderungen einbezogen werden. Hier wird untersucht, ob wesentliche nachhaltige Konzepte implementiert wurden. Es geht im Wesentlichen um Transparenz und Kontrolle. Zusätzlich wird die Erfüllung externer Einflüsse wie Gesetze geprüft.

Auf Basis der Auffälligkeiten, die in diesen Kategorien gefunden werden, lässt sich sehr gut der Sicherheitszustand der Datenbank in Bezug auf den Schutzbedarf bestimmen. Welche Untersuchungen in den einzelnen Kategorien sinnvoll sind, wird in den nachfolgenden Abschnitten detailliert beschrieben.

Die nachfolgenden Abfragen und Untersuchungen selektieren Informationen aus der Datenbank. Deshalb ist es auch wichtig, parallel zu den Untersuchungen in der Datenbank ein Interview durchzuführen, welches das komplette Bild, in dem die Datenbank eingebettet ist, vervollständigt. Ziel ist es, sich mit dem System vertraut zu machen und dessen Zweck, externe Einflüsse sowie Architektur zu kennen. Ein Datenbankbetreiber, der sich mit seinem System auskennt, braucht lediglich die nachfolgenden Skripts (siehe Abschnitt 5.7) auszuführen. Alles Weitere, beispielsweise welche Prozesse außerhalb der Datenbank existieren, sollte bekannt sein. Für jene Leser dieses Buchs, die die zu untersuchende Datenbank nicht kennen, habe ich ein paar Fragen, die helfen sollen, das Bild der Datenbank zu vervollständigen, als Beispiel in den Anhang übernommen.

Nutzen Sie das Buch als Nachschlagewerk für die einzelnen Untersuchungen, die für die Überprüfung des Sicherheitszustands einer Oracle-Datenbank benötigt werden. Alle aufgezeigten SQL-Abfragen und Skripts finden Sie im Download-Bereich vom Hanser Verlag *http://downloads.hanser.de/*. Es sind Skripts vorbereitet, die Daten automatisch aus den Datenbanken lesen. Die einzige zu erledigende Aufgabe nach der Ausführung ist die korrekte Interpretation der Daten in den generierten Log-Dateien.

Nachfolgend finden Sie die einzelnen Abfragen in den einzelnen Kategorien. Ich werde die einzelnen Abfragen genau erklären und meine Erkenntnisse und weiterführende Fragen aufzeigen.

Mit der neuen Version 12 c der Oracle-Datenbank hat sich die Architektur der Datenbank verändert. Es gibt nun sogenannte Container. Ich gehe mal davon aus, dass nach der Veröffentlichung der Version 12c erst mal nur Non-CDB-Datenbanken existieren, die von Version 11g auf Version 12c umgezogen sind oder sogar migriert wurden. Für solche Datenbanken werden die nachfolgenden Abfragen (bzw. das Tool aus dem Download-Bereich) einmal gegen die Datenbank ausgeführt, gleichgültig, um welche Version der Datenbank (10g, 11g, 12c) es sich handelt.

Existieren in der 12c-Datenbank bereits verschiedene Container (also CDB und PDBs), muss der Security Check gegen alle Container-Datenbanken ausgeführt werden. Die Trennung zwischen dem Root-Container und den pluggable Datenbanken der Anwendungen sollte dann ersichtlich sein. Eine 12c-Datenbank mit einem Root und zwei PDBs muss den Security Check also drei Mal ausführen, so dass jeder Container untersucht wird.

Es gibt Abfragen für Datenbanken der Versionen 10g, 11g und 12c. Alle diese Datenbanken befinden sich derzeit im aktiven Betrieb. Natürlich gibt es auch ältere Versionen, die sich im Einsatz befinden, doch das sollten nur noch ganz wenige sein.

## 5.1 Datenbankkonfiguration

Als erster Untersuchungsbaustein wird die Konfiguration der Datenbank überprüft. Insbesondere werden die SQL*Net-Konfiguration, der Aufbau der Datenbank (Datenbankdateien, *init.ora*, Umgebung etc.), das Patching und die Prüfung vorhandener Konzepte untersucht. Hier sind verschiedene Abfragen notwendig, die in den nachfolgenden Abschnitten vorgestellt werden.

### 5.1.1 Abfrage 1001: erste Datenbankinformationen

Die nachfolgende Abfrage liefert Informationen über die Oracle-SID, das Erstelldatum, Archive LOG, den Flashback und die Standby-Datenbank.

Für 12c wird zusätzlich die Spalte CON_ID angezeigt. Diese Spalte ist in der Abfrage auskommentiert.

**Listing 5.1** ABFRAGE 1001 (10g, 11g, 12c)

```
set heading on echo off feedback off verify off underline on timing off;
set linesize 400
prompt
prompt --1001
prompt Database Information
prompt ============================
column name format a8
column log_mode format a12
column platform_name format a30
column guard_status format a10 heading Guard|Status
column flashback_on format a9 heading flashbank|on
column controlfile_type format a11 heading controlfile|type
column open_mode format a11 heading open|Mode
column protection_mode format a11 heading protect|mode
column database_role format a11 heading DB|role
column switchover_status format a11 heading switch|Status
column force_logging format a11 heading Force|logging
select name,
       created,
       log_mode,
       platform_name,
       guard_status,
       flashback_on,
       controlfile_type,
       open_mode,
       protection_mode,
       database_role,
       switchover_status,
       force_logging
from v$database;
column name clear
column log_mode clear
column platform_name clear
column guard_status clear
column flashback_on clear
column controlfile_type clear
column open_mode clear
column protection_mode clear
column database_role clear
column switchover_status clear
column force_logging clear
```

Ein typisches Ergebnis würde sich wie folgt darstellen:

**Listing 5.2** Ergebnis 1001:

```
--1001
Database Information
============================
                                    Guard  flaank    contrile open  protect  DB      switch   Force
NAME CREATED   LOG_MODE   PLATFORM_NAME Status bank on  filetype  Mode  mode     role    Status   logging
---- --------- ---------- ------------- ------ -------- --------- ----- -------  ------- -------- -------
ORCL 30-OCT-09 NOARCHIVE  Linux IA      NONE   NO       CURRENT   READ  MAXIMUM  PRIMARY NOT      NO
               LOG          (32-bit)                              WRITE PERFOR-          ALLOWED
                                                                      MANCE
```

**Erkenntnis**

Die Datenbank wurde am 30.10.2009 erstellt und hat die SID=ORCL. Sie wird auf einem Linux-System betrieben, verfügt über keine Standby-Datenbankanbindung und eine Flashback-Datenbank ist nicht aktiviert. Des Weiteren werden keine Archive Logs geschrieben. Es sind keine Funktionen für eine bessere Verfügbarkeit und auch kein Schutz gegen menschliche Fehler vorhanden (Flashback-Datenbank).

**Weiterführende Fragen:**

- Wie schützt man sich vor menschlichen Fehlern (z. B. truncate Table (also ein Entleeren des Tabelleninhalts ohne die Möglichkeit eines Rollback) ohne Flashback-Datenbank?
- Ist eventuell doch eine Standby-Datenbank notwendig aufgrund von Hardware- oder Performance-Anforderungen (z. B. Auslagern des Reportings oder Backups auf die Standby-Datenbank)?
- Ohne Archive Logs ist ein schnelles Recovery nicht möglich und ein Datenverlust abhängig von der Backup Methode sehr wahrscheinlich, ist das so gewollt?

### 5.1.2 Abfrage 1001.1: Datenbankinformationen zu 12c (Container)

Die nachfolgende Abfrage zeigt erste Informationen zu den Containern in einer 12c-Datenbank. Oracle unterscheidet in der Version 12c prinzipiell zwischen drei Arten von Datenbanken:

- NON-CDB
  Das sind die Datenbanken, also die Single-Instanzen, die wir bis zu der 11g-Datenbankversion kennen. Mit der 12c nennen wir diese Instanzen Non-CDB.
- CDB
  Das ist die Container-Datenbank, also das ROOT.
- PDB
  Pluggable Datenbank, die in einem Container ROOT angelegt ist.