Security requirement

Oracle Database Systems

Deutsche Telekom Group

Version	6.0
Date	Dec 1, 2023
Status	Released

Publication Details

Published by
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Germany

File name	Document number	Document type
	3.29	Security requirement

Version	State	Status
6.0	Dec 1, 2023	Released

Contact	Validity	Released by
Telekom Security psa.telekom.de	Dec 1, 2023 - Nov 30, 2028	Stefan Pütz, Leiter SEC-T-TST

Summary
This security document has been prepared based on the general security policies of the group. Based on the security requirement for “Database systems – General security requirements”, it contains manufacturer-specific security requirements for Oracle database systems, with the objective of creating a uniform security standard.

Table of Contents

1. Introduction

This security document has been prepared based on the general security policies of the group.
The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for units which do not participate in the PSA process. These requirements shall be taken into account from the very beginning, including during the planning and decision-making processes.
When implementing these security requirements, the precedence of national, international and supranational law shall be observed.

1.1. Responsibilities

The system owner must ensure the implementation of the security requirements from this requirements document.

These security requirements are intended for all individuals who are responsible for the development and operation of database systems, and/or who develop database systems (or have them developed) or procure them.

2. General Information

These security requirements contain numerous minor and major enhancements, updates and error corrections compared with the previous version. It is therefore strongly recommended that the updated security requirements are examined in detail.

Below, the terms “DBA User” and/or “DBA Account” refer to a database account which has DBA (database administrator) authorizations or takes on DBA-related activities according to the principle of segregation of functions (e.g., user administration, backup, etc.).

3. Basic security requirements

3.1. Version of the database used

Req 1	The Oracle database software must be approved by the manufacturer for productive operation and must be in in the “Premier Support” or “Extended Support” phase, in accordance with the Oracle Lifetime Support Policy.

11.1: Premier support up to August 2012, extended support up to August 2015.

11.2: Premier support up to January 2015, extended support up to January 2018.

All older versions are in the “Sustaining Support” phase, for which the manufacturer no longer provides regular security updates. This significantly increases the likelihood of an attack.

Reference:
http://www.oracle.com/us/support/lifetime-support/index.html
See PDF -> Resources ->“Lifetime Support Policy: Oracle Technology Products” and/or direct link.
http://www.oracle.com/us/support/library/lifetime-support-technology-069183.pdf

Motivation: Only a database software version for which the manufacturer provides complete support services guarantees secure, stable operation of the product.

For this requirement the following threats are relevant:
- Disruption of availability
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.29-1/6.0

4. Database hardening

This section lists the requirements for hardening the database system. The measures are comparable with those for operating system hardening. The requirements below are also used to minimize the impact of unauthorized access to database systems.

4.1. Service elimination

Req 2 Components of the Oracle-DBMS which are not required must not be installed.

During the installation it should be ensured that only the necessary components (for example, not Apache HTTP server, APEX, Spatial, etc.) are installed.

References:
http://docs.oracle.com/cd/B28359_01/install.111/b32002/install_overview.htm
http://docs.oracle.com/cd/E11882_01/install.112/e24321.pdf

Motivation: Additional components increase the likelihood of weaknesses arising and unnecessarily increase operating costs.

For this requirement the following threats are relevant:
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-2/6.0

Req 3 The Oracle HTTP server must not be installed on the DBMS server or muss be deactivated.

The Oracle HTTP server must not be installed on the same operating system instance as the DBMS or must be deactivated in order to comply with the N-tier architecture.

If use of the Oracle HTTP server is required, those responsible must ensure that it is installed on a dedicated server separate from the database system.

Check:
With the command $ORACLE_HOME/apache/apache/bin/apachectl the administrator can check whether the server is active.

Motivation: Additional functionalities result in additional security vulnerabilities

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Disruption of availability
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.29-3/6.0

Req 4 Oracle Application Express (APEX) must not be installed on the DBMS server or must be deactivated.

Oracle Application Express (APEX) must not be installed on the same operating system instance as the DBMS or must be deactivated in order to comply with the N-tier architecture.

Oracle Application Express is pre-installed as standard, together with the database, from Oracle 11g.

If use of Oracle Application Express is required, those responsible must ensure that it is installed on a dedicated server separate from the database system.

Check:
After installing APEX, the files are located in $ORACLE_HOME/apex and user accounts of the type FLOWS_*
exist in the database.

References:
http://docs.oracle.com/cd/E14373_01/install.32/e13366/overview.htm#i46634
http://docs.oracle.com/cd/E14373_01/appdev.32/e11838/sec.htm
http://www.oracle.com/technetwork/developer-tools/apex/overview/index.html
http://en.wikipedia.org/wiki/Oracle_Application_Express
http://www.springer.com/computer/database+management+%26+information+retrieval/book/978-1-4302-4731-9

Motivation: Additional functionalities result in additional security vulnerabilities.

Implementation example: To uninstall APEX, administrators need to log in and run the query @apxremov.sql (see http://download.oracle.com/docs/cd/E17556_01/doc/install.40/e15513.pdf, p. A-6)

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unnoticeable feasible attacks
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

For this requirement the following warranty objectives are relevant:

ID: 3.29-4/6.0

Req 5	Only one instance of the Oracle database system must be installed on one operating-system instance (hardware platform or virtualization guest).

Motivation: Motivation: If server hardware is used multiple times by multiple database systems, the risk increases of a larger group of people obtaining unauthorized access to systems for which they are not responsible technically or in terms of administration.

Implementation example: *If multiple database instances for different tasks (e.g., Internet and intranet) are running on one operating-system instance, the instances are not separated from one another. There is a risk that attackers could also corrupt the second database system. To separate the individual database instances deploy virtualization solutions.*

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-5/6.0

Req 6	(Default) databases that are not required must be deleted on the database system.

Motivation: When installing database systems, test or practice databases are often installed which are not required once the database goes productive. In the past, vulnerabilities of these test databases have become known which allow an attacker to gain privileged rights on the database system. Such knowledge enables an attacker to access the database system. Therefore, all databases that are not required shall be deleted.

For this requirement the following threats are relevant:
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.16-7/6.0

Req 7	Only those database instances which -have the same protection requirements (data protection class), -are under a single customer administrative authority and; -are operated, in terms of administration, by the same group of people,may be operated on one operating system instance (with physical or virtualized hardware).

Motivation: If server hardware is used multiple times by multiple database systems, the risk increases of a larger group of people obtaining unauthorized access to systems for which they are not responsible technically or in terms of administration.

Implementation example: If server hardware is used multiple times by multiple database systems, the risk increases of a largergroup of people obtaining unauthorized access to systems for which they are not responsible technically or interms of administration.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized use of services or resources
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.16-9/6.0

Req 8	All database services muss be set up in accordance with the least privilege principle on operating-system level.

Motivation: By using the least privilege principle, the risk of system corruption can be reduced.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.16-17/6.0

4.2. Users and roles

Req 9	(Default) user accounts and (default) roles must be deleted if these are not required. Where this is not possible or sensible, the user status must be set to “EXPIRED & LOCKED”.

NB: Database user accounts with the “EXPIRED & LOCKED” status can continue to perform activities in the database instance even though they are blocked. An active audit, where applicable, is also of no benefit here since such activities are not recorded. The blocking procedure is therefore associated with the risk of hidden misuse (automated jobs, for example). This means that blocking users is always the less satisfactory alternative.

The following default users are not required in a production environment:

ADAMS
ADLDEMOANDY
BLAKE
CDEMOCOR
CDEMORID
CDEMOUCB
CDEMO82
CLARK
DEMO
DIANE
FROSTY
HLW
HR
IX
JAKE
JILL
JONES
OE
PM
QS
QS_ADM
QS_CB
QS_CBADM
QS_CS
QS_ES
QS_OS
QS_WS
SCOTT
SH

Unless they are required, the following users should have the status “EXPIRED & LOCKED”:

Account	Explanation
ANONYMOUS	Oracle XML DB
CTXSYS	CTXSYS
DBSNMP	Oracle Enterprise Manager
DIP	Oracle Label Security
DSSYS	Dynamic Services Engine
EXFSYS	Rules Manager & Expression Filter
FLOWS_30000	Oracle Database Application Express
FLOWS_FILES	Oracle Database Application Express
LBACSYS	Oracle Label Security
MDDATA	Oracle Spatial
MDSYS	Oracle Spatial/Oracle Multimedia Locator
MGMT_VIEW	Oracle Enterprise Manager Database Control
ODM	Oracle Data Mining
ODM_MTR	Oracle Data Mining
OLAPSYS	OLAP
ORDDATA	Oracle Multimedia DICOM

ORDPLUGINS	Oracle Multimedia
ORDSYS	Oracle Multimedia
OUTLN	plan stability
ORACLE_OCM	Oracle Configuration Manager
OWBSYS	Oracle Warehouse Builder
REPADMIN	Replication user
SI_INFORMTN_SCHEMA	SQL/MM Still Image Standard
SPATIAL_CSW_ADMIN_USR	Oracle Spatial CSW Cache Manager
SPATIAL_WFS_ADMIN_USR	Oracle Spatial WFS Cache Manager
TRACESVR	Oracle Trace Server
WK_TEST	Oracle Ultra Search
WKPROXY	Oracle Ultra Search
WKSYS	Oracle Ultra Search
WMSYS	Oracle Workspace Manager
XDB	Oracle XML DB
XS$NULL	internal

References:
Sample schemas 11.2 http://download.oracle.com/docs/cd/E11882_01/server.112/e10831.pdf

Default passwords and their hash codes http://www.petefinnigan.com/default/default_password_list.htm

List of default users http://www.orafaq.com/wiki/List_of_default_database_users

Securing Oracle Database User Accounts
http://docs.oracle.com/cd/E11882_01/server.112/e10575/tdpsg_user_accounts.htm

Motivation: Installation programs of database systems automatically set up a large number of users and roles which are not needed by the people who use the database. These roles are envisaged, e.g., for practice and test databases. Knowledge of default users and roles allows an attacker to gain (privileged) access to the database system. Users and roles which are not required should therefore be deleted or blocked (users only).

Implementation example: No generally applicable procedure can be specified for deleting these schemas, please refer to each one individually in “My Oracle Support” (MOS).

To set the status of a user to “EXPIRED & LOCKED”:
ALTER USER <username> ACCOUNT LOCK;
ALTER USER <username> PASSWORD EXPIRE;

Check: Show all users with status:
SELECT Username, Status FROM DBA_USERS;

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized use of services or resources
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-9/6.0

4.3. Passwords

Req 10	Default passwords must be changed.

Default passwords must be changed immediately after the system installation and before a network listener is started.

Check:
Display user accounts with default passwords:

```
SELECT * FROM dba_users_with_defpwd;
```

Command to change a password:

```
ALTER USER <username> IDENTIFIED BY <new password>;
```

References:
http://www.oracle.com/technetwork/articles/sql/11g-security-100258.html

A list of default passwords and their hash codes is available at:
http://www.petefinnigan.com/default/default_password_list.htm

Motivation: Numerous default passwords are known and enable direct access to the system or database.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-10/6.0

Req 11	The number of failed logins for non-SYSDBA accounts must be limited for all existing profiles by the FAILED_LOGIN_ATTEMPTS parameter.

Check:

```
SELECT Username, Profile FROM DBA_USERS;
SELECT Profile, Resource_Name, Limit FROM DBA_PROFILES
WHERE Resource_Name = 'FAILED_LOGIN_ATTEMPTS';
```

Motivation: Limiting the login attempts reduces the risk of a brute force attack on passwords.

Implementation example: The FAILED_LOGIN_ATTEMPTS parameter should be set to an appropriate number for all available profiles.

```
ALTER PROFILE <profile name> LIMIT FAILED_LOGIN_ATTEMPTS <number>;
```

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized use of services or resources
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-11/6.0

Req 12	The password lock time for non-SYSDBA accounts muss be defined for all existing profiles.

Check:

```
SELECT Username, Profile FROM DBA_USERS;
SELECT Profile, Resource_Name, Limit FROM DBA_PROFILES
WHERE Resource_Name = 'PASSWORD_LOCK_TIME';
```

Motivation: The setting reduces the likelihood of a brute force attack on passwords without enabling a denial-ofservice attack on the accounts.

Implementation example: The administrator sets the PASSWORD_LOCK_TIME parameter for all available profiles, e.g., to 1/2,880 days (= 30 seconds).

```
ALTER PROFILE <profile name> LIMIT PASSWORD_LOCK_TIME <number>;
```

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized use of services or resources
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-12/6.0

Req 13	If a password is used as an authentication attribute, it must have at least 12 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

A system may only accept passwords that comply with the following complexity rules:
- Minimum length of 12 characters.
- Comprising at least three of the following four character categories:
  - lower-case letters
  - upper-case letters
  - digits
  - special characters

The usable maximum length of passwords shall not be limited to less then 25 characters. This will provide more freedom to End Users when composing individual memorizable passwords and helps to prevent undesired behavior in password handling.

When a password is assigned, the system must ensure that the password meets these policies. This must be preferably enforced by technical measures; if such cannot be implemented, organizational measures must be established.
If a central system is used for user authentication [see also Root Security Requirements Document[i] "3.69 IAM (Identity Access Management) - Framework"], it is valid to forward or delegate this task to that central system.

Permissible deviation in the password minimum length
Under suitable security-related criteria, conditions can potentially be identified for a system that enable the minimum password length to be reduced:
- It is generally permissible to reduce the minimum password length for systems that use additional independent authentication attributes within the authentication process in addition to the password (implementation of 2-Factor or Multi-Factor Authentication).
- Any reduction in the minimum password length must be assessed individually by a suitable technical security advisor (e. g. a PSM from Telekom Security) and confirmed as permissible. In the assessment, the surrounding technical, organizational and legal framework parameters must be taken into account, as well as the system-specific protection requirements and the potential amount of damage in the event of security incidents.
- The absolute minimum value of 8 characters length for passwords must not be undercut.

Motivation: Passwords with the above complexity offer contemporary robustness against attacks coupled with acceptable user friendliness. Passwords with this level of complexity have proven their efficiency in practice. Trivial and short passwords are susceptible to brute force and dictionary attacks and are therefore easy for attackers to determine. Once a password has been ascertained it can be used by an attacker for unauthorized access to the system and the data on it.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

For this requirement the following warranty objectives are relevant:

ID: 3.01-26/7.0

Req 14	If a password is used as an authentication attribute for technical accounts, it must have at least 30 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

Technical user accounts are characterized by the fact that they are not used by people. Instead, they are used to authenticate and authorize systems to each other or applications on a system.

A system must only use passwords for technical user accounts that meet the following complexity:
- Minimum length of 30 characters
- Comprising at least three of the following four character categories:
  - lower-case letters
  - upper-case letters
  - digits
  - special characters

Motivation: Due to their use in machine-to-machine (M2M) communication scenarios, technical user accounts are often equipped with privileges that can be of high interest to an attacker to compromise infrastructures. Without mechanisms of extensive compromise detection, the risk of a password being determined or broken by an attacker can increase significantly over time. A significant increase in password length counteracts these risks and can also be implemented particularly easily in M2M scenarios, since handling a very long password is not a particular challenge for a machine (as opposed to a person).

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

For this requirement the following warranty objectives are relevant:

ID: 3.01-27/7.0

Req 15	If a password is used as an authentication attribute, it must be changed after 12 months at the latest.

The maximum permitted usage period for passwords is 12 months.
If a password reaches the maximum permitted usage period, it must be changed.

For this purpose, the system must automatically inform the user about the expired usage period the next time he logs on to the system and immediately guide him through a dialog to change the password. Access to the system must no longer be permitted without a successfully completed password change.
For technical user accounts (M2M or Machine-2-Machine), which are used for the authentication and authorization of systems among themselves or by applications on a system, automated solutions must also be implemented to comply with the permitted usage period for passwords.

Alternatively, if such an automatic mapping of the process for changing the password cannot be implemented, an effective organizational measure must be applied instead, wich ensures a binding manual password change at the end of the permissible period of use.

Motivation: Unlike more modern authentication attributes, passwords are easier to attack. Without specific measures for reliable, technically automated detection of compromises, the risk of a password being discovered or broken by an attacker can increase considerably over time.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

For this requirement the following warranty objectives are relevant:

ID: 3.01-30/7.0

Req 16	If a password is used as an authentication attribute, the reuse of previous passwords must be prevented.

A history of the previously used passwords must be recorded for each user account. When a password change is initiated for a user account, the new password must be compared with this password history. If the reuse of a password is detected, the password change must be rejected. This validation process must be implemented in the system on the basis of technical measures. If a central IAM system is used for user authentication, the implementation can be forwarded to the central IAM system or outsourced there [see also Root Security Requirements Document[i] "3.69 IAM (Identity Access Management) - Framework"].

In general, the password history should ensure that a password that has already been used can never be used again.

However, due to technical limitations, a password history cannot be recorded indefinitely in many IT/NT products. In this case, the following basic rules must be observed:

- a password that has already been used must not be reusable for a period of at least 60 days (measured from the point in time at which the affected password was replaced by another)
- in systems in which the period of at least 60 days cannot be implemented, the longest possible period must be configured. In addition, it must be confirmed by a Project Security Manager (PSM) that the configured period is still sufficient in the overall context of the system with regard to the security requirement.

Annotation:
Some IT/NT products do not offer any technical configuration parameters with which the password history can be linked directly to a time period, but only allow the definition of the number of passwords to be recorded. In such cases, the time period can alternatively be ensured by linking the following, usually generally available configuration parameters. Within the resulting policy, a user can only change his password once a day and, due to the number of passwords recorded, can reuse an old password effectively after 60 days at the earliest.

- Minimum Password Age: 1 day

- Password History: Record of the last 60 passwords used

With this implementation variant, it should be noted that the minimum age for the password should not be more than one day in order not to inappropriately restrict the user with regard to the fundamental need to be able to change the password independently at any time.

Motivation: Users prefer passwords that are easy to remember and often use them repeatedly over long periods of time when the system allows. From the user's point of view, the behavior is understandable, but effectively leads to a considerable reduction in the protective effect of this authentication parameter. With adequate knowledge of the user or information obtained from previous system compromises, an attacker can gain access to supposedly protected user accounts. Particularly in situations in which new initial passwords are assigned centrally as part of an acute risk treatment, but users change them immediately to a previous password for the sake of simplicity, there is a high risk that an attacker will resume illegal access. It is therefore important to prevent users from reusing old passwords.

Implementation example: [Example 1]
Linux System

set entry in /etc/login.defs

```
PASS_MIN_DAYS 1
```

and additionaly set entries in PAM Konfiguration

```
password requisite pam_pwquality.so try_first_pass local_users_only enforce-for-root retry=3 remember=60
password sufficient pam_unix.so sha512 shadow try_first_pass use_authtok remember=60
```

[Example 2]
Windows System

set entries in GPO

Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies\Password Policy\Minimum password age = **1**
Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies\Password Policy\Enforce password history = **24** (technical maximum)

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

For this requirement the following warranty objectives are relevant:

ID: 3.01-28/7.0

Req 17	If passwords are used as an authentication attribute, those must be stored using a suitable and approved "Password Hashing" method to protect against offline-attacks like brute force or dictionary attacks.

This requirement relates to the storage of passwords in all types of user databases, as used in this system, in order to authenticate incoming access (local or remote) by users or other systems.

If an attacker obtains the copy of a user database of the system, he is able to bring it into a fully independent environment and utilize automatized dictionary or brute force attacks to determine contained passwords. Specialized tools in combination with high computing power allow for high cracking rates in a relatively short period of time, if protective measures are insufficient. Due to the independency from the source system, such an offline attack happens unnoticed.

The following countermeasure must be implemented, since this ensures best possible protection against offline attacks:

- passwords must be stored using a cryptographic one-way function ("Password Hashing") which is suitable for that purpose and verifiably secure as matters stand

Please Note:
valid password hashing algorithms are described in Security Requirement Catalog "3.50 Cryptographic Algorithms and Security Protocols".

Explicitly NOT PERMISSIBLE is:

- to store passwords in cleartext
- to store passwords in any format which can be directly backcalculated
- to store passwords using reversible encryption

Please Note:
In this context, "directly backcalculatable formats" refers to those that simply encode the password, without involving a secret key in the transformation process. Since the password will no longer show up as original cleartext after it has been processed, those formats may easily be mistaken to provide confidentiality. Effectively, they do not offer any protection. The enconding is fixed and therefore an attacker can easily make use of it to compute the original cleartext password from the encoded string.
Examples for directly backcalculatable formats are: "base64", "rot13"
"Reversible" are all encryption methods which, using the appropriate key, enable encrypted content to be transformed back into the original content. Accordingly, with reversible encryption there is always the challenge of keeping the key secure and protecting it from unauthorized access. Reversibility is a required fundamental property in many areas of encryption applications, e.g. for transferring confidential messages, but it is counterproductive for storing passwords: a stored password must remain comparable by means of technical methods, but it must no longer be possible to convert it back into plain text in order to protect it as well as possible from unauthorized viewing.
Examples for reversible encryption are: "AES", "CHACHA20", "3DES", "RSA"

Motivation: Without protective measures, an attacker in possession of a user database copy is able to determine masses of contained passwords in short time by merely trying out character string combinations or making use of dictionaries. Passwords stored in cleartext or any backcalculatable format are fully defenseless to an offline attack. Once a password has been ascertained it can be used by an attacker for unauthorized access to the system and the data on it.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

For this requirement the following warranty objectives are relevant:

ID: 3.01-24/7.0

Req 18	Only authorized users and SYS may be entered in the password authentication file.

The password file is used to save usernames and passwords which should have the SYSDBA privilege and/or the SYSOPER privilege. The password file enables the authentication of these users even if a database instance is not running. Due to these high levels of authorization, they should be limited to a minimum number. A regular check of the user accounts included in the password file should be established.

Check:
The database view SYS.V$PWFILE_USERS shows the users in the password file:SELECT Username, Sysdba, Sysoper FROM V$PWFILE_USERS; or
SELECT Username, Sysdba, Sysoper FROM sys.V_$PWFILE_USERS;

Motivation: Minimizing the area open to attack by minimizing authorizations.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-18/6.0

4.4. Principle of least privilege

Req 19	All rights to the DBMS_OBFUSCATION_TOOLKIT must be revoked from the PUBLIC pseudo role.

Although Oracle has replaced the DBMS_OBFUSCATION_TOOLKIT with the DBMS_CRYPTO package, the former is still required for certain tasks.

Check:
```
SELECT Table_Name, Owner, Grantee, Privilege FROM DBA_TAB_PRIVS
WHERE Grantee = 'PUBLIC'
AND Table_Name = 'DBMS_OBFUSCATION_TOOLKIT';
```

Motivation: By revoking the rights of the PUBLIC role, the administrator prevents unauthorized users from being able to decrypt the data.

Implementation example: REVOKE EXECUTE ON DBMS_OBFUSCATION_TOOLKIT FROM PUBLIC;

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-19/6.0

Req 20	A database service must not run with root rights or other operating system-related administrative rights.

Exception are: Oracle Grid Infrastructure (ASM/RAC), for instance ora_asm, ora_dism

Check: Look at the process lists.

Motivation: If this requirement is not met, a security vulnerability in the database service could cause the operating system to be compromised (e.g., through a buffer overflow).

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

For this requirement the following warranty objectives are relevant:

ID: 3.29-20/6.0

4.5. External procedures

Req 21	If it is not necessary to call external procedures, the configuration entry “PROGRAM=extproc” must be removed from the listener.ora file.

Motivation: These configurations make it possible to run external programs under the operating system account under which the database instance is running. These settings should be deactivated in order to safeguard the database system.

For this requirement the following threats are relevant:
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-21/6.0

Req 22	If the “extproc” functionality is required, in addition to the normal database listener, a separate EXTPROC listener must be used which runs under its own unprivileged account.

Motivation: There is a risk of attackers infiltrating an external procedure as a Trojan and thus taking control of the super user privileges. A dedicated listener with restricted privileges reduces the risk of this type of attack.

Implementation example: The administrator creates an additional listener.ora under the unprivileged user account of the operating system and configures the entry for the extproc listener in this.

Reference:
http://www.oracle.com/technetwork/database/security/twp-security-checklist-database-1-132870.pdf

For this requirement the following threats are relevant:
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-22/6.0

Req 23	Calling external procedures must be restricted to only the required functions in the configuration file listener.ora.

Motivation: In order to make the non-audit-proof exchange of the external procedures which can be called more difficult, only the EXTPROC listener and the system administrator may have access rights to the directories in which the external procedures are saved. Application responsible should have no access rights after that.

Implementation example: Restriction with the option EXTPROC DLLS=ONLY:DLL1:DLL2 in the extproc.ora file and specifying the absolute path for each required DLL (see http://download.oracle.com/docs/cd/E11882_01/appdev.112/e17125.pdf S.14-6).

During implementation, it should be noted that the configuration of the listener.ora (or extproc.ora) imposes a restriction through explicitly numbering and naming the permissible external procedures. The impact of this can be seen with the ONLY option, e.g., EXTPROC DLLS=ONLY:DLL1:DLL2:..., whereby this option requires the absolute path to be specified for the required DLLs.

Reference:
http://download.oracle.com/docs/cd/E11882_01/appdev.112/e17125.pdf Seite.14-6

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-23/6.0

4.6. SQL extensions with operating-system or network access

Req 24	The use of external jobs (local external job or remote external job) must be restricted to DBA user accounts.

Reference:
http://docs.oracle.com/cd/B28359_01/server.111/b28310/schedover004.htm

Motivation: The functions enable operating system calls to be carried out locally or remotely. The risk is that with this functionality, in reality a login on the database server is replaced (carried out by the function itself). Therefore, this function should only granted to DBA user accounts.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-24/6.0

Req 25	The UTL_FILE_DIR parameter must not be used.

Motivation: Special access rights to directories on the database server or the files it contains cannot be adequately configured using UTL_FILE_DIR. When using the UTL_FILE_DIR initialization parameter, in principle all database users have read and write rights to the files located in the specified directories.

Implementation example: The person responsible must remove the UTL_FILE_DIR parameter from the init.ora file. However, if the user still wants to have read or write access to the files in the directories located on the database server, database objects of the DIRECTORY type must be created for this and approved for read or write access. It is then possible to assign access rights to these directories per user or role. It is also necessary to create directories which are to be approved for read and write access which remain under the control of system administrators. As a result, the CREATE DIRECTORY system privilege may only be assigned to system administrators and, under no circumstances, to application accounts.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-25/6.0

4.7. Access rights

Req 26	The “Oracle Software Owner” must be a system account that is only used for the administration of the database system.

Motivation: There is a risk that an unauthorized and uncontrolled change could be made to the installation.

Implementation example: The “root” system account or another system account must not be made available as the Database Software Owner. Generally, the administration account for the database is “Oracle”.

Exception: Oracle Grid Infrastructure (ASM/RAC).

In addition, conversely, this system account must not be used for the installation of other products (e.g., Oracle HTTP server) at the same time.

For this requirement the following threats are relevant:
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-26/6.0

Req 27	The accounts of the Oracle Software Owner and those of the (personalized) administrators must be the individual members of the “dba” OS group in the “/etc/group” file.

Motivation: To ensure a controlled administration, there should be only one database administrator account or administrator accounts personalized in another way.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-27/6.0

Req 28	The Oracle Software Owner must be the owner of $ORACLE_HOME/bin and all the files located in it.

Motivation: There is a risk of uncontrolled access or an uncontrolled execution.

Implementation example: Please note: The following executable files in the $ORACLE_HOME/bin directory emtgtctl2, extjob, jssu, nmb, nmhs, nmo, oracle, oradism
belong to the Oracle Software Owner and/or have the SUID bit set. Administrators should not change this.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-28/6.0

Req 29	The access rights for the bin directory ($ORACLE_HOME/bin) must be set to 0755 (rwxr-xr-x) or lower.

Motivation: There is a risk of uncontrolled access and uncontrolled execution due to missing or incorrect access rights for the Oracle bin directory.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.29-29/6.0

Req 30	Access to files which may contain Oracle data with a need for protection must be restricted to the DBA group.

Files belonging to this category include:

- Files with information on configuration and authentication (e.g., init.ora, spfile.ora, snmp_ro.ora, snmp_rw.ora, catsnmp.sql, orapw<SID>, listener.ora, xsqlconfig.xml, soapConfi.xml, ...)
- Net8 trace and log files
- data files
- control files
- online redo files
- archive files
- audit files

Motivation: There is a risk of unauthorized access to confidential information.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data

- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-30/6.0

5. Data communication

This section summarizes the requirements concerning queries from other database systems or the exchange of data between database systems. This can be implemented either by means of so-called ad-hoc queries, or by setting a data connection to share data between the systems. Oracle realizes data connections between databases via database links.

Req 31	Public database links must not be used.

Motivation: Public database links can be used by anyone, which means that there is potential for misuse. In the case of fixed database links (= a user with a fixed username/password), the opportunity for misuse is obvious. However, even the case of a connected user database link is not desirable because a compromised password on the source database can immediately also be used on the remote database, since these are identical.

Implementation example: Delete public database links and revoke the privilege CREATE PUBLIC DATABASE LINK from owners.

Check:

```
SELECT Owner, Db_Link FROM DBA_DB_LINKS WHERE Owner = 'PUBLIC';
SELECT COUNT(0), owner, db_link, username FROM sys.dba_db_links
GROUP BY owner, db_link, username ORDER BY 1,3;
```

This statement should not contain any lines with Owner=‘PUBLIC’ or any lines missing a username.

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-31/6.0

Req 32	Private database links must only be used with a “CONNECT TO” clause (fixed private database links).

Reference:
http://docs.oracle.com/cd/E11882_01/server.112/e25494/ds_concepts002.htm

Check:
In the following output, all links must have a username entered and the owner must not be PUBLIC.

```
SELECT Owner, Db_Link, Username, Host FROM DBA_DB_LINKS;
SELECT COUNT(0), owner, db_link, username FROM sys.dba_db_links
GROUP BY owner, db_link, username ORDER BY 1,3;
```

This statement should not contain any lines with Owner=‘PUBLIC’ or any lines missing a username.

Motivation: Private database links without a “CONNECT TO“”clause require identical user accounts on the local and on the remote database. This results in the local database account being compromised along with the remote user account. A private database link with the “CONNECT TO” clause and a password enables the authorized local user, along with the user specified in the link, to access the remote database. The area open to attack can be reduced by restricting the link as a private link (only the owner can use this) and by restricting it to the users defined in the link.

Implementation example: The example below creates a private fixed database link with the name sales.us.americas.example_auto.com to the remote database 'sales_us'. To do this, the current user (database link owner) uses the remote user scott and his password “password”.

```
CREATE DATABASE LINK sales.us.americas.example_auto.com CONNECT TO scott IDENTIFIED BY password USING 'sales_us';
```

If an additional or other user now accesses the objects in the remote database, a view of the owner’s objects is created and the local user is given the necessary privileges for this. Thus only one user uses the database link directly.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-32/6.0

6. System monitoring

6.1. Logging

Req 33	The system clock must be synchronized to an accurate reference time (Time Standard).

A time reference source must be used which provides a time signal based on the Coordinated Universal Time ("UTC" = "**U**niversal **T**ime **C**oordinated").

Please Note: The UTC-synchronized system time may be transformed to local time using a corresponding timezone configuration setup for any output of time information, as long as this timezone adjustment is fully accountable.

Systems belonging to the same security domain must synchronize to one and the same time reference source.

Motivation: Reference time synchronization may be a technical prerequisite for many time-dependent mechanisms, for example: Validation of Certificates; Authentication. It is also much-needed to generate exact timestamps for logged events, since without the often required time-related correlation in case of a Security Incident or during a Problem Analysis cannot be achieved.

Implementation example: some valid time reference sources:

- trustworthy NTP ("**N**etwork**T**ime**P**rotocol") Server on the IP network
- DCF77 radio signal received via a physically connected receiver
- GPS radio signal received via a physically connected receiver

For this requirement the following threats are relevant:
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.01-32/7.0

Req 34	Accesses to critical database procedures and database content must be logged.

Logging of security-relevant user actions must comply with national legislation currently in force. When implementing measures resulting from this requirement, the applicable participation rights of the responsible employee representatives/trade unions as well as the works and collective agreements shall be observed.

Motivation: Secure, traceable database operation requires important operating information to be logged. This includes, for example, the logging of failed login attempts to uncover possible intrusion attempts.

Implementation example: Using Oracle Auditing and the implementation of triggers, almost any access can be logged.

For this requirement the following threats are relevant:
- Denial of executed activities

For this requirement the following warranty objectives are relevant:

ID: 3.29-34/6.0

Req 35	Database Event Triggers must not contain application objects of any kind. Only those procedures saved in the “SYS” dictionary schema may be called from the Database Event Trigger.

Motivation: If an application procedure is included in a Database Event Trigger, there is a risk that an experienced and malicious user could exploit this trigger by manipulating the process code and extending his user rights.

For this requirement the following threats are relevant:
- Disruption of availability
- Denial of executed activities

For this requirement the following warranty objectives are relevant:

ID: 3.29-35/6.0

Req 36	Security relevant events must be logged with a precise timestamp and a unique system reference.

Systems must log the occurrence of security-relevant incidents. So that these events can be evaluated and classified, they must be logged together with a unique system reference (e.g., host name, IP or MAC address) and the exact time the incident occurred ("Timestamp").

Exceptions of this requirement are systems for which logging cannot be implemented because of building techniques, use case or operation area. Examples for these kind of systems are customer devices such as Smartphones or IADs/ home gateways (e.g. Speedport).

The Timestamp of a logged event must contain at least the following information:
- date of the event (Year, Month, Day)
- time of the event (Hours, Minutes, Seconds)
- Timezone, those information belongs to

When logging, the applicable legal and operational regulations must be observed. The latter also include agreements that have been made with the company's social partners. Following these regulations logging of events is only allowed for a defined use case. Logging of events for doing a work control of employees is not allowed.

In addition - as for any data that is processed by a system - an appropriate protection requirement must also be taken into account and implemented for logging data; this applies to storage, transmission and access. In particular, if the logging data contains real data, the same protection requirements must be taken into account that is also used for the regular processing of this real data within the source system.

Typical event that reasonable should be logged in many cases are:

Event	**Event data to be logged**
Incorrect login attempts	• User account, • Number of failed attempts, • Source (IP address, client ID / client name) of remote access
System access from user accounts with administrator permissions	• User account, • Access timestamp, • Length of session, • Source (IP address) of remote access
Account administration	• Administrator account, • Administered user account, • Activity performed (configure, delete, enable and disable)

Change of group membership for accounts	• Administrator account, • Administered user account, • Activity performed (group added or removed)
Critical rise in system values such as disk space, CPU load over a longer period	• Value exceeded, • Value reached (Here suitable threshold values must be defined depending on the individual system.)

Logging of additional security-relevant events may be meaningful. This must be verified in individual cases and implemented accordingly where required.

Motivation: Logging security-relevant events is a basic requirement for detecting ongoing attacks as well as attacks that have already occurred. This is the only way in which suitable measures can be taken to maintain or restore system security. Logging data could be used as evidence to take legal steps against attackers.

For this requirement the following threats are relevant:
- Denial of executed activities
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.01-33/7.0

Req 37	Applicable retention and deletion periods must be observed for security-relevant logging data that is recorded locally.

From an IT security perspective, local storage of security-relevant logging data on a system is not mandatory. Since the local storage can be damaged in the event of system malfunctions or manipulated by a successful attacker, it can only be used to a limited extent for security-related or forensic analyses. Accordingly, it is relevant for IT security that logging data is forwarded to a separate log server.

Local storage can nevertheless take place; for example, if local storage is initially indispensable when generating the logging data due to technical processes or if there are justified operational interests in also keeping logging data available locally.

The following basic rules must be taken into account when storing logging data locally:

- Security-related logging data must be retained for a period of 90 days.
  (*This requirement only applies if no additional forwarding to a separate log server is implemented on the system and the logging data is therefore only recorded locally.*)
- After 90 days, stored logging data must be deleted immediately.

Deviances

Different retention periods and deletion periods may exist due to legal or regulatory requirements (especially in connection with personal data) or may be defined by contractual agreements. In these cases, the applicable periods must be agreed individually with a Project Security Manager (PSM) / Data Privacy Advisor (DPA) or are specified by them.

Motivation: Logging data is an immensely important IT security tool for preventing, detecting and clearing up system faults, security and data privacy incidents. On the other hand, the recording of logging data, like any other data processing, is also subject to legal and regulatory requirements. Accordingly, guidelines must be adhered to that reconcile the two.

Implementation example: Taking into account the current legal situation and applicable data privacy regulations, the following deletion periods for locally stored security-relevant logging data are implemented on an exemplary telecommunications system:

- Standard System Logs: Deletion after 90 days at the latest
- Logging of public IP addresses: Deletion (or anonymization) after 7 days at the latest
- Logging of the assignment of dynamic public IP addresses by the telecommunication solution: Deletion after 7 days at the latest
- Logging of non-billing-relevant call detail records: Deletion after 7 days at the latest
- Logging of the content of e-mail and SMS: Deletion after 24 hours at the latest
- Logging of the domain queries handled by the DNS server of the telecommunications solution: Deletion after 24 hours at the latest

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Denial of executed activities
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.01-34/7.0

Req 38	Security-relevant logging data must be forwarded to a separate log server immediately after it has been generated.

Logging data must be forwarded to a separate log server immediately after it has been generated. Standardized protocols such as Syslog, SNMPv3 should be preferred.

Motivation: If logging data is only stored locally, it can be manipulated by an attacker who succeeds in compromising the system in order to conceal his attack and any manipulation he has performed on the system. This is the reason why the forwarding must be done immediately after the event occurred.

For this requirement the following threats are relevant:
- Unauthorized modification of data
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.01-35/7.0

Req 39	For security-relevant logging data that is forwarded to the separate log server, compliance with the applicable retention and deletion periods must be ensured.

The following basic rules must be taken into account:

- security-related logging data must be retained for a period of 90 days on the separate log server.
- after 90 days, stored logging data must be deleted immediately on the separate log server.

Deviances

Different retention periods and deletion periods may exist due to legal or regulatory requirements (especially in connection with personal data) or may be defined by contractual agreements. In these cases, the applicable periods must be agreed individually with a Project Security Manager (PSM) / Data Privacy Advisor (DSB) or are specified by them.

Log server under the responsibility of a third party
If the selected separate log server is not within the same operational responsibility as the source system of the loggin data, it must be ensured that the responsible operator of the log server is aware of the valid parameters for the logging data to be received and that they are adhered to in accordance with the regulations mentioned here.

Motivation: Logging data is an immensely important IT security tool for preventing, detecting and clearing up system faults, security and data privacy incidents. On the other hand, the recording of logging data, like any other data processing, is also subject to legal and regulatory requirements. Accordingly, guidelines must be adhered to that reconcile the two.

Implementation example: Taking into account the current legal situation and applicable data privacy regulations, the following deletion periods for forwarded security-relevant logging data from an exemplary telecommunications system are implemented on the separate log server:

- Standard System Logs: Deletion after 90 days at the latest
- Logging of public IP addresses: Deletion (or anonymization) after 7 days at the latest
- Logging of the assignment of dynamic public IP addresses by the telecommunication solution: Deletion after 7 days at the latest
- Logging of non-billing-relevant call detail records: Deletion after 7 days at the latest
- Logging of the content of e-mail and SMS: Deletion after 24 hours at the latest
- Logging of the domain queries handled by the DNS server of the telecommunications solution: Deletion after 24 hours at the latest

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Denial of executed activities
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.01-36/7.0

Req 40	The system must provide logging data that is required to detect the system-specific relevant forms of attack in a SIEM.

The forms of attack that are typically to be expected for the present system must be systematically analyzed and identified.
The MITRE Attack Matrix (https://attack.mitre.org) can be used as a structured guide during such an identification.

It must be ensured that the system generates appropriate logging data on events that are or may be related to these identified forms of attack and that can be used to detect an attack that is taking place.

The logging data must be sent to a SIEM immediately after the system event occurs.
SIEM (Security Information & Event Management) solutions collect event log data from various source systems, correlate it and evaluate it automatically in real time in order to detect anomalous activities such as ongoing attacks on IT/NT systems and to be able to initiate alarms or countermeasures.
The immediate receipt of system events is therefore absolutely crucial for the SIEM to fulfill its protective functions.

Note:
The immediate need to connect a system to a SIEM is specifically regulated by the separate "Operation" security requirements catalogs.
If the present system does not fall under this need, the requirement may be answered as "not applicable".

Motivation: A SIEM as an automated detection system for attacks can only be effective if it continuously receives sufficient and, above all, system-specific relevant event messages from the infrastructures and systems to be monitored.

General standard event messages may not be sufficient to achieve an adequate level of detection and only allow rudimentary attack detections.

Implementation example: An example system allows end users to log in using a username and password. One of the typical forms of attack for this system would be to try to discover and take over user accounts with weak or frequently used passwords by means of automated password testing (dictionary or brute force attack). The example system is configured to record every failed login event in system protocols ("logs"). By routing this logging data in parallel to a SIEM, the SIEM can detect in real time that an attack is obviously taking place, alert it and thus enable immediate countermeasures.

ID: 3.01-37/7.0

Req 41	Important database services and instances must be monitored continually for misuse scenarios.

The monitoring of user actions for misuse shall comply with national legislation currently in force (for details see the “Security Requirement on Misuse Detection”).

Motivation: There are many conceivable ways to misuse database systems. Users generate an unusually high data usage rate or operate at unusual times of day. Attackers utilize unusual and critical commands for database queries, as well as tools and malware to extend their rights. To detect misuse, database systems should be continually monitored for misuse scenarios, e.g. by means of database triggers and log monitoring.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.16-32/6.0

7. Oracle-specific requirements

In addition to the security requirements derived from the general database security requirements, the manufacturer-specific requirements listed below should be taken into account. Below, the terms “DBA User” and/or “DBA Account” refer to a database account which has DBA (database administrator) authorizations or takes on DBA-related activities according to the principle of segregation of functions (e.g., user administration, backup, etc.).

7.1. Authentication

Req 42	It must be ensured that the “Proxy Authentication” function is only assigned to technical accounts and not to the accounts of natural persons.

References:
http://www.oracle.com/technetwork/database/security/index-092912.html
http://docs.oracle.com/cd/E11882_01/java.112/e16548/proxya.htm
http://www.databasejournal.com/features/oracle/article.php/3910651/Using-Proxy-Authentication-Methods-in-Oracle-Database-11g.htm

Motivation: The Proxy Authentication function is typically used in conjunction with web and application servers and enables these to handle a variety of user logins and sessions using one single database connection. The web or application server, as a “proxy”, logs the user into the database using his proxy account. Thus each user of the web application works with his own individual Oracle account and the rights to the database linked with this and not with the global technical account of the web or application server.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-42/6.0

Req 43	The operating system authentication must be deactivated for all non-DBA and non-operational users who do not require automatic login from the operating system.

Check:
Check OS group dba for unwanted members.
Check users of the following query:

```
SELECT Username FROM DBA_USERS
WHERE PASSWORD = 'EXTERNAL';
```

Motivation: The database security should depend as little as possible on the security of the underlying operating system.

Implementation example: The number of members in the DBA operating system group should be restricted. The IDENTIFIED EXTERNALLY mode should not be used in user profiles. Accounts which are not used for operational purposes, and for which automatic login via the operating system is not required (e.g., for backup), must not have account names with OS_AUTHENT_PREFIX. The default value for OS_AUTHENT_PREFIX is OPS$.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-43/6.0

Req 44 User login via an external (remote) operating system authentication must be deactivated.

Motivation: Oracle provides the option of relying on an external (remote) operating system authentication for the user login. Since the remote system cannot be relied upon, these options must be set to FALSE.

Implementation example: Administrators must set the parameters in the init.ora configuration file as shown below:

```
REMOTE_OS_AUTHENT=FALSE
REMOTE_OS_ROLES=FALSE
```

The default value for Oracle11g is FALSE.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Unnoticeable feasible attacks

For this requirement the following warranty objectives are relevant:

ID: 3.29-44/6.0

Req 45 Database roles must not be assigned via the operating system.

Motivation: Oracle provides the option of having database user roles managed by the operating system. This option carries hidden security risks; the person responsible must therefore deactivate this function. The tasks and responsibilities of DBA and operating system administration must be separated.

Implementation example: The FALSE default value of the parameter “OS_ROLES” in init.ora must not be changed.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-45/6.0

7.2. Listener and network connection

Req 46 Access to the TNS listener must be restricted to authorized users through technical measures.

Various technical options are available for restricting the access to authorized users.

Dynamic IP address
For computers with a dynamic IP address, access via a secure jump service / terminal server or certificate-based authentication is required since access restriction at IP level is no longer provided.

Fixed IP address
For computers with a fixed IP address, the access restriction can be implemented using Oracle valid node checking, Access Control Lists (ACLs) and network or host-based systems.

Motivation: Restricting access to authorized users drastically reduces the likelihood of attacks on the database system.

Implementation example: Example of implementation of Oracle Valid Node Checking:
For Oracle valid node checking, the following lines are to be added to the listener sqlnet.ora configuration file (for older systems protocol.ora):

```
TCP.VALIDNODE_CHECKING = YES
TCP.INVITED_NODES = (<host_1>, <host_2>, ...)
```

(The value for <host_x> is either an IP address or a DNS name)

Please note: TCP.INVITED_NODES have priority over TCP.EXCLUDED_NODES. Furthermore, the Oracle listener should be protected via the IP filter of the operating system.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-46/6.0

Req 47	The TNS listener must be protected against unauthorized configuration.

The TNS listener is protected against unauthorized configuration by adding the ADMIN_RESTRICTIONS_<listenername>=ON parameter to the listener.ora configuration file.

Reference:
Listener Control Utility
http://docs.oracle.com/cd/B28359_01/network.111/b28317/lsnrctl.htm
Oracle Corporation: Security Guide, 11g Release 2 (11.2)
http://docs.oracle.com/cd/E14072_01/network.112/e10574.pdf
Integrigy: White Paper “Oracle Database Listener Security Guide”, April 2007,
http://www.integrigy.com/security-resources/whitepapers/Integrigy_Oracle_Listener_TNS_Security.pdf

Motivation: Unauthorized persons must be prevented from reconfiguring the listener since this has an impact on the security of the database system.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Disruption of availability

For this requirement the following warranty objectives are relevant:

ID: 3.29-47/6.0

7.3. Privileges for users and roles

Req 48	It must be ensured that no database privileges for application objects are granted to the PUBLIC user group.

Check:

```
SELECT table_name, owner, privilege
from dba_tab_privs
where grantee = 'PUBLIC'
and owner not in (‘SYS,’SYSTEM’,’OUTLN’,’DBSNMP’);
```

The exclusion of further user accounts (e.g., XDB,CTXSYS, WMSYS, etc.) to be assigned to the system area may be necessary if these components are to be installed.

Motivation: The assignment of privileges for application objects to "PUBLIC" violates the need--to-know principle and increases the risk of misuse by approving resources and functionality which is not globally required. The authorization concept of an application can also only then be classified as reliable if the accesses to application resources or functionality are granted individually and not just assigned across the board via "PUBLIC".

Checking the rights assigned to PUBLIC can only ever be worthwhile for application objects and not for objects found in the data dictionary.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-48/6.0

Req 49	The O7_DICTIONARY_ACCESSIBILITY parameter must be set to FALSE.

Motivation: If the 07_DICTIONARY_ACCESSIBILTY initialization parameter is set to FALSE (default value since Oracle 9i), access to the dictionary objects is no longer possible using ANY privileges.

Implementation example: If access to dictionary objects (e.g., for applications and tools) is required, the administrator must individually grant object privileges for access to these dictionary objects.

The GRANT ALL OBJECT PRIVILEGES privilege does not include the option, e.g., to assign the select privilege for the dba_users dictionary view.
Oracle Corporation: Security Guide, 11g Release 2 (11.2), Document Number E16543-05, April 2011

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-49/6.0

Req 50 Non-DBA users must not have object privileges (directly or via roles) for the following objects:

- AUD$
- ALL_SOURCE
- DBA_AUDIT_TRAIL
- DBA_ROLE_PRIVS
- DBA_SYS_PRIVS
- DBA_TAB_PRIVS
- DBA_USERS
- DBMS_FILE_TRANSFER
- DBMS_IJOB
- DBMS_ISCHED
- DBMS_SYS_SQL
- DBMS_NETWORK_ACL_ADMIN
- DBA_USERS_WITH_DEFPWD
- FGA_LOG$
- LINK$
- USER$
- USER_HISTORY$

Motivation: The views and packets named either allow access to confidential information (e.g., password hash in LINK$) or the performance of potentially dangerous actions (e.g., sending e-mails from the database with UTL_MAIL).

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-50/6.0

Req 51 Non-DBA user accounts must not have one of the following roles (directly or indirectly):

- AQ_ADMINISTRATOR_ROLE
- DATAPUMP_IMP_FULL_DATABASE
- DBFS_ROLE
- DBA
- DELETE_CATALOG_ROLE
- EXECUTE_CATALOG_ROLE
- EXP_FULL_DATABASE
- GATHER_SYSTEM_STATISTICS
- IMP_FULL_DATABASE
- LOGSTDBY_ADMINISTRATOR
- OEM_MONITOR
- OEM_ADVISOR
- RECOVERY_CATALOG_OWNER
- SCHEDULER_ADMIN
- SNMPAGENT
- SELECT_CATALOG_ROLE

Motivation: These roles have critical object and/or system privileges. Thus, for example, the DATAPUMP_IMP_FULL_DATABASE role has a system privilege which enables any privilege to be assigned to anyone.

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources

For this requirement the following warranty objectives are relevant:

ID: 3.29-51/6.0