Advances in Science and Technology
Research Journal
Volume 10, No. 30, June 2016, pages 69–75
DOI: 10.12913/22998624/62514

Research Article

IMPROVING DATA ACCESS SECURITY BY SERVER-SIDE FUNCTIONAL EXTENSIONS

Marek Miłosz[1], Dariusz Draganek[1]

[1] Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36b, 20-618 Lublin, Poland, e-mail: m.milosz@pollub.pl, d.draganek@pollub.edu.pl

Received: 2016.03.01
Accepted: 2016.04.25
Published: 2016.06.01

ABSTRACT
All Database Management Systems used in the industry provide secure access to data at the server level. The level of security is influenced by technology, security model, password encryption method, password strength and others. The human factor – unreasonable behaviour of users – also has a significant impact on safety. Developers of database applications often implement their security policy by limiting the risk caused by users. This implementation has a disadvantage – it does not work outside of the application. A large number and variety of applications that work with the database server may then create a security gap. The paper presents the authors' extensions of the functional features implemented in the Oracle database server, increasing the security of data access. The implemented methods of controlling the time of user access to data and limiting the use of serial and modular passwords are also discussed.

Keywords: security, data access, access time control, password policy.

INTRODUCTION

Security is one of the key elements determining the proper operation of database (DB) industrial information systems. The problem of its enhancement became more significant with increasing risks resulting from the spread of information technology (enlargement of people's knowledge and skills), development of the Internet (increased access to information for everyone) and the opening of the industry to a broader environment (development of remote access to enterprise information systems).

Industrial information security problems in multi-tier systems can be resolved at different levels, for example application, network (e.g. Directory Services, DS), application server, or database. Some methods can only be carried out in a particular tier (e.g. [7]). Oracle Database Management System (DBMS) supports the following methods of user authentication [8]:

- Password – through the ORACLE DB server.
- External – through an operating system, directory services (usually Microsoft Active Directory or analogical).
- Global – by an external authentication service (e.g. Oracle Internet Directory).

External and Global methods, using central authentication and user roles management, enable the realisation of the policy of "Single Sign-On." For this purpose the Windows Active Directory with Windows Native Authentication can be used, or Kerberos - in heterogeneous environments. However, in these cases the installation and maintenance services in the system are required. Very different methods of implementation of an authentication policy are available in Active Directory.

Industrial information systems are characterised by a large number of various applications, including legacy applications [9]. Very often they do not use the Directory Services, so the only common point is the database.

One of the many elements of information system security is the limiting of access to data. The need for restrictions is caused by legal regulations (e.g. in the area of access to personal data) or the well-understood interests of companies.

Industrial information systems implement different security models. The most important are [1, 2, 10]:

- Discretionary Access Control (DAC), in which the owner of the resource determines its accessibility and the rights of other users to it.
- Mandatory Access Control (MAC), which implies the existence of pre-defined access rules classifying the data due to their level of confidentiality, and the users by assigning them a confidence class, which automatically regulates access permissions to all resources.
- Role-Based Access Control (RBAC), in which the roles assigned to the user (or vice versa) are used for the management of access rules. Roles are created and assigned manually by a central administrator.

Modern Database Management Systems (DBMS), which are an integral part of industrial systems, usually implement hybrid models that combine the features of a number of security models. This also applies to the Oracle DBMS [20].

Unfortunately, even if DB industrial system security models perform correctly, after the collection and consolidation of data in a DBMS a problem emerges. The data in fact lose contact with the application – they live outside it. This may pose a significant threat to their safety. The situation becomes even more complicated in a multiple application environment, which DB systems are. In such environments, the user can perform many roles with different access rules to the data. Simultaneous use of different roles can lead to violations of data security. An important limitation of this risk is transferring the rules of secure data access to the database server level [22].

The safety level of a particular DBMS depends on the supplier, and the developer of industrial systems, unfortunately, does not have any effect on it (except the choice of one or another DBMS). Security levels are determined by standards. One of them is an ISO/IEC 15408 [3], which defines seven levels of security: EAL1... EAL7 (EAL – Evaluation Assurance Level). Industrial DBMSs obtain a maximum level of EAL4. DBMSs at a very high level of security include, among others, servers [5]: IBM DB2 v. 10.1, MS SQL Server 2012 EE x64, Oracle Database 11g, Teradata Database 13.0, Sybase Adaptive Server Enterprise 15.0.1.

METHODS OF ACCESS CONTROL IN ORACLE DBMS

The current version of the Oracle DBMS provides EAL4 level of data security. It is ensured through the implementation of two tasks: protection of data and protection of the system. The term protection of the system is meant as: control of access to the DBMS (access authorisation), control of resources allocated to the user (e.g. disk space or CPU time) and monitoring their actions in terms of resource consumption. The concept of data protection is related to defining the access rights to the objects in the database, authorised operations on these objects and monitoring performed operations.

Access control on the server level in the Oracle DBMS is implemented by the following items:

- user account,
- profile,
- privilege,
- role.

The user account is classically characterised by a name and a password. Other parameters of the account are the default and temporary table space, profile, set of roles and permissions.

During the Oracle DBMS installation, an account is created for an administrator, who disposes and manually assigns permission (RBAC model). In addition to the user account, access control is provided by profiles limiting the use of system resources by the user.

Privileges are divided into system ones (defining user rights to perform the operation) and object ones (defining database objects on which operations can be performed). Under certain circumstances, one user may transfer his or her own privileges to other users (DAC model).

Roles are used to granting or revoking an access to groups of users (or other roles). Therefore they simplify permissions management, enable the selective and dynamic usage and enhance system performance.

The MAC model is implemented in the DBMS Oracle in a special version of the Trusted Oracle, and in cooperation with the operating system, which has the appropriate level of security. In this process a method of labelling information is used which assigns a label to each object in the database. Only users who have the assigned label (or a dominant label) may have access to such an object.

An important element of the Oracle system security (starting from version 11) is the possibility of distinguishing uppercase and lowercase letters in passwords, and generating a hash of concatenation of a user's name and password (for the purpose of storage in the database) using algorithm SHA-1 with salt [8, 11]. This algorithm is much more secure than the previously used DES/MD5. Unfortunately, to encrypt historical passwords stored in the database (e.g. for control of duplications) the old SHA-1 algorithm is used. In certain circumstances this may constitute a serious hazard of guessing a user's password.

THE HUMAN FACTOR IN SECURITY POLICY

Security policy is created in three dimensions [14]: technical, human and organisational. The human factor in data security is very important [13], even if technical and organisational security is perfect. There are some aspects of the human factor which expose weaknesses in the security system and should be carefully considered, for example user authentication [12, 17].

One essential element of security policy strongly dependent on the human factor is the password policy. Rules applied in the security policy may be as follows [4, 14, 15, 18, 23]:

- forcing a change of password after a certain period (e.g. monthly) or user activity (e.g. every 20th logging into the system);
- determining the minimum number of characters allowed in a password;
- forcing the use of an appropriately broad set of characters – uppercase and lowercase letters, numbers, special characters;
- blocking easy passwords relating to the application and the user's account (the same or similar password to the name of the user, system, server or database, etc.) and the names of months, seasons, colours, etc.;
- blocking the possibility of using periodically repeating (looping) passwords for implementation of security policies, enforcing periodical change of password to another (lock level: N previous passwords, N = 1, 2, 3);
- forcing differences in passwords at the appropriate level, e.g. at least 4 characters of the password must be different compared to the previously used password;
- exclusion of a frequently used, shape-based, switch of letters to numbers: I-1, Z-2, E-3, A-4, S-5, G-6, T-7, B-8, Q-9, O-0;
- limiting the use of serial passwords (so called serialisation of passwords) – blocking the possibility of setting passwords too similar to those used in the past by, for example, [19]:
  - changing the numbers at the end, e.g. password1, password2, password3, ...;
  - converting lowercase to uppercase letters and vice versa, e.g. pass1, Pass1, paSS1, etc.;
  - using modular passwords, which join a number of consecutive characters in a variety of configurations, e.g. marta12jan, jan34marta, martajan4b, etc.

Unfortunately, the application of these rules (generally correct in their assumption) induces objective problems (such as: forgetting passwords and the need for their recovery, writing down passwords on removable media and storing them in places easily accessible to unauthorised persons, reducing the list of potential passwords and the complexity of the process of their creation) and a negative reaction of the user, who bypasses inconvenient security rules and regulations (by for example saving passwords on pieces of paper, using serialised or culture-related passwords). At the same time, research shows that the behaviour of different groups of users is different [6].

This shows that the weakest element in the process of password management will always be its user. Even the best-designed and implemented security can become insufficient with a wrong approach to the access to information systems. Therefore, it is necessary to systematically educate users and remind them of their responsibility for the security of the system for each user, regardless of privilege level [12, 14].

The Oracle DBMS and it security policy parameters

In the standard versions of the Oracle DBMS password policy is defined in a user profile with a number of parameters – Table 1 [11].

Parameters `PASSWORD_REUSE_MAX` and `PASSWORD_REUSE_TIME` (Table 1) require saving the hash function for previously used passwords. In Oracle, the MD5 algorithm and `sys.user_history$` system table are used for each user whose profile has the value of `PASSWORD_REUSE_MAX` and `PASSWORD_REUSE_TIME` other than the default. The `sys.user_history$` table

Table 1. Parameters defining the password policy for the user profile of the Oracle DBMS

Parameters	Meaning
FAILED_LOGIN_ ATTEMPTS	maximum number of times an incorrect password may be entered after which an account will be blocked
PASSWORD_LIFE_ TIME	maximum password lifetime
PASSWORD_REUSE_ TIME	time, counted in days, after which a password can be reused
PASSWORD_REUSE_ MAX	minimum number of password changes, after which the currently used one can be reused
PASSWORD_LOCK_ TIME	specifies the number of days the account will be locked after exceeding FAILED_ LOGIN_ATTEMPTS
PASSWORD_GRACE_ TIME	number of days in which to change a password, counted from the first login after exceeding PASSWORD_LIFE_TIME
PASSWORD_VERIFY_FUNCTION	name of the function verifying the correctness of the password

has three columns: USER# (a unique user code in the database), PASSWORD (hash of password) and PASSWORD_DATE (password expiration date).

The existence in the profile of the PASSWORD_VERIFY_FUNCTION parameter allows to create more complex, personalised password security policies.

The authors' extensions to increasing data access security

Increasing security of access to data in the Oracle DBMS is possible through the development of additional server-side functionality. Authors' extensions perform the Time Control of User Access to Data (TCAD) and Limiting the Use of Unacceptable Passwords (LUUP). They are targeted at industrial applications.

MODULE OF THE TIME CONTROL OF USER ACCESS TO DATA

The management of the time of user access to the data means defining the intervals in which the user cannot log in to the DB server. Typically, the functionality is realised on the application side, connected to the DB. The Oracle DBMS does not have this functionality built into the server, so it is necessary to develop it on the server side.

The TCAD tool consists of a web application for defining intervals of access/access block and the extended DB trigger function that prevents users from logging into the DB except at the defined intervals. This tool uses the AFTER LOGIN trigger and a table consisting of defined intervals of access blocks.

A specially developed Access Control Definition (ACD) application (see Fig. 1) allows to define intervals of logging restriction for each user. Due to the industrial use, the schedule of access blocks is defined to the minute (server time): during the particular period (i.e. from the date of the day), for the particular days of the week (e.g. for all weekends) and selectively for specific days. In the ACD application, the adoption of group defining limits for the days of the week results from a typical cyclic scheduling of workers (i.e. users of the DB) in the industry.

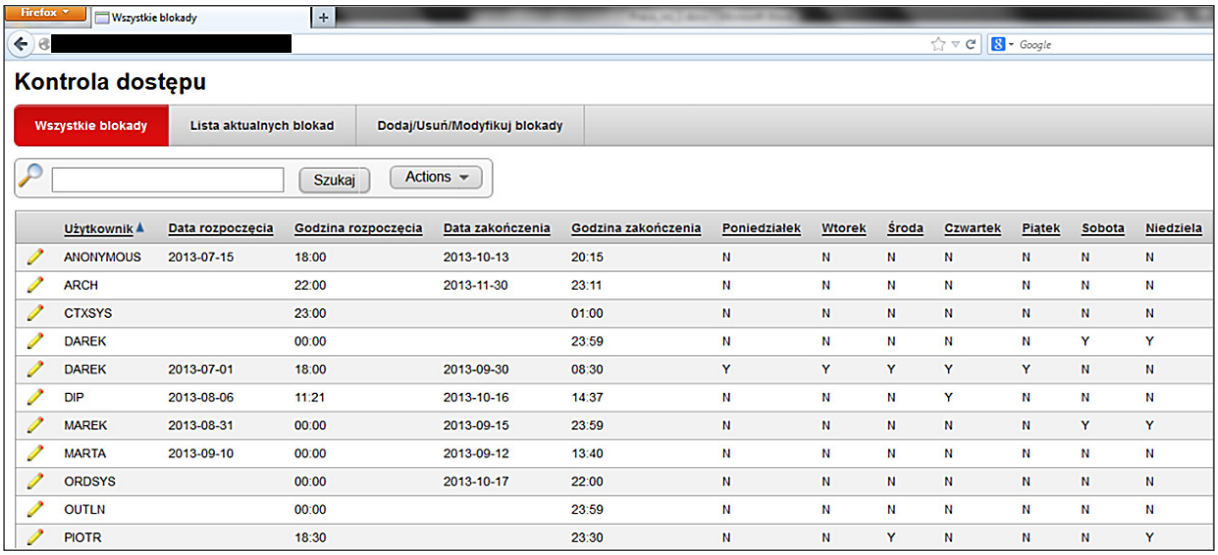
The ACD application, beyond the ability to edit blocks of user access, has expanded reporting and filtering options.

The restriction of access to the data corresponds to the trigger which on the basis of entries blocks the access. The trigger is activated by the AFTER LOGIN event. In the case of blockade detection (based on the contents of a table filled with ACD applications) the trigger performs the DISCONNECT IMMEDIATE operation.

MODULE OF LIMITING THE USE OF UNACCEPTABLE PASSWORDS

The LUUP tool is a function stored in the Oracle DB (and executed by the server), which after its indication in the profile of a DB user (parameter PASSWORD_VERIFY_FUNCTION) replaces the standard Oracle solution (function VERIFY_FUNCTION_11G) with the one developed buy authors. The standard function VERIFY_FUNCTION_11G performs a very simple password security policy. This policy is in many cases insufficient.

The Oracle function VERIFY_ FUNCTION_11G controls the following situations [10]:



Użytkownik	Data rozpoczęcia	Godzina rozpoczęcia	Data zakończenia	Godzina zakończenia	Poniedziałek	Wtorek	Środa	Czwartek	Piątek	Sobota	Niedziela
ANONYMOUS	2013-07-15	18:00	2013-10-13	20:15	N	N	N	N	N	N	N
ARCH		22:00	2013-11-30	23:11	N	N	N	N	N	N	N
CTXSYS		23:00		01:00	N	N	N	N	N	N	N
DAREK		00:00		23:59	N	N	N	N	N	Y	Y
DAREK	2013-07-01	18:00	2013-09-30	08:30	Y	Y	Y	Y	Y	N	N
DIP	2013-08-06	11:21	2013-10-16	14:37	N	N	N	Y	N	N	N
MAREK	2013-08-31	00:00	2013-09-15	23:59	N	N	N	N	N	Y	Y
MARTA	2013-09-10	00:00	2013-09-12	13:40	N	N	N	N	N	N	N
ORDSYS		00:00	2013-10-17	22:00	N	N	N	N	N	N	N
OUTLN		00:00		23:59	N	N	N	N	N	N	N
PIOTR		18:30		23:30	N	N	Y	N	N	N	Y

Fig. 1. Interface of the Access Control Definition application

O1. Is the minimum password length greater than 8 characters?
O2. Is the new password not the same as the username or username + a number between 1 and 100?
O3. Is the password not the username written backwards?
O4. Is the password different from the database name or database name plus a number between 1 and 100?
O5. Is the password one of the strings: 'welcome1', 'database1', 'account1', 'user1234', 'password1', 'oracle123', 'computer1', 'abcdefg1', 'change_on_install'?
O6. Is the password 'oracle + number in the range 1-100'?
O7. Does the password contain at least one letter and one number?
O8. Is the password different from the previous one by at least three characters?

In the case of even one negative response, the `VERIFY_FUNCTION_11G` function returns a value, indicating an incorrect password.

The authors' extensions modify the performance of a function defined in `PASSWORD_VERIFY_ FUNCTION` into the function verifying:

- the existing provisions of the standard function `VERIFY_FUNCTION_11G`;
- additional conditions.

The additional conditions used in the authors' LUUP tool are:

A1. Does the password contain at least one lowercase letter, one uppercase letter, one number and one special character?
A2. Does the password not contain the same N element string as the previous password?
A3. Is the password not similar to the n previous passwords: by changing the uppercase and lowercase letters and/or by changing the numbers?

In addition, the dictionary of verified words from condition O5 was expanded into a personalised one, developed on the basis of the most commonly used passwords. This dictionary must be adapted to the specific language of the user (in this case Polish).

Condition A1 forces the user to use a more diverse set of characters and is "patching a hole" in the DBMS Oracle 11g (only: "one letter and one number" – condition O7). Condition A2 controls the use of modular passwords, i.e. permutations of the same strings. Control of the use of passwords similar to those already used is implemented by condition A3.

SECURED PROFILE

The authors' TCAD and LUUP tools will work only for those users that have a profile using it assigned for them. To facilitate its application, a model profile called RESTRICTED has been created, containing the following settings of a password's grace period:

```
CREATE PROFILE RESTRICTED LIMIT
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LIFE_TIME 30
PASSWORD_REUSE_TIME 180
```

```
PASSWORD_REUSE_MAX 7
PASSWORD_LOCK_TIME UNLIMITED
PASSWORD_GRACE_TIME 7
PASSWORD_VERIFY_FUNCTION
FUNCTION_RESTRICTED_DD;
```

The proposed password policy settings in the profile `RESTRICTED` introduce the following limitations:

- 3 consecutive failed login attempts block the account;
- password expiration: 30 days;
- the time the password will be remembered for: 180 days;
- minimum number of passwords, after which the password can be used again: 7;
- the time for which a password is blocked after exceeding the allowed number of logins: permanently;
- the time in which the password can be changed after its expiration: 7 days;
- the function verifying a password change: `FUNCTION_RESTRICTED_DD` (name of the LUUP tool function).

Each user who has this profile set up will be subject to all the restrictions. The parameters that control the password expiration period and history: `PASSWORD_LIFE_TIME`, `PASSWORD_REUSE_ MAX` and `PASSWORD_REUSE_TIME` are asymmetrically selected in order to force the user to use more than the average number of passwords.

CONCLUSIONS AND FUTURE WORK

The Oracle DBMS, in addition to the standard built-in security of access control processes, offers opportunities for developing these procedures. It is the chance to create a personal software and policy of protecting access to industrial systems data.

The developed TCAD and LUUP tools are examples of actions that enhance the security of data access in industrial applications, especially in multiple application systems.

An unresolved problem in the TCAD tool is treating the session of users logged into the database before activating the block. It cannot under any circumstances be assumed that at the time of activating the block the session is automatically closed, because it might lead to loss of user performance, and in addition it is difficult to estimate the time of rolling back the changes. The option of session closing after confirming the last transaction has been used. The TCAD application controls only the newly opened access to the DB. This tool seems to be better, because it limits the possibility of further work, but does not interfere with the implementation of a long operation.

The TCAD tool does not implement the following possible features:

- monitoring the correct (in the sense of: user name + password) login attempts, but in the unauthorised time the access blocks that occur are only mapped in the system tables and are not supported by the author's extensions tool;
- notifying the administrator about blocked access attempts in real-time (by SMS).

These features will be the tasks of future work. The LUUP module can be expanded to realise more and more complex algorithms. These algorithms are the result of the actions of the system users. Analysis of these actions provides information for detecting new, bad patterns of users' password policy. Spotting these patterns allows to create new algorithms and programs to neutralise violations of security policy.

REFERENCES

1. Ahn G.-J., Sandhu R., Role-based authorization constraints specification. ACM Transactions on Information & System Security, 3 (4), 2000, 207–226.
2. Bouzida Y., Logrippo L., Mankovski S., Concrete- and abstract-based access control. International Journal of Information Security, 10 (4), 2011, 223–238.
3. CC. Common Criteria, 1/25/2016, http://www.commoncriteriaportal.org.
4. Chiasson S., van Oorschot P.C., Quantifying the security advantage of password expiration policies. Designs Codes and Cryptography, 77, 2015, 401–408.
5. CP. Certified Products. Common Criteria, (1/25/2016), http://www.commoncriteriaportal.org/products/.
6. Duggan G.B., Johnson H., Grawemeyer B., Rational security: Modelling everyday password use. International Journal of Human Computer Studies, 70 (6), 2012, 415–431.
7. El Menshawy D., Mokhtar H., Hegazy O., A Keystroke Dynamics Based Approach for Continuous Authentication. In: Kozielski, A., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., Kostrzewa, D. (eds) Beyond Databases, Architectures, and Structures, CCIS, 424, 2014, 415–424.

8. Fogel S., Oracle Database Administrator's Guide. 11g Release 2 (11.2). Oracle Corp., 2013.
9. Gruner F., Kassel S., Extending Lifecycle of Legacy Systems – An Approach for SME to Enhance Their Supported Business Processes through a Service-Integration-System. International Federation For Information Processing –Publications, 372, 2012, 43–50.
10. Hasani S. M., Modiri N., Criteria Specifications for the Comparison and Evaluation of Access Control Models. International Journal of Computer Network & Information Security, 5 (5), 2013, 19–29.
11. Huey P., Oracle Database Security Guide. 11g Release 2 (11.2). Oracle Corp., 2012.
12. Juszczyk M., Digital identity acceptance at Polish large enterprises: The survey results. Actual Problems of Economics, 132, 2012, 474–481.
13. Juszczyk M., Impact of human factor in data security. Actual Problems of Economics, 120, 2011, 359–364.
14. Kozieł G., Information security policy creating. Actual Problems of Economics, 126, 2011, 367–380.
15. Lichtfield D., Anley C., Heasman J., Grindlay B., The Database Hacker's Handbook: Defending Database Server. Wiley&Sons, 2005.
16. Lorenz B., Kikkas K., Klooster A., The four most-used passwords are love, sex, secret, and god: Password security and training in different user groups. In: Marinos, L., Askoxylakis, I. (eds) First Int. Conference on Human Aspects of Information Security, Privacy, and Trust. LNCS, 8030, 2013, 276–283.
17. Milosz E., Milosz M., Digital Identity Management at Polish SMEs. Actual Problems of Economics, 120, 2011, 340–345.
18. Natan R.B., Implementing Database Security and Auditing. Elsevier Inc., 2005.
19. Neagu A., Oracle 11g Anti-Hackre's Cookbook. PACKT Publishing, 2012.
20. Ni Q., Bertino E., Lobo J., Brodie C., Karat C., Karat J., Trombetta A., Privacy-Aware Role-Based Access Control. ACM Transactions on Information & System Security, 13 (3), 2010, 24–31.
21. Shaul J., Ingram A., Practical Oracle Security: Your Unauthorized Guide to Relational Database Security. Syngress Publishing Inc., 2007.
22. Thion R., Coulondre S., A relational database integrity framework for access control policies. Journal of Intelligent Information Systems, 38 (1), 2012, 131–159.
23. Zezschwitz E., Luca A., Heinrich Hussmann H., Survival of the Shortest: A Retrospective Analysis of Influencing Factors on Password Composition. In: Human-Computer Interaction – INTERACT 2013. LNCS, 8119, 2013, 460–467.