# Chapter 6
# Security Gaps in Databases: A Comparison of Alternative Software Products for Web Applications Support

**Afonso Araújo Neto**
*University of Coimbra, Portugal*

**Marco Vieira**
*University of Coimbra, Portugal*

## ABSTRACT

*When deploying database-centric web applications, administrators should pay special attention to database security requirements. Acknowledging this, Database Management Systems (DBMS) implement several security mechanisms that help Database Administrators (DBAs) making their installations secure. However, different software products offer different sets of mechanisms, making the task of selecting the adequate package for a given installation quite hard. This paper proposes a methodology for detecting database security gaps. This methodology is based on a comprehensive list of security mechanisms (derived from widely accepted security best practices), which was used to perform a gap analysis of the security features of seven software packages composed by widely used products, including four DBMS engines and two Operating Systems (OS). The goal is to understand how much each software package helps developers and administrators to actually accomplish the security tasks that are expected from them. Results show that while there is a common set of security mechanisms that is implemented by most packages, there is another set of security tasks that have no support at all in any of the packages.*

## INTRODUCTION

This paper studies the security features offered by different software packages in the context of database-centric web applications, focusing particularly on the mechanisms provided by Operating Systems (OS) and Database Management Systems (DBMS), the two main software elements that can be found in any database server. The study focuses in seven largely used software packages, composed of four different DBMS engines (Oracle 10g, SQL Server 2005, PostgreSQL 8, and MySQL

DOI: 10.4018/978-1-4666-2482-5.ch006

Community Edition 5) and two different operating systems (Windows XP and Red Hat Enterprise Linux 5). The main goal of the work is to understand how much support these products provide for fulfilling well-known security best practices that administrators are expected to follow. Even though there are not the latest versions, these products are representative of the state of the art technology currently available for provisioning database installations and are frequently used to support the deployment of business critical web applications.

The study consists of comparing the characteristics of the software packages against a comprehensive list of security concerns, which should be taken into account when deploying a database-centric web application. This list was created based on an extensive field research where we were able to identify the most important security practices for database systems. In our methodology, each security best practice was mapped onto a desirable system state - called a *System State Goal* - that represents the state of the system when the practice is being correctly applied. That system goal was used to extrapolate and identify the functionalities and mechanisms needed to implement the practice. The final list of security mechanism was used to build a gap analysis table that maps the security features actually provided by the software packages assessed with the mechanisms that are needed to fulfill the security best practices initially identified.

This work is motivated by the central role played by databases in the security of today's web applications. In fact, it is well known that security aspects must be an everyday concern of a system administrator, in particular of Database Administrators (DBA). When installing a database server to support a web application, the DBA must consider several complex requirements, including performance, recovery time and security, following a defense-in-depth approach (Howard & Leblanc, 2002). These requirements have strong implications in the hardware and software to be used in the database infrastructure. The problem of performance and recovery in databases has been largely studied in the past (Vieira & Madeira, 2002; Kanoun & Spainhower, 2008; TPC, 2011). However, assessing and comparing the security features provided by software packages is still an open issue.

Web application security (and, in particular, database security) arises from the need to protect from unauthorized attempts to access private data and loss or corruption of critical data due to malicious actions (Bertino et al., 1995; Castano et al., 1994, Pernul & Luef, 1992; Schell & Heckman, 1987). Other concerns include protecting against malicious interferences that may cause undue delays in accessing or using data, or even denial of service.

The problem is that different software products offer different sets of security mechanisms that target diverse security concerns. This, adding to the fact that several software systems are needed to install a database server (the minimum set is composed by an OS and a DBMS), makes it very difficult to map what is nowadays supported by software with the actual database administrators security needs. Our study shows that different software packages for databases allow implementing different security concerns and that there are security concerns for which few (or none) software products provide easy to use mechanisms. Furthermore, there is a small set of security mechanisms that is available in all the packages considered.

Several security evaluation methods have been proposed in the past (Cachin et al., 2000; Commission of the European Communities, 1993; Common Criteria, 1999; Department of Defense, 1985; Sandia National Laboratories, 2011; Vieira & Madeira, 2005). However, to the best of our knowledge, none of the existing methods provide a comprehensive comparison of widely used software products for database installations, or even a methodology to fairly achieve one. Furthermore, practical experience shows that these methods are very complex and hardly could be used by sys-