# Sicherheitstechnische Überprüfung und Zertifizierung von Datenbankinstallationen

Olivier Angyal[1], Ulrich Greveler[2], Nils Tekampe[1]

[1]TÜV Informationstechnik GmbH, 45141 Essen
olivier.angyal@web.de – N.Tekampe@tuvit.de

[2]Fachhochschule Münster, Labor für IT Sicherheit, 48565 Steinfurt
greveler@fh-muenster.de

## Zusammenfassung

Die Sicherheit der Daten, die heute in den verschiedensten Datenbanken von Unternehmen, Behörden und anderen Instituten lagern und verarbeitet werden, ist ein integraler Bestandteil der heutigen Informationssicherheit – sowohl in nationalem als auch internationalem Kontext. Gleichzeitig steigen die Anforderungen an die Funktionalität, die von Datenbankmanagementsystemen (DBMS) zur Verfügung gestellt werden.
Die Sicherheit der Installation und Konfiguration eines DBMS ist dabei immer noch ein Faktor, der häufig vernachlässigt wird. Zwar wurden und werden die von den verschiedenen kommerziellen und nichtkommerziellen DBMS zur Verfügung gestellten Sicherheitsmechanismen häufig auf ihre Effektivität (und Effizienz) überprüft; die Fragestellung, ob diese Mechanismen in einer konkreten Installation des Produkts aber angemessen ausgewählt und korrekt eingestellt wurden, wird allerdings häufig vernachlässigt.
Im Rahmen dieses Beitrages soll ein Ansatz für die Beurteilung der Sicherheitseigenschaften einer Datenbankinstallation vorgestellt werden, mit dem in Zukunft auch die unabhängige Prüfung und Zertifizierung der Sicherheitseigenschaften einer Datenbankinstallation ermöglicht werden kann.

# 1 Ausgangssituation

Die Sicherheit von Datenbanksystemen wird meist auf der Ebene von Produkten bzw. gemäß generischer Sicherheitseigenschaften und nach formalen Kriterien betrachtet [ArVi08]. Während fast alle DBMS, die zur Umsetzung einer sicheren Datenhaltung notwendigen Grundfunktionen mitbringen (dies umfasst die generischen Funktionen zur Zugriffskontrolle, Authentifizierung von Benutzern und zur Auditierung), unterscheiden sich die DBMS verschiedener Hersteller in spezielleren Aspekten solcher Funktionen und insbesondere in der Konfiguration dieser Mechanismen.

Die Wirksamkeit der grundlegenden und auch erweiterten Sicherheitsmechanismen wird bisher ausschließlich auf Produktebene einem standardisierten Prozess der Überprüfung unterzogen. Fast alle großen Hersteller weisen die korrekte Funktionsweise der Mechanismen in verschiedensten Prüfungen und Zertifizierungen nach. Als prominentes Beispiel seien hierbei

Evaluierungen und Zertifizierungen nach *Common Criteria* genannt, deren Resultate auch international anerkannt werden. [CCRA08]

Zum Zeitpunkt der Prüfung von Datenbanksystemen fehlt allerdings in aller Regel eine sehr wichtige Information: Es ist nicht bekannt, welche Art von Daten später in einem DBMS gespeichert werden und welchen Schutzbedarf diese Daten haben bzw. welches Sicherheitskonzept sich auf die Datenhaltung auswirkt. Auch spezifische Eigenschaften der Einsatzumgebung einer Datenbank sind in aller Regel nicht bekannt. Allenfalls kann von einem typischen Einsatzszenarium ausgegangen werden, aus dem sich nur verallgemeinerte Aussagen der Art: „*Dieses DBMS ist zur Speicherung und Verarbeitung von Daten mit Schutzbedarf X geeignet in einer Umgebung vom Charakter Y*“ gewinnen lassen.

Viel schwerer aber wiegt die Tatsache, dass im Rahmen solcher Prüfungen die konkrete Konfiguration und Umsetzung der von einem DBMS angebotenen Mechanismen in aller Regel nicht betrachtet werden kann. Nur selten lässt sich die Umsetzung einer Maßnahme bereits durch das Produkt umsetzen (z. B. in Fällen, in denen ein Mechanismus nicht deaktiviert oder konfiguriert werden kann sondern stets aktiv ist).

## 1.1 „Datenpannen“

Die jüngeren Beispiele von „Pannen“ in der Datenhaltung bei großen deutschen Unternehmen haben eindrucksvoll demonstriert, dass auf der Ebene von konkreten Installationen eines DBMS häufig selbst einfachste Mechanismen nicht korrekt implementiert werden. Nur zum Teil ist dies der teilweise enormen Komplexität solcher Installationen geschuldet. Häufig ist auch zu erkennen, dass ein umfassender Ansatz für die Sicherheitseigenschaften von Datenbankinstallationen fehlt. Dem Datenbankadministrator obliegt es dann, die Sicherheitseigenschaften sicherzustellen, ohne dass er auf eine dokumentierte Konfiguration oder verifizierbare Überprüfungsmöglichkeit zurückgreifen kann.

Es bleibt zudem die Frage, wie Unternehmen die Sicherheit ihrer eigenen Daten aber nicht zuletzt auch der fremden Daten, die sie speichern und verarbeiten, nicht nur sicherstellen sondern auch Dritten gegenüber die Durchsetzung eines Sicherheitskonzeptes mit Auswirkung auf die Datenhaltung nachweisen können. Der Einsatz einer grundlegend geeigneten bzw. geprüften Technologie kann hier nur ein erster Schritt sein.

# 2 Lösungsansatz

Dieser Beitrag beschreibt einen Ansatz, auf dessen Basis ein Verfahren zur Evaluierung und Zertifizierung von komplexen Datenbankinstallationen realisiert werden kann. Der übergreifende Ansatz besteht dabei aus den folgenden Teilen:

1. Best Practices
2. Evaluierung
3. Zertifizierung

Dieses Paper fokussiert sich dabei auf den ersten und zweiten Schritt: Die Entwicklung von generischen Best Practices, mit deren Hilfe die Sicherheit einer konkreten Datenbankinstallation gewährleistet werden kann und die ihrerseits in der Anwendung einer formale Evaluation unterworfen werden können. Diese Best Practices sollen daher auf der einen Seite dazu geeignet sein, dem Administrator einer Datenbankinstallation als Leitfaden für die Umsetzung

eines umfassenden Sicherheitskonzepts zu dienen; auf der anderen Seite sollen sich von diesen Best Practices Kriterien ableiten lassen, deren Einhaltung im Rahmen einer Evaluierung und anschließenden Zertifizierung nachgewiesen werden kann. Die Best Practices stellen eine Lösung dar, die alle Aspekte berücksichtigt, die die Sicherheit einer Datenbankinstallation beeinflussen können.

## 2.1 Grundlagen von Best Practices

Um einen ganzheitlichen Ansatz für die Sicherheit einer Datenbankinstallation zu gewährleisten, orientierte sich die Entwicklung der Best Practices an den folgenden Grundlagen:

1) Die Best Practices sollen – soweit dies sinnvoll und möglich ist – generisch gehalten werden und unabhängig vom Einsatz eines bestimmten DBMS sein.
2) Die Best Practices sollen sich nicht nur auf technische oder organisatorische Aspekte fokussieren sondern alle für den sicheren Betrieb eines DBMS notwendigen Aspekte abdecken.
3) Die Best Practices sollen technisch so konkretisiert sein, dass sie einem Datenbankadministrator als Leitfaden für die Implementierung der Sicherheit seiner Datenbankinstallation dienen können.
4) Die Best Practices sollen in einer Weise parametrisierbar sein, dass sie sich an die jeweiligen individuellen Eigenschaften einer Datenbankinstallation anpassen lassen.
5) Die Best Practices sollen in einer Art beschrieben sein, in der ihre konkrete Umsetzung in einer realen Installation einfach beurteilt werden kann und sollen Hinweise auf „klassische“ Nachweise enthalten, die von einem Administrator oder Betreiber einer Datenbankinstallation im Rahmen einer Evaluierung bereitgestellt werden können.

## 2.2 Klassen von Best Practices

Die folgende Liste gibt eine Übersicht der Bereiche, denen sich die Best Practices widmen. Diese wurden im Rahmen einer Diplomarbeit [Angy09], gemeinsam betreut von der TÜV Informationstechnik GmbH in Essen und dem Labor für IT-Sicherheit der Fachhochschule Münster, entwickelt.

- *Main policies:* Als Basis für alle weiteren Best Practices werden in diesem Bereich zunächst grundlegende Politiken wie z.B. die Definition eines Schutzbedarfs für die gespeicherten Daten definiert. Diese Klasse enthält die folgenden Best Practices:
  - Physical Security
  - Data Classification
  - Responsibilities and Segregation of Duties
- *Identity and Access Management:* In diesem Bereich werden Best Practices zur Authentifizierung von Benutzern und der Umsetzung von angemessenen Zugriffskontrollpolitiken definiert. Dazu werden auch die Verwaltung von Konten sowie die Benutzung von privilegierten Konten erläutert. Diese Klasse enthält die folgenden Best Practices:
  - Identification
  - Authentication

  - Authorization
  - Use of Privileged Accounts
  - Maitenance of Accounts

- *Network Security:* Dieser Bereich widmet sich der Absicherung des Netzwerks der Datenbankinstallation, um eine „gesunde“ Umgebung der Datenbankinstallation sicherzustellen. Diese Klasse enthält nur eine Best Practice: *Network Security*.
- *Protection from application vulnerabilities:* Die Erfahrungen mit vielen Vorkommnissen in der Vergangenheit haben gezeigt, dass durch Applikationen, denen grundsätzlich der Zugriff auf die Datenbank erlaubt ist, häufig Angriffe auf die Datenbank realisiert wurden. Die Best Practices in diesem Bereich widmen sich daher dem grundsätzlichen Schutz vor solchen Schwachstellen. Diese Klasse enthält die folgenden Best Practices:
  - Separation of DBMS and Application Components
  - Additional and Optional Components
  - SQL Injection
- *Change Management:* Änderungen an einem Datenbankmanagementsystem müssen häufig vorgenommen werden. Da solche Änderungen unerwartete Auswirkungen haben können, müssen sie im Rahmen eines *Change-Management*-Prozess betrachtet und dokumentiert werden. Dies betrifft die Umsetzung von Änderungen am Datenmodell sowie die Installation der in regelmäßigen Abständen veröffentlichten Patches. Diese Klasse enthält die folgenden Best Practices:
  - The Change Management Process
  - Patch Management
- *Encryption:* In diesem Bereich wird die Verschlüsselung von Daten erläutert. Dies betrifft Verschlüsselung von Daten „in transit“ (Verschlüsselung von Kommunikationen mit der Datenbank) und „at rest“ (Verschlüsselung von Daten in der Datenbank und auf dem Medium). Diese Klasse enthält eine einzige Best Practice: *Encryption*.
- *Audit:* Dieser Bereich der Best Practices widmet sich der Umsetzung einer angemessenen Strategie zur Auditierung aller wichtigen Aktionen in der Datenbankinstallation. Diese Klasse enthält eine einzige Best Practice: *Audit*.
- *Backup:* Eine durchgängige Strategie für die Sicherung von Daten gehört zum Kern einer Strategie für eine sichere Datenbankinstallation. Diese Klasse enthält eine einzige Best Practice: *Backup*.

## 2.3 Gliederung einer Best Practice

Die einzelnen Best Practices sind wie folgt gegliedert:

1. *Übersicht:* in der Übersicht wird die Best Practice eingeführt. Dazu gehört eine generelle Beschreibung der Best Practice und die Erklärung, womit sie sich beschäftigt.

2. *Relevanz für die Sicherheit einer Datenbankinstallation:* hier wird beschrieben warum eine Best Practice wichtig für die Sicherheit einer Datenbankinstallation ist. Dazu wird insbesondere dargestellt, welche Bedrohungen gegen eine Datenbankinstallation wirken können, wenn die Best Practice nicht (oder nicht hinreichend) umgesetzt wird.

3. *Ziele:* in diesem Teil der Best Practice werden die Ziele erläutert, die mit Hilfe der Best Practice erreicht werden sollen.

4. *Beschreibung:* in diesem Abschnitt erfolgt eine detaillierte Beschreibung der verschiedenen Aspekte der Best Practice. Zu jeder Best Practice werden Methoden, Maßnahmen und Mechanismen eingeführt, die zu der Verbesserung der Sicherheit in der Datenbankinstallation beitragen können.

5. *Grad der Umsetzung:* Jede Best Practice kann mit verschiedenen Graden umgesetzt werden so dass die Implementierung der Best Practice den Sicherheitsanforderungen der Datenbank angepasst werden kann.

## 2.4 Inhaltsbeschreibung einer Best Practice

Insgesamt existieren 17 Best Practices, deren vollständige Beschreibung den Rahmen dieses Beitrags sprengen würde. Daher wird hier eine Best Practice beispielhaft im Detail beschrieben, die sich im Kern des Themas Datenbanksicherheit befindet: die Best Practice zur „Zugriffskontrolle“ (*Authorization*).

### 2.4.1 Übersicht

Zugriffskontrollen ermöglichen es, den Datenzugriff auf berechtigte Subjekte zu beschränken. Ein Autorisierungsprozess läuft dabei wie folgt ab: Nachdem ein Subjekt eine Operation auf ein Datum anfordert, überprüft das DBMS zunächst, ob das Subjekt über die entsprechenden Rechte verfügt. Wenn dies der Fall ist, wird die Operation ausgeführt; ansonst wird sie abgelehnt.

Zugriffskontrollen basieren auf Privilegien. Ein Privileg kann dabei als die Erlaubnis eine Operation auf eine Ressource durchführen zu können betrachtet werden. Dies kann beispielsweise der lesende oder schreibende Zugriff auf ein Datum oder die Ausführung einer Prozedur in einer Datenbank sein.

Im Falle von DBMS existieren verschiedene Autorisierungsmodelle. Die Bekanntesten sind:

- *Discretionary Access Control* (DAC): in diesem Modell wird der Zugriff auf Objekte auf Basis der Identität des Subjekts beschränkt.
- *Role-Based Access Control* (RBAC): in diesem Modell wird der Zugriff auf Objekte auf Basis der Rollenzugehörigkeit des Subjekts beschränkt.
- *Mandatory Access Control* (MAC): in diesem Modell existieren die Zugriffsberechtigungen auf Objekte in Form von Labels zur Sensibilität von Objekten und Subjekten.

Im ersten Abschnitt der Best Practice werden diese drei grundlegenden Autorisierungsmodelle diskutiert.

Ein wichtiges Konzept der Sicherheit von Datenbankinstallationen ist das Prinzip des „Least Privilege“ (geringste notwendige Berechtigung). Dieses Prinzip wird von Saltzer und Schröder wie folgt definiert: *„Every program and every user of the system should operate using the least set of privileges necessary to complete the job“* [SaSc75].

Über die zuvor genannten Mechanismen hinaus bieten die meisten DBMS-Produkte erweiterte Zugriffskontrollmechanismen an, die eine detaillierte Kontrolle des Datenzugriffs erlauben. Dabei sind zwei Konzepte vorherrschend:

- Feingranulare Sicherheitsmechanismen (auch *fine-grained access control* oder *row-level security*)
- Views

Diese Mechanismen werden im letzten Teil dieser Best Practice erläutert.

### 2.4.2 Relevanz für die Sicherheit einer Datenbankinstallation

Ohne Autorisierungssystem wären Daten für jeden Benutzer uneingeschränkt zugreifbar. Dies würde den unerlaubten Zugriff auf sensible Daten ermöglichen und würde daher ein erhebliches Risiko für die Vertraulichkeit, Verfügbarkeit und Integrität der Daten darstellen.

Darüber hinaus gilt: Wenn Berechtigungen nicht gemäß dem *Least Privilege* Prinzip zugewiesen sind, besteht die Gefahr, dass ein Subjekt Zugriff auf Daten erhält, die er nicht lesen, modifizieren oder ausführen darf. Er würde dann in der Lage sein, unerlaubte Operationen auf diesen Daten auszuführen.

### 2.4.3 Ziele

Das Ziel dieser Best Practice ist es, Richtlinien festzulegen, die mit Hilfe eines Autorisierungsmodells gewährleistetem, dass Subjekte ausschließlich auf Daten zugreifen können, für die sie auch die Berechtigung besitzen.

### 2.4.4 Beschreibung

Erst werden Richtlinien für das Autorisierungsmodell formuliert. Dabei wird zunächst das Konzept der Privilegien und die Autorisierungsmodelle DAC und RBAC erläutert. Im zweiten Abschnitt wird anschließend eine Alternative zu diesen Konzept vorgestellten, das MAC-Autorisierungsmodell. Es folgt ein Ausriss aus der Best-Practice-Beschreibung.

> There are many different privileges for databases, among which one can distinguish two types of privileges:
> - System privileges
> - Object privileges
>
> A system privilege gives the permission to perform an operation on the database and to manage the database objects, such as creating and deleting a table or managing user accounts, database roles and privileges. Accounts (or roles) that are granted system privileges are also known as privileged accounts. Thus, these privileges are often granted only to administrators, like database or account administrators. The privileges often vary according to the DBMS. Some of the most common system privileges are:
>
> - Data Definition Language (DDL) privileges, used to define and alter objects in the database (some SQL Statements: CREATE, ALTER, TRUNCATE, DROP)
> - Data Control Language (DCL) privileges, used to control access to data (some SQL Statements: GRANT, REVOKE)
>
> An object privilege gives the permission to perform an action on an object, for example read or manipulate data inside tables.

**(Ausriss aus der Best Practice 3.5.3: *Authorization* [Angy09])**

Ein Objektprivileg eröffnet die Möglichkeit, Operationen auf Datenbankobjekten durchzuführen, zum Beispiel Lese- und Schreibzugriffe auf Tabellen. Die wichtigsten Objektprivilegien werden in der Best Practice aufgeführt:

> - Data Query Language (DQL) privileges: used to retrieve data from a table of the database (SQL statement: SELECT)
> - Data Manipulation Language (DML) privileges: used to manipulate the data in tables (some SQL statements: SELECT, INSERT, UPDATE, DELETE)
> - Privileges on stored procedures and functions, such as execution permission (SQL statement: EXECUTE)
> - Transaction Control Language (TCL) privileges: used to manage transactions (In SQL, transactions are made using DML statements: COMMIT, SAVE POINT, ROLLBACK)

**(Ausriss aus der Best Practice 3.5.3: *Authorization* [Angy09])**

Ob eine Rolle oder ein Subjekt System- bzw. Objektprivilegen besitzt, ist abhängig von der Sensibilität der Daten, der Umgebung der Datenbank und weiteren Parametern. Diese Privilegien sollten gemäß dem Least Privilege Prinzip zugewiesen werden, das im zweiten Teil dieser Best Practice erklärt wird.

Für die Zuweisung von Priviligien werden die im nächsten Ausriss beschriebenen DAC und RBAC Autorisierungsmodellen am häufigsten benutzt.

> There are different ways to assign privileges in databases. Two of the most widespread approaches are:
>
> - Directly to the subjects (DAC)
> - Indirectly through roles or groups (RBAC)
>
> To assign database privileges directly to subjects, one gives each needed privilege to each single subject.
>
> Privileges can also be given using roles or groups, in accordance with the Role-based Access Control (RBAC) model. A role is an entity of the database or operating system to which one assigns a set of privileges, and that is in turn assigned to subjects. Consequently, in this approach, the user acquires or loses database privileges through roles.
>
> The privileges assigned to role are often based on the functions required for a subject to do its job. For instance, a role can give the permission to a subject to execute DDL statements on the database or DML statements on a certain table. Another example is when two groups of users need to access the same table differently: users belonging to the first group only need read access while users of the second group need in addition to modify the data in the table. To achieve this, one can define two roles; the first would only give read access on the database (SELECT statement in SQL), the second would additionally allow to execute DML statements.

**(Ausriss aus der Best Practice 3.5.3: *Authorization* [Angy09])**

Das RBAC-Autorisierungsmodell hat mehrere Vorteile gegenüber DAC:

- Es vereinfacht die Privilegienstruktur und Verwaltung; der Administrator gewinnt einen klaren Überblick über die zugewiesenen Privilegien.
- Es reduziert die Komplexität und den Aufwand der Privilegienverwaltung und begrenzt so das Risiko von menschlichen Fehlern. Man muss nicht einzeln jedes Privileg zuweisen, sondern nur einmalig Rollen mit den benötigten Privilegien festlegen.

Das Mandatory Access Control (MAC) wird in der Best Practice anschließend wie folgt erläutert.

> The Mandatory Access Control (MAC) model is a military approach that provides an alternative to the concept of privileges. According to the American Department of Defense (DoD), MAC is *"a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity"* [DoD85]. In the MAC model, each object and subject is assigned a sensitivity, on which the authorization system relies to decide whether or not a subject is allowed to access an object. When a subject attempts to access an object, the sensitivity of the subject is compared with the sensitivity of the object. If the subject has higher or same sensitivity as the object, the access is allowed, otherwise, the access is denied. One of the most known implementation of the MAC model in databases is row-level security.

**(Ausriss aus der Best Practice 3.5.3: *Authorization* [Angy09])**

Zum Least-Privilege-Prinzip wird des Weiteren ausgeführt:

> **The least privilege principle**
>
> Implementing the principle of least privilege on a database gives every subject of this database the bare minimum of privileges required to do his job. By assigning to a subject exactly the privileges he needs to fulfill his duties and not more, one ensures it cannot make unauthorized actions on the database, while still being able to perform its daily operations. The task of determining what access roles and subjects really need falls within the responsibility of the information owner; or should be accomplished in accordance with his decisions; for example by the information security administrator. It can be based amongst others on the data classification established in (...) (especially for the privileges on database objects).

**(Ausriss aus der Best Practice 3.5.3: *Authorization* [Angy09])**

Der Einsatz von zusätzlichen Zugriffskontrollen auf der Ebene von Tabellen kann den Schutz von sensitiven Daten erheblich verbessern. Dies eröffnet die Möglichkeit, eine Tabelle mit unterschiedlichen Festlegungen des Schutzbedarfs der Spalten zu verwenden, ohne dass die Sicherheit vernachlässigt wird. Zwei verschiedene Ansätze die dabei zur Sicherheit vor unberechtigtem Zugriff beitragen können, sind *Views* und *Row-Level Security*, die in der Best Practice wie folgt erläutert werden.

Beispiel *View*:

> A view is a database object that allows, amongst others, to restrict to a subject the data available on a table. It is a virtual table formed with a static query on the original table; the query adapts the table to what the subject initiating the request is allowed to see. The subset of rows and columns resulting from the query can be accessed like a normal table and is dynamic, what means that the data contained in a view changes according to the data contained in the original table.
>
> Views may be very helpful to ensure no unauthorized actions happen on a table when subjects which do not have the same permissions on a table need to access it. In such a case, one may create as many views as there are different set of privileges for this table, so that subjects can see and access the data of this table according to their permissions.
>
> The following SQL statement creates a view of the customers table. The resulting view contains the first name, last name and city of customers living in Germany:
>
> CREATE VIEW V_Customers
> SELECT first_name, last_name, city
> FROM customers
> WHERE country = ‘GERMANY’;

**(Ausriss aus der Best Practice 3.5.3: *Authorization* [Angy09])**

Für die Row-Level Security wird hierbei ausgeführt:

> Row-level security is an advanced access control that permits to “filter out” the data of a table so that subjects can access only the data they are allowed to see. As a consequence, the table on which row-level security is implemented may not contain the same information depending on the subject accessing it and the privileges it owns.
>
> Row-level security relies on different mechanisms depending on the DBMS product. It can be based for instance on the concept of security labels (similarly to the MAC model) that establishes the sensitivity of each data and account. These security labels are ordered so that a security label has a lower, higher, or the same sensitivity as another one. When attempting to access a table, a subject can only access the data that has a lower or the same sensitivity as its database account. This approach is present in DB2 UDB for z/OS and some Oracle products. (...)

**(Ausriss aus der Best Practice 3.5.3: *Authorization* [Angy09])**

Die Best Practice geht desweiteren noch auf die Tiefe der Implementierung ein, die zur Umsetzung der Vorgaben erforderlich sind.

> Different levels of implementation
>
> Authorization can be implemented in several ways, depending on the risk associated to the sensitivity of the data, the number of subjects in the DBMS environment, and many more factors. To cover as many different database environments as possible, this best practice offers the following three levels of implementation:
>
> - For the implementation level 1, at least the following shall be implemented:
>
> 1. One shall use either the RBAC or MAC model. The use of DAC is prohibited. Therefore privileges shall not be assigned directly to subjects.
>
> 2. Permissions on the data shall be implemented based on the data classification established in section (...).
>
> 3. Roles privileges and subject roles (for RBAC) and sensitivities of subjects and objects (for MAC) shall be reviewed regularly to ensure that the least privilege principle is respected.
> (…)

**(Ausriss aus der Best Practice 3.5.3: *Authorization* [Angy09])**

Es folgen die Hinweise für weitere Implementierungstiefen. Für Level drei (bei dieser Best Practice höchste Stufe der Implementierungstiefe) wird zusätzlich gefordert:

> For the implementation level 3, the database installation shall fulfil implementation level 2 plus the following:
>
> 1. For database tables containing sensitive data on which subjects have different privileges, one shall apply a table-level security access control such as views or a row-level security solution.

**(Ausriss aus der Best Practice 3.5.3: *Authorization* [Angy09])**

Diese Best Practice ist insgesamt auf elf Seiten englischsprachiger Text ausgeführt. Es wurden nur einzelne Abschnitte zur Illustration wiedergegeben. Aus Platzgründen muss auf eine

vollständige Wiedergabe der Best Practice in diesem Beitrag verzichet werden. Der vollständige Text dieser Best Practice kann bei den Autoren angefordert werden.

# 3 Ausblick

Die Entwicklungen der jüngsten Zeit haben gezeigt, dass die nachweisbare Sicherheit von Datenbankinstallationen ein zentrales Thema für IT-Sicherheitsbeauftragte und Datenschutzbeauftragte von Unternehmen ist. Der Handlungsbedarf in diesem Bereich kann kaum überschätzt werden.

Jedoch ist heute kein allgemein akzeptiertes Verfahren verfügbar, mit dem einem Betreiber einer Datenbankinstallation sowohl Richtlinien für die Absicherung seiner Installation an die Hand gegeben werden können als auch eine Möglichkeit geschaffen wird, die konsequente Umsetzung dieser Richtlinien zu dokumentieren.

Die in diesem Beitrag vorgestellten Best Practices sind ein erster Schritt zur Entwicklung eines solchen Schemas. Sie können dem Administrator einer Datenbankinstallation als roter Faden zur Umsetzung von Sicherheitsmechanismen dienen. Im Weiteren können auf Basis dieser Best Practices konkrete Anforderungen an ihre Umsetzung definiert werden.

Mit Hilfe solcher konkreter Anforderungen ist dann auch eine unabhängige Prüfung (und Zertifizierung) der Umsetzung möglich. Durch eine solche unabhängige Prüfung wäre es dann einem Administrator einer Datenbankinstallation nicht nur möglich, die durchgehende und konstante Umsetzung eines Sicherheitskonzepts sicherzustellen; der definierte und sichere Umgang mit Daten könnte nach einer erfolgreichen Prüfung auch dritten Parteien gegenüber nachvollziehbar demonstriert werden.

## Literatur und Quellen

[Angy09] Olivier Angyal: Database Security: Towards the Development of a Certification Scheme for Database Installations, Diploma Thesis at Fachhochschule Münster, January 2009.

[ArVi08] Afonso Araujo Neto, Marco Vieira: Towards Assessing the Security of DBMS Configurations. International Conference on Dependable Systems & Networks: Anchorage, Alaska, 2008.

[CCRA08] Common Criteria Recognition Agreement: Certified product list – databases. http://www.commoncriteriaportal.org/products_DB.html#DB
(Stand: Dez. 2008).

[CFMS94] Silvana Castano, Mariagrazia Fugini, Giancarlo Martella, Pierangela Samarati: Database Security (Acm Press Books), ISBN 978-0201593754, 1994.

[DISA07] Defense Information Systems Agency: Database Security Technical Implementation Guide, Version 8, Release 1, 2007.

[DoD85] Department of Defense: Trusted Computer System Evaluation Criteria, 1985.

[GeJa08] Michael Gertz, Sushil Jajodia (Herausgeber): Handbook of Database Security Applications and Trends, ISBN: 978-0387485324, 2008.

[Nata05] Ron Ben Natan: Implementing Database Security and Auditing, ISBN: 978-1555583347, 2005.

[SaSc75] Saltzer, Schroeder: The Protection of Information in Computer Systems, April 1975.