Title: *Database Management Systems Essential Security Requirements*
Maintained by: *Database Management System Working Group (DBMS WG)*
Version: *1.1*
Date of issue: *September 28, 2016*

Status

The Database Management Systems Technical Community (DBMS TC) following the certification of the Base Protection Profile for Database Management Systems (DBMS PP) version 2.07 has decided to develop an Essential Security Requirements (ESR) that specifies the essential requirements for Database Management Systems. This initial draft contains material that was provided by the DBMS TC, for a cPP.

Background and Purpose

This document describes the high-level set of security requirements that a Database Management System (hereafter 'DBMS') will satisfy when evaluated against the cPP written for such technology.

A DBMS is a software system that lets one or more computer users create and manage access to data in a database. The DBMS manages incoming data, organizes it, and provides ways for the data to be modified or extracted by users or other programs.

Formally, "database" refers to the data itself and supporting data structures. Because they are so closely related, the term "database" when used casually often refers to both a DBMS and the data it modifies. The supporting data structures allow for efficient modification and retrieval of the data in the database (e.g. an index to speed up data retrieval) as well as to establish and maintain logical relationships between data objects (e.g. the set of columns that belong to a table). The DBMS manages requests so that users and other programs are free from having to understand where the data is physically located on storage media and, when allowing concurrent requests, who else may be accessing the data at the same time. Establishing, protecting, and maintaining the correctness of these structures (aka "metadata") is a key responsibility of the DBMS.

Databases and DBMSs can be categorized according to the database model(s) that they support (such as relational or object-oriented), the type(s) of system they run on (from a server cluster to a personal device), the query language(s) used to access the database (such as SQL or XQuery), and their internal engineering, which affects performance, scalability, resilience, and security. General-purpose DBMSs aim to meet the needs of as many applications as possible, which contribute significantly to their complexity and functional richness. The most typical commercial DBMS products are general purpose relational database management systems (RDBMS), whose user and program interface is the Structured Query Language (SQL).

Physically, database servers are usually dedicated systems that hold the actual databases and run only the DBMS and related software. Database servers are often multiprocessor computers, with generous memory and RAID disk arrays used for stable storage. Hardware database accelerators, connected to one or more servers via a high-speed channel, are also used in large volume transaction processing environments. DBMSs are found at the heart of most applications that deal with data and relations between the data.

Use Case(s)

DBMS are used where the storage, modification, and retrieval of large amounts of data is required. Very different use cases are conceivable for DBMS. These can be described for example by the following categories:

- Number of users (ranging from one user for a private database to a large number of users for a database in a big company)
- Complexity in terms of user groups (one or more user groups with different access rights)
- Access (configured for local access only (e.g. for use by the in-house finance group) or made available for access by users physically distributed around the world)
- Infrastructure (different parts of an enterprise may share one instance of a DBMS, or have their own dedicated DBMS and underlying physical server machines and disk storage, or they use a shared disk architecture, where each part has its own DBMS, but all DBMS share the other storage)

The evaluation activities (EA) in the supporting document (SD) are expected to cover each use case, starting from a very simple DBMS and ending with a very complex DBMS.

Rather than simply providing a store/retrieve capability as would a file system, a DBMS provides for semantically rich, value-based modification and querying of data with the ability to define and apply data consistency logic to automatically enforce user-specified business rules.

The basic interactions supported by most existing DBMS fall into four main groups:

1. **Data definition**. Defining new data structures for a database, removing data structures from the database, modifying the structure of existing data.
2. ***Update****. Inserting, modifying, and deleting data.*
3. ***Retrieval****. Obtaining information either for end-user queries and reports or for processing by applications.*
4. ***Administration****. Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information if the system fails.*

Personnel interacting with a DBMS and the databases it manages typically fall into one or more of the following roles. Not all roles necessarily exist, or are filled by distinct people in a given installation, depending on the size, complexity, and criticality of the DBMS they serve.

- ***Database Administrators*** *(short form "DBA"): A DBA is responsible for the installation, configuration, upgrade, administration, monitoring and maintenance of databases in an organization. They may also plan, co-ordinate and implement security measures to safeguard the database.*
- ***Application Programmers****: These personnel write application programs to interact with the database. Application programs can be written in some programming language such a COBOL, PL/I, C++, JAVA or some higher level fourth generation language. Such programs access the database by issuing the appropriate request, which would take the form of a query statement for an RDBMS.*

- ***End Users**: End users are the personnel who use the applications developed by application programmers. End users need not know about the inner workings of the application, database, or DBMS. Their focus in on using the application to accomplish its designed business task.*
- ***System Analyst**: A system Analyst determines the requirements of end users, and determines how to achieve these using the capabilities of the DBMS. System Analysts play a major role in database design, developing detailed technical specifications and accounting for such considerations as feasibility, cost, performance, and capacity.*

Resources to be protected

- The data stored in databases, against unauthorized access, tampering, and destruction.
- The metadata describing objects in databases and their relationships, again against unauthorized access, tampering, and destruction.
- User identities and their authentication credentials and other security attributes as stored and/or used by the DBMS, against unauthorized access or tampering.
- Audit logs and any other activity tracking and monitoring mechanisms, against tampering and destruction, if audit storage and review is part of the security functionality (see chapter Optional Extensions).
- Compute resources, ensuring they are neither consumed to excess or to the point of denying availability to legitimate users.
- Configuration data against tampering via the TOE interfaces.

In summary, the DBMS is responsible for protecting the resources under its control from disclosure, tampering, and destruction, and to ensure the ongoing availability of these resources to legitimate users.

Attacker access

- Design flaws and programming bugs in the DBMS and the associated programs and systems, creating various security vulnerabilities (e.g. weak or ineffective access controls) which can lead to data loss/corruption, performance degradation etc.
- Unauthorized or unintended activity or misuse by authorized database users, database administrators, or network/systems managers, or by unauthorized users or hackers (e.g. inappropriate access to sensitive data, metadata or functions within databases, or inappropriate changes to the database programs, structures or security configurations)
- Malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services
- Data corruption and/or loss caused by the entry of invalid data or commands, mistakes in database or system administration processes, sabotage/criminal damage etc.

Attacker Resources

For the development of the cPP it is assumed that the value of the data to be stored in databases we consider is not so high, so that the motivation or resources of attackers are assumed to be limited in terms of amount of time, available software, expertise, knowledge and equipment.

Boundary of Device

A DBMS is a software system running within an environment consisting of - at a minimum - compute, storage, and networking capabilities. The security boundary of the DBMS includes all DBMS software and the functions it performs, and the DBMS' correct and safe use of resources and capabilities within the environment, such as (for example)

- Protecting stored data using the access control mechanisms as provided by the storage subsystem
- Protecting stored data from disclosure by using encryption or equivalent mechanism
- Protecting data in transmission by configuring network channels using the appropriate protocols and channel protection mechanisms

Essential Security Requirements

The DBMS will provide the following security capabilities:

- Identification and Authentication
    - Confirm the identity of any user which attempts to use the DBMS, and
    - Verify that the user has been authorized to access resources under control of the DBMS.

  These actions may be accomplished by the DBMS acting as the authenticator based on the user information it manages, or by making use of an authenticator in the operating system or other mechanism available in the environment.
- Discretionary Access Control (DAC)
    - Control who is allowed to access what information in a database.
    - Control access to objects using access control rules based on the identity of the users or groups requesting access, attributes of the users and groups and potentially other input (e. g. time of day).

  The rules determine what specific permissions or privileges have been granted to the requesting users or groups. The information may comprise specific database objects (e.g., record types, specific records, data structures), certain computations over certain objects (e.g., query types, or specific queries), or utilizing specific access paths to the former (e.g., using specific indexes or other data structures to access information).
- Audit Capture
    - Create information on all auditable events.
    - Create a record of events that are potentially security relevant.
- Authorized administration role
    - Allow authorized administrators to configure the policies for discretionary access control, identification and authentication, and auditing.
- Replication
    - Ensuring consistency of data when replicated between different parts of the DBMS.

Assumptions (*This is an optional section)*

- The DBMS is protected from physical attacks by the wider environment.
- There are no general purpose computing or storage repository capabilities (e.g., compilers or user applications) available on the DBMS server, other than those necessary for the operation, administration and support of the DBMS.
- All remote trusted IT systems trusted by the DBMS to provide data or services to or to support the DBMS in the enforcement of security policy decisions are assumed to correctly implement the functionality used by the DBMS.
- Any information provided by a trusted entity in the IT environment and used to support the provision of time and date, information used in audit capture, user authentication, and authorization that is used by the DBMS is correct and up to date.
- The TOE security functionality is managed by one or more competent and trusted administrators.
- All connections to and from remote trusted IT systems and between separate parts of the TSF are physically or logically protected within the TOE environment.

Optional Extensions

Requirements captured in this section may already be realized in some products in this technology class, but this ESR is not mandating these capabilities exist in "baseline" level products.

- Audit storage and review functionality

Objective Requirements

Requirements specified here specify security-relevant behaviour that is not expected to be realized currently in products, but capabilities that may be mandated in future versions of the ESR and resulting cPPs. Currently there are no extensions planned.

Outside the Scope of Evaluation

The external IT entities, with which the DBMS may interact, if they are outside the TOE, include the following:

- Client applications that allow users to interface with the DBMS server.
- The host operating system (host OS) on which the TOE has been installed.
- The networking, printing, data-storage, and other devices and services with which the host OS may interact on behalf of the DBMS or the DBMS user; and the other IT products such as application servers, web servers, authentication servers, directory services, audit servers, and transaction processors with which the DBMS may interact to perform a DBMS function or a security function.

The DBMS must specify the host OS on which it must reside to provide the desired degree of security feature integration as well as the configuration of those operating systems required to support the DBMS functions. However, the goals of confidentiality, integrity, and availability for the TOE must be met by the total package: the DBMS and the external IT entities with which it interacts. In all cases, the TOE must be installed and administered in accordance with the TOE installation and administration instructions.

All cryptography used to protectively store or transmit data is performed in the environment and not is part of the DBMS evaluation.