Segregation of Duty: Konzeption und Einführung

Sven Vetter
Senior Technology Manager
Trivadis AG, Zürich

DOAG SIG Security
03.03.2011

trivadis
makes IT easier.

Agenda

- Ausgangslage
- Datenkategorisierung
- Zugriff auf Server (DBA)
- Zugriff auf Server (User)
- Zugriff auf Datenbank (DBA)
- Zugriff auf Server (User)
- Zugriff auf Applikationen / Auditing
- Offene Punkte

Ausgangslage

- Externe Revision eines Kunden stellt diverse "Mängel" fest
- Die kritischsten waren:
  - DBA hat Vollzugriff auf alle Daten
  - root sowieso
  - Häufig wird mit Sammelbenutzern gearbeitet (nicht nur oracle, sys, system, ...) auch Applikationsbetreuer
  - Nicht nachvollziehbar, wer etwas gemacht hat
- Umfeld
  - Ca. 200 produktive Datenbanken
  - Neben Oracle auch Microsoft und Sybase – aber das ist nicht der Scope
  - Daten nicht nur in Europa

Auftrag

- Erstellen eine Konzepts für "Gewaltentrennung"
- Erstellen eines Prototypes im Testumfeld
- Ausrollen des Konzepts vorerst auf eine Applikation (betrifft mehrere DBs
- Schulen der DBAs und anderer beteiligten Personen

Agenda

- Ausgangslage
- Datenkategorisierung
- Zugriff auf Server (DBA)
- Zugriff auf Server (User)
- Zugriff auf Datenbank (DBA)
- Zugriff auf Server (User)
- Zugriff auf Applikationen / Auditing
- Offene Punkte

Datenkategorisierung (1)

- Es sind zu viele Datenbanken, um in vernünftigen Zeitrahmen alles einzuführen
- Für manche Datenbanken ist es auch nicht notwendig
- Definition von Securityklassen (öffentlich, intern, vertraulich, geheim) und Auswirkungen, wenn Daten gestohlen oder manipuliert werden
  - Wettbewerbsnachteile
  - Direkte Geschäftsschädigung, zusätzliche Kosten
  - Öffentliches Vertrauen / Imageverlust
  - Geschäftsunterbruch
  - Gesetzliche Haftung
  - ...

Datenkategorisierung (2)

D) Impact Analyse

Vertraulichkeit

	Schadenszenarien	Schadensausmass A	B	C	Beschreibung
1	**Wettbewerbsnachteile** Wie schädlich sind die Auswirkungen, wenn der Konkurrenz Daten offen gelegt würden?	☐	☒	☐	
2	**Direkte Geschäftsschädigung** Wie hoch wäre der direkte Schaden durch die Offenlegung von Informationen bzw. in welchem Ausmass könnten dadurch Geschäfte verloren gehen?	☐	☐	☒	
3	**Öffentliches Vertrauen** In welchem Ausmass können durch die Offenlegung von Informationen das Vertrauen der Kunden, das öffentliche Image und der gute Ruf oder das Vertrauen der Aktionäre und Lieferanten gestört werden?	☐	☐	☒	
4	**Zusätzliche Kosten** Wie hoch sind die entstehenden Zusatzkosten, wenn Informationen öffentlich werden?	☐	☒	☐	
5	**Gesetzliche Haftung** Welche Auswirkungen hat die Offenlegung von Informationen auf gesetzliche oder vertragliche Verpflichtungen?	☐	☒	☐	
6	**Betrug** Wie schädlich wäre ein Betrug, der durch Offenlegung von Informationen begangen wird?	☐	☐	☒	
	Höchste Schadenstufe (Maximum der oben stehenden Einschätzungen)	☐	☒	☐	

Datenkategorisierung (3)

- Definition pro Klasse:
  - Benötigter Schutz
  - Bekanntes Risiko
  - Massnahmen (vom Passwortschutz bis zu Database Vault)
  - Kosten (!)
- Gemeinsam mit Applikationsverantwortlichen wurde pro Applikation die Security-Klasse definiert

Datenkategorisierung (4)

- Ein Script für die Überprüfung der Einhaltung der Massnahmen wurde erstellt
- Dies ist inzwischen ein "offizielles" Trivadis Tool (Tvd-SecAudit)

7. ASO (Advanced Security Option)			
7.1. Check ASO installed	Passed	ASO is installed	*[aso100]* ASO muss installiert sein, um Datenfiles und Netzwerkverke
7.2. Column encryption	Passed	Encrypted column(s): BANK.CUSTOMERS: CUST_NAME (AES 256 - salt) BANK.CUSTOMERS_VALUES: CUST_TURNOVER (3 Key Triple DES 168 - salt)	*[aso210]* Durch Column Encryption werden die Daten transparent fue Oracle 11g kann anstatt mit Spalten- auch mit Tablespaceverschluess
7.3. Column encryption - check old algorithm	Failed	3DES encrypted column(s): BANK.CUSTOMERS_VALUES: CUST_TURNOVER (3 Key Triple DES 168 - salt)	*[aso215]* Veraltete und unsichere Encryption Algorithmen wie 3DES s ressourcenintensiver.
7.4. Tablespace encryption	Passed	Encrypted tablespace(s): ENC_TEST (AES256) ENC_TEST3 (3DES168)	*[aso220]* Durch Tablespace Encryption werden die Daten transparent
7.5. Tablespace encryption - check old algorithm	Failed	3DES encrypted tablespace(s): ENC_TEST3 (3DES168)	*[aso225]* Veraltete und unsichere Encryption Algorithmen wie 3DES s ressourcenintensiver.
7.6. Network encryption	Passed	Encrypted session(s) - osuser: oracle: AES256	*[aso400]* Durch Netzwerk Encryption werden die Daten transparent fu Werden verschluesselte Session gefunden, ist zu kontrollieren, ob die

Auditieren oder Limitieren?

- Bei den Diskussionen über die Kategorisierung wurde eine wichtige Frage beantwortet:
- Reicht es, wenn die Zugriffe der DBAs (möglichst) lückenlos überwacht werden – oder soll man versuchen, diese zu limitieren (einschliesslich 4-Augen-Prinzip)?
- Hier wurde der Überwachungsansatz gewählt (jedenfalls momentan, das Konzept sieht eine Erweiterung durchaus vor)
- Deshalb kann auf Tools wie z.B. Database Vault, TDE, ... verzichtet werden
- Trotzdem wurde dies im Laufe des Projekts getestet und dokumentiert → wir wären ready

Agenda

- Ausgangslage
- Datenkategorisierung
- Zugriff auf Server (DBA)
- Zugriff auf Server (User)
- Zugriff auf Datenbank (DBA)
- Zugriff auf Server (User)
- Zugriff auf Applikationen / Auditing
- Offene Punkte

Zugriff auf Server durch DBAs (1)

- Schnell war klar, das der Zugriff auf den Server besser geregelt werden musst
- Bis dahin hat sich jeder DBA als oracle, jeder OS-Admin als root angemeldet
- Einführung von persönlichen DBA-Accounts:
  - Jeder Administrator arbeitet mit seinem persönlichen Benutzer
  - Dieser existiert nur im LDAP (hat also genau ein Passwort)
  - Er kann alle notwendigen Tätigkeiten durchführen – ohne grosse Einschränkungen bzw. Mehraufwand
  - Seine Tätigkeiten werden protokolliert – und es ist dabei sichergestellt, dass sein persönliche Name sichtbar ist
  - Der Benutzer oracle ist für interaktive Anmeldungen gesperrt

Zugriff auf Server durch DBAs (2)

- "Ummeldung" auf oracle erfolgt per sudo
- Dabei dürfen die DBAs alle Befehle ausführen, also auch eine shell öffnen
- Eine Reihe von Aliasen wurde (zentral) definiert, so dass übliche Programme direkt wie gewohnt ausgeführt werden konnten
- sudo Protokollierung ist eingeschaltet
- Session können per sudo_replay wieder abgespielt werden

Zugriff auf Server durch DBAs - Beispiele

- sudo-Definition:

```
Defaults        log_output
Defaults        log_input
…
User_Alias      ORA_DBA = user1, user2, user3
Runas_Alias     DB = oracle
…
ORA_DBA  ALL=  (DB) NOPASSWD: ALL
```

- Aliase

```
alias runo='sudo -u oracle '
alias swio='sudo -u oracle -i'
alias db.ksh='sudo -u oracle db.ksh'
alias listener.ksh='sudo -u oracle listener.ksh'
```

Agenda

- Ausgangslage
- Datenkategorisierung
- Zugriff auf Server (DBA)
- Zugriff auf Server (User)
- Zugriff auf Datenbank (DBA)
- Zugriff auf Server (User)
- Zugriff auf Applikationen / Auditing
- Offene Punkte

Zugriff auf Server durch Benutzer

- Zuerst einmal müssen alle Benutzer erkannt werden
- Und auch die Befehle, die sie ausführen/brauchen...
- Oft wurde mit unpersönlichen Benutzern gearbeitet...
- Massnahmen:
  - Nur noch persönliche Benutzer zugelassen
  - Benutzer nach "least privilege" Prinzip angelegt
  - sudo-Konzept
  - Persönlicher Benutzer darf sich aber nicht komplett ummelden
  - Seine erlaubten Befehle sind in sudo definiert
  - Aufzeichnung durch sudo
- Gilt auch für Batchuser (Jobs, ...)

Agenda

- Ausgangslage
- Datenkategorisierung
- Zugriff auf Server (DBA)
- Zugriff auf Server (User)
- Zugriff auf Datenbank (DBA)
- Zugriff auf Server (User)
- Zugriff auf Applikationen / Auditing
- Offene Punkte

Zugriff auf Datenbank durch DBAs (1)

- Nicht mehr als sys und system... – oder nicht immer ☺
- Persönliche OS-Benutzer sind in der DBA-Gruppe, können dadurch ein "connect / as sysdba" machen (bzw. ein sq)
- Aber: Im Auditprotokoll wird dann der persönliche OS-Benutzer protokolliert → Nachvollziehbarkeit ist gewährleistet
- Jeder DBA erhält trotzdem einen persönlichen Account, welcher auch die sysdba-Privilegien hat
- Dieser wird für jeden Remote-Zugriff benutzt (für Programme wie TOAD oder auch Grid Control)

Zugriff auf Datenbank durch DBAs (2)

- Für die Nachvollziehbarkeit von DB-Operationen wird im Moment Oracle Standard Auditing benutzt
- Parameter:
  - audit_trail=db, extended
  - audit_sys_operations=true
- Protokolliert werden u.a. jede Benutzung von ANY-Privilegien, da dies auf eine Umgehung des Berechtigungskonzepts hinweist
- sys und system wurden auf unmögliche Passwörter gesetzt

Agenda

- Ausgangslage
- Datenkategorisierung
- Zugriff auf Server (DBA)
- Zugriff auf Server (User)
- Zugriff auf Datenbank (DBA)
- Zugriff auf Server (User)
- Zugriff auf Applikationen / Auditing
- Offene Punkte

Zugriff auf Datenbank durch interaktive Benutzer

- Kein Sammelbenutzer mehr erlaubt
- Zwei Varianten des kontrollierten Zugriffes:
  - Persönliche Benutzer nach least privilege Prinzip
    - Schwierig, da sehr viele Benutzer und die Anforderungen nicht bekannt
    - Umsetzung nur für einige Applikationen
  - Sammelbenutzer bleibt bestehen, wird aber gesperrt
    - Dafür werden Proxybenutzer angelegt, die die Rechte "erben"

```
CREATE USER user01 IDENTIFIED BY userpwd;
ALTER USER hr GRANT CONNECT THROUGH user01;
CONNECT user01[hr]/userpwd
```

    - Leider unterstützen nicht alle Applikationen diese Connect-Syntax

Agenda

- Ausgangslage
- Datenkategorisierung
- Zugriff auf Server (DBA)
- Zugriff auf Server (User)
- Zugriff auf Datenbank (DBA)
- Zugriff auf Server (User)
- Zugriff auf Applikationen / Auditing
- Offene Punkte

Zugriff auf Applikationen / Auditing

- Applikationsverantwortliche haben security-relevante Tabellen definiert
- Diese werden mit Oracle Standard Auditing überwacht
- Kritisch hierbei ist die Art, wie Oracle Auditing einschaltet:
  - Eigentlich sollten die Zugriffe auf diese Tabellen für alle Benutzer überwacht werden – ausser für spezielle (der Applikations-User)
  - Aber genau das "ausser" geht mit Oracle nicht
  - Wenn ein neuer Benutzer angelegt wird, werden eventuell die Audit-Definitionen vergessen
  - Deshalb wurde eine kleine Applikation geschrieben, die regelmässig die Audit Optionen überwacht und setzt
  - Gesteuert wird diese durch eine Metadatentabelle, die auch Sachen wie `user like 'A%'` oder `user not in ('A,','B')` zulässt

Agenda

- Ausgangslage
- Datenkategorisierung
- Zugriff auf Server (DBA)
- Zugriff auf Server (User)
- Zugriff auf Datenbank (DBA)
- Zugriff auf Server (User)
- Zugriff auf Applikationen / Auditing
- Offene Punkte

Offene Punkte (1)

- Auditing ist heute nicht befriedigend gelöst:
  - Kein zentrales Auditing
    - Manipulierbar für DBA
    - Keine schönen Auswertungen
    - (Noch) keine Alarmierung
  - Schwierige Definition des Auditings
  - Hier sind evaluieren anderer Lösungen (Sentrigo Hedgehog, ...)
- Die sudo-Protokollierung ist auch noch nicht zentralisiert

Offene Punkte (2)

- root ist heute noch nicht gesperrt, aber das ist nur noch eine Frage der Zeit
- sys und system können sich in einigen Applikationen immer noch anmelden, da es die Applikation verlangt ☹
- Die Passwörter der persönlichen DBA-Accounts sind noch nicht synchronisiert, eine Lösung wird momentan gesucht (LDAP/ Enterprise Users, CUA4DB, Passwort-Tools für DBs wie z.B. Cyper Ark)

Fragen …

→ sven.vetter@trivadis.com

Basel · Baden · Bern · Lausanne · Zürich · Düsseldorf · Frankfurt/M. · Freiburg i. Br. · Hamburg · München · Stuttgart · Wien