Preface

Welcome to *Oracle Database Security Guide*. This guide describes how you can configure security for Oracle Database by using the default database features.

Audience

Oracle Database Security Guide is intended for database administrators (DBAs), security administrators, application developers, and others tasked with performing the following operations securely and efficiently.

It covers these areas:

- Designing and implementing security policies to protect the data of an organization, users, and applications from accidental, inappropriate, or unauthorized actions
- Creating and enforcing policies and practices of auditing and accountability for inappropriate or unauthorized actions
- Creating, maintaining, and terminating user accounts, passwords, roles, and privileges
- Developing applications that provide desired services securely in a variety of computational models, leveraging database and directory services to maximize both efficiency and ease of use

To use this document, you need a basic understanding of how and why a database is used, and basic familiarity with SQL.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

For more security-related information, see these Oracle resources:

- *Oracle Database Data Redaction Guide*
- *Oracle Database Transparent Data Encryption Guide*
- *Oracle Database Vault Administrator's Guide*
- *Oracle Label Security Administrator's Guide*
- Oracle Key Vault documentation library
- Audit Vault and Database Firewall documentation library
- Oracle Data Masking and Subsetting documentation library
- Oracle Data Safe documentation library
- Oracle Database Security Assessment Tool
- *Oracle Database PL/SQL Packages and Types Reference*
- *Oracle Database Reference*
- *Oracle Database SQL Language Reference*
- *Oracle Database Net Services Reference*
- *Oracle Database Administrator's Guide*
- *Oracle Database Concepts*
- *Oracle Multitenant Administrator's Guide*

Many of the examples in this guide use the sample schemas of the seed PDB, which you can create when you install Oracle Database. See *Oracle Database Sample Schemas* for information about how these schemas were created and how you can use them yourself.

Oracle Technical Services

To download the product data sheet, frequently asked questions, links to the latest product documentation, product download, and other collateral, visit Oracle Technical Resources (formerly Oracle Technology Network). You must register online before using Oracle Technical Services. Registration is free and can be done at

`https://www.oracle.com/technical-resources/`

My Oracle Support

You can find information about security patches, certifications, and the support knowledge base by visiting My Oracle Support (formerly Oracle*MetaLink*) at

https://support.oracle.com

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Changes in This Release for Oracle Database Security Guide

This preface contains:

Changes in Oracle Database Security 23ai

Oracle Database Security Guide for Oracle Database 23ai has new security features.

- Transport Layer Security 1.3 Protocol Now Supported in Oracle Database
  Starting with Oracle Database 23ai, Oracle Database supports Transport Layer Security (TLS) version 1.3, which uses newer and more secure cipher suites that improve confidentiality of data in transit.
- Simplified Transport Layer Security Configuration
  Starting with Oracle Database 23ai, the Transport Layer Security (TLS) configuration between the database client and server has been simplified yet made more secure.
- Schema Privileges to Simplify Access Control
  Starting with Oracle Database 23ai, Oracle Database supports schema privileges in addition to the existing object, system, and administrative privileges.
- Oracle SQL Firewall is Now Built into Oracle Database
  Starting with Oracle Database 23ai, Oracle SQL Firewall inspects all incoming SQL statements and ensures that only explicitly authorized SQL is run.
- Increased Maximum Password Length
  Starting with Oracle Database 23ai, Oracle Database supports passwords up to 1024 bytes in length.
- Read-Only Users and Sessions
  Starting with Oracle Database 23ai, you can control whether a user or session is enabled for read-write operations, irrespective of the privileges of the user that is connected to the database.
- New Database Role for Application Developers
  Starting with Oracle Database 23ai, a new role specifically for application developers, `DB_DEVELOPER_ROLE`, is introduced for stronger security using the least privilege principle.
- Oracle Data Dictionary Protection Extended to Non-SYS Oracle Schemas with Separation of Duties Protection
  Starting with Oracle Database 23ai, Oracle Database schemas now can have data dictionary protection with additional separation of duties protection for the `SYSBACKUP`, `SYSKM`, `SYSRAC`, and `SYSDG` schemas.
- Strict DN Matching with Both Listener and Server Certificates
  Starting with Oracle Database release 23ai, the behavior of the `SSL_SERVER_DN_MATCH` parameter has changed.

- Ability to Configure Transport Layer Security Connections without Client Wallets
  Starting with Oracle Database 23ai, for Linux, non-Linux, and Microsoft Windows platforms, an Oracle Database client is no longer required to provide a wallet to hold well-known CA root certificates if they are available in the local system.
- Updated Kerberos Library and Other Improvements
  Starting with Oracle Database 23ai, Oracle Database supports MIT Kerberos library version 1.21.2, and provides cross-domain support for accessing resources in other domains.
- Improved and More Secure Local Auto-Login Wallets
  Starting with Oracle Database 23ai, newly created local auto-login wallets (or pre-release 23ai wallets that have been updated for release 23ai) are more secure.
- New sqlnet.ora Parameter to Prevent the Use of Deprecated Ciphers
  Starting with Oracle Database 23ai, you can block the use of deprecated ciphers by setting the `SSL_ENABLE_WEAK_CIPHERS` `sqlnet.ora` parameter to `FALSE`.
- Enhancements to RADIUS Configuration
  Starting with Oracle Database 23ai, Oracle Database supports the Requests for Comments (RFC) 6613 and 6614 guidelines, and updates to RADIUS security with the latest standards.
- Enhancements to the DBMS_CRYPTO PL/SQL Package
  Starting with Oracle Database 23ai, the `DBMS_CRYPTO` PL/SQL package has APIs to support several customer needs, such as elliptic-curve Diffie–Hellman (ECDH) operations, updated signature and verification algorithms, and other enhancements.
- Authenticating and Authorizing IAM Users to Oracle Autonomous Database on Dedicated Exadata Infrastructure
  Starting with Oracle Database 23ai, users can authenticate and authorize IAM users to Oracle Autonomous Database on Dedicated Exadata Infrastructure.
- Ability of Azure Users to Log in to Oracle Database with Their Azure AD OAAuth2 Access Token
  Available initially for the Oracle Autonomous Database in June 2022, Microsoft Azure Active Directory (Azure AD) users can now log in to Oracle Databases on-premises and in the cloud.
- Ability to Audit Object Actions at the Column Level for Tables and Views
  Starting with Oracle Database 23ai, you can create unified audit policies to audit individual columns in tables and views.
- Consolidation of the FIPS_140 Parameter
  Starting with Oracle Database 23ai, you can use the `FIPS_140` parameter to configure FIPS in a uniform way with multiple Oracle Database environments and features.
- Desupport of Case Insensitive Passwords
  Starting with Oracle Database 23ai, case-insensitive passwords are no longer supported.
- Desupport of Traditional Auditing
  Starting with Oracle Database 23ai, traditional auditing is desupported.

Transport Layer Security 1.3 Protocol Now Supported in Oracle Database

Starting with Oracle Database 23ai, Oracle Database supports Transport Layer Security (TLS) version 1.3, which uses newer and more secure cipher suites that improve confidentiality of data in transit.

Because TLS 1.3 handles initial session setup more efficiently than earlier TLS versions, users moving to TLS 1.3 will see improvements in TLS performance. TLS 1.3 also implements

newer, more secure cipher suites that improve confidentiality of data in transit. Oracle recommends that you move immediately from the desupport TLS protocol versions (1.0 and 1.1) to version 1.3. Version 1.2 is still supported.

Related Topics

- Configuring Transport Layer Security Encryption
  Use Transport Layer Security (TLS), a secure and standard protocol, to encrypt your database client and server connections.
- Migrating to and Configuring Transport Layer Security Version 1.3
  Version 1.3 of Transport Layer Security (TLS) provides stronger security and faster TLS handshakes, when compared to previous versions of TLS.

Simplified Transport Layer Security Configuration

Starting with Oracle Database 23ai, the Transport Layer Security (TLS) configuration between the database client and server has been simplified yet made more secure.

The changes are as follows:

- Update to the default for the client `WALLET_LOCATION` parameter so that if it is not set, then the value of the `TNS_ADMIN` parameter is used instead.
- Update to the `SSL_VERSION` parameter so that it can accept a comma-separated list of strings such as `(TLSv1.3, TLSv1.2)`.
- Introduction of the `ALLOWED_WEAK_CERT_ALGORITHMS` parameter for users whose environments still require the use of the earlier certificate signature algorithms. This parameter replaces the `ALLOW_MD5_CERTS` and `ALLOW_SHA1_CERTS` parameters. If `ALLOWED_WEAK_CERT_ALGORITHMS` is set, then Oracle Database ignores `ALLOW_MD5_CERTS` and `ALLOW_SHA1_CERTS`. However, if `ALLOWED_WEAK_CERT_ALGORITHMS` is not set, then Oracle Database checks and uses the `ALLOW_MD5_CERTS` and `ALLOW_SHA1_CERTS` settings. By default, SHA1 certificate are allowed and MD5 certificates are disallowed.
- Deprecation of the following parameters:
  - `ADD_SSLV3_TO_DEFAULT`
  - `ALLOW_MD5_CERTS`
  - `ALLOW_SHA1_CERTS`
- Modifications to how wallets are loaded
  - Server-side wallets: The `WALLET_LOCATION` parameter for server-side wallets is deprecated. Instead, use the `WALLET_ROOT` initialization parameter in the `init.ora` file.
  - Client-side wallets: The `WALLET_LOCATION` parameter can still be used for client-side wallets.
- Improved performance for the processing of wallets
- For users to enable TLS between the database client and the server, the only required and minimum configuration is putting a pair of wallets in client side `TNS_ADMIN` directory, and server side `WALLET_ROOT` directory.

Related Topics

- *Oracle Database Net Services Reference*

Schema Privileges to Simplify Access Control

Starting with Oracle Database 23ai, Oracle Database supports schema privileges in addition to the existing object, system, and administrative privileges.

The following new system privileges are required if you plan to manage the security policies for row level security, fine-grained auditing, or Oracle Data Redaction. They can be granted to enable the security policy across all non-`SYS` schemas in the database or to restrict the security policy to one schema.

- `ADMINISTER ROW LEVEL SECURITY POLICY`, for when the `DBMS_RLS` package is used for row level security policies
- `ADMINISTER FINE GRAINED AUDIT POLICY`, for when the `DBMS_FGA` package is used for fine-grained audit policies
- `ADMINISTER REDACT POLICY`, for when the `DBMS_REDACT` package is used for data redaction policies

As part of this new feature, the following views are introduced:

- `DBA_SCHEMA_PRIVS`
- `ROLE_SCHEMA_PRIVS`
- `USER_SCHEMA_PRIVS`
- `SESSION_SCHEMA_PRIVS`
- `V$ENABLEDSCHEMAPRIVS`

In previous releases, object privileges provided fine-grained control over access to individual objects, such as the `HR.EMPLOYEES` table. System privileges were designed for administrators to grant similar access to all objects in the database of a certain type (for example, the `SELECT ANY TABLE` system privilege). For applications that only need to provide enough privileges (least privilege principle) for users to application objects, every privilege for every object had to granted and tracked. Hence, new objects in the same schema required new object privileges. With the new schema privileges, you can grant a privilege for the entire schema, thereby simplifying application authorizations and improving security. For example:

```
GRANT SELECT ANY TABLE ON SCHEMA HR TO SCOTT;
```

Related Topics

- Managing Schema Privileges
  Schema privileges enable certain system privileges to be granted on a schema.
- Administering Schema Security Policies
  To manage schema security policies for row level security, fine-grained auditing, and Oracle Data Redaction, users must be granted the appropriate system privilege.

Oracle SQL Firewall is Now Built into Oracle Database

Starting with Oracle Database 23ai, Oracle SQL Firewall inspects all incoming SQL statements and ensures that only explicitly authorized SQL is run.

You can use SQL Firewall to control which SQL statements are allowed to be processed by the database. You can restrict connection paths associated with database connections and SQL statements. Unauthorized SQL can be logged and blocked.

Because SQL Firewall is built into the Oracle database, it cannot be bypassed. All SQL statements are inspected, whether local or network, or encrypted or clear text. It examines top-level SQL, stored procedures and the related database objects.

SQL Firewall provides real-time protection against common database attacks by restricting database access to only authorized SQL statements or connections. It mitigates risks from SQL injection attacks, anomalous access, and credential theft or abuse.

SQL Firewall uses session context data such as IP address, operating system user name, and operating system program name to restrict how a database account can connect to the database. This helps mitigate the risk of stolen or misused application service account credentials. A typical use case for SQL Firewall is for application workloads.

You can use SQL Firewall in both the root and a pluggable database (PDB).

Related Topics

-

Increased Maximum Password Length

Starting with Oracle Database 23ai, Oracle Database supports passwords up to 1024 bytes in length.

In previous releases, the Oracle Database password length and the secure role password length could be up to 30 bytes. The increased maximum password length to 1024 bytes provides the following benefits:

- It accommodates passwords that are used by Oracle Identity Cloud Service (IDCS) and Identity Access Management (IAM). The increase to 1024 bytes enables uniform password rules for all Cloud deployments.
- The 30-byte limitation was too restrictive when password multi-byte characters used more than 1 byte in an NLS configuration.

Related Topics

- Minimum Requirements for Passwords
  Oracle provides a set of minimum requirements for passwords.

Read-Only Users and Sessions

Starting with Oracle Database 23ai, you can control whether a user or session is enabled for read-write operations, irrespective of the privileges of the user that is connected to the database.

The `READ_ONLY` session applies to any type of user for any type of container. The `READ_ONLY` user only applies to local users.

Providing the capability to disable and re-enable the read-write capabilities of any user or session without revoking and re-granting privileges provides you with more flexibility to temporarily control the privileges of users or sessions for testing, administration, or application development purposes. It also gives you a simple way to control the read-write behavior within different parts of an application that are used by the same user or session.

Related Topics

- Configuring Read-Only Users
  You can override the privileges and roles that have been granted to a user by making the user a read-only user.

- *Oracle Multitenant Administrator's Guide*

New Database Role for Application Developers

Starting with Oracle Database 23ai, a new role specifically for application developers, `DB_DEVELOPER_ROLE`, is introduced for stronger security using the least privilege principle.

Oracle Database has many distinct privileges that can be granted to schema users or roles, as well as numerous stored or built-in PL/SQL packages that can be executed. Developers who design, develop, and deploy an application need a subset of these. Because an application developer or owner may not know or understand all the privileges that are needed by application developers, this could potentially result in database administrators granting all-encompassing privileges to developers. Providing developers with more privileges than necessary could pose a potential security risk. An alternative to granting all-encompassing privileges is to selectively grant privileges on demand as the application developer identifies the privileges they require that are not currently granted.

The benefit of the `DB_DEVELOPER_ROLE` role is that it quickly and easily provides the application developer with only the privileges that they need to design, implement, and deploy applications on Oracle databases.

Related Topics

- Use of the DB_DEVELOPER_ROLE Role for Application Developers
  The `DB_DEVELOPER_ROLE` role provides most of the system privileges, object privileges, predefined roles, PL/SQL package privileges, and tracing privileges that an application developer needs.

Oracle Data Dictionary Protection Extended to Non-SYS Oracle Schemas with Separation of Duties Protection

Starting with Oracle Database 23ai, Oracle Database schemas now can have data dictionary protection with additional separation of duties protection for the `SYSBACKUP`, `SYSKM`, `SYSRAC`, and `SYSDG` schemas.

Dictionary protection has been applied to Oracle schemas such as `AUDSYS` and `LBACSYS`. For the full list of dictionary protected Oracle schemas, run the following query:

```
SELECT USERNAME, DICTIONARY_PROTECTED FROM DBA_USERS WHERE
DICTIONARY_PROTECTED='YES';
```

The dictionary protection includes the underlying schemas for the `SYSDBA`, `SYSBACKUP`, `SYSKM`, `SYSRAC`, and `SYSDG` administrative privileges. These have additional separation of duties protections. Direct and proxy logins are blocked and password changes are restricted to the user only.

Oracle schemas provide critical functionality for Oracle Database features. By enabling these schemas to have dictionary protection, you can prevent inadvertent and malicious changes within these schemas that could endanger Oracle Database functionality.

Related Topics

- Managing Dictionary Protection for Oracle-Maintained Schemas
  Oracle-maintained schemas such as `AUDSYS` have dictionary protection to prevent users from using system privileges on these schemas.

Strict DN Matching with Both Listener and Server Certificates

Starting with Oracle Database release 23ai, the behavior of the `SSL_SERVER_DN_MATCH` parameter has changed.

Previously, Oracle Database performed the DN check only with the database server certificate, and both the `HOSTNAME` and the `SERVICE_NAME` setting in the connect string could be used for a partial DN match.

With Oracle Database 23ai, Oracle Database checks both the listener and server certificates. In addition, the `SERVICE_NAME` setting in the connect string is not used to check during a partial DN match. The `HOSTNAME` setting can still be used for partial DN matching with the certificate DN and subject alternative name (SAN), on both the listener and server certificates.

When set to `TRUE`, the `SSL_ALLOW_WEAK_DN_MATCH` parameter reverts `SSL_SERVER_DN_MATCH` to the behavior earlier than release 23ai and enables DN matching to only check the database server certificate (but not the listener) and enable the service name to be used for partial DN matching.

DN matching with both the listener and server certificates provides better security to ensure that the client is connecting to the correct database server. The service name setting is also removed from `SSL_SERVER_DN_MATCH` for better security and partial DN matching can still be performed with the `HOSTNAME` connect string parameter with the certificate DN and subject alternative name (SAN) matching.

The `SSL_ALLOW_WEAK_DN_MATCH`, though new to this release, is marked as deprecated because it is a temporary mechanism to enable interoperability with releases prior to 23ai.

Related Topics

- Enable Weak DN Matching
  The `SSL_ALLOW_WEAK_DN_MATCH` parameter control reverts the DN matching behavior to prior database versions.
- Enabling Distinguished Name (DN) Matching
  DN matching allows a connection to the Oracle Database server when the server certificate name or DN matches what the client expects.

Ability to Configure Transport Layer Security Connections without Client Wallets

Starting with Oracle Database 23ai, for Linux, non-Linux, and Microsoft Windows platforms, an Oracle Database client is no longer required to provide a wallet to hold well-known CA root certificates if they are available in the local system.

The Oracle Database wallet search order determines the location (Windows (Microsoft Certificate Store) or Linux) of these certificates in the local system.

Transport Layer Security (TLS) requires either one-way authentication or two-way authentication. In one-way TLS authentication, which is commonly used for HTTPS connections, you will no longer need to install and configure a client wallet to hold the server's CA certificate as long as it is already available in the local system. If the server's CA certificate is not installed in the local systems, then client wallet is still required.

This enhancement greatly simplifies the Oracle Database client installation and the use of TLS protocol to encrypt Oracle Database client-server communications.

Related Topics

- Configuring TLS Using a Public Certificate Authority Root of Trust for the Database Server Certificate
  Before you can configure TLS without using client wallets, you must first create the server wallet and ensure that the database and listener are properly configured.
- Oracle Database Wallet Search Order
  The search order that Oracle Database uses to find wallets depends on the feature for which the wallet was created, such as Transparent Data Encryption (TDE).

Updated Kerberos Library and Other Improvements

Starting with Oracle Database 23ai, Oracle Database supports MIT Kerberos library version 1.21.2, and provides cross-domain support for accessing resources in other domains.

This Kerberos enhancement improves security and allows Kerberos to be used in more Oracle Database environments.

Related Topics

- Configuring Kerberos Authentication
  Kerberos is a trusted third-party authentication system that relies on shared secrets and presumes that the third party is secure.

Improved and More Secure Local Auto-Login Wallets

Starting with Oracle Database 23ai, newly created local auto-login wallets (or pre-release 23ai wallets that have been updated for release 23ai) are more secure.

A local auto-login wallet is now more tightly bound to the host where it was created or modified. The local auto-login process is also more secure, does not require additional deployment requirements, and does not require root access.

Local auto-login wallets are more secure now and support both bare metal and virtual environments.

This enhancement also applies to Tranparent Data Encryption (TDE) local auto-login keystores.

Related Topics

- About Managing Oracle Database Wallets and Certificates with the orapki Utility
  The `orapki` command-line utility enables you to create and manage wallets and certificates from the command line.

New sqlnet.ora Parameter to Prevent the Use of Deprecated Ciphers

Starting with Oracle Database 23ai, you can block the use of deprecated ciphers by setting the `SSL_ENABLE_WEAK_CIPHERS` `sqlnet.ora` parameter to `FALSE`.

You can prevent the use of deprecated ciphers, which are less secure than the latest ciphers, in an Oracle database if you do not have a dependency on them. This simplifies the passing of compliance audits and improves the overall security of your database.

Related Topics

- Enabling Weak Cipher Suites
  You can enable deprecated cipher suites by setting the `SSL_ENABLE_WEAK_CIPHERS` parameter. For the connections to be successful with the weak cipher suites, all three components (client, listener, and server) need to have the weak cipher suites enabled.
- Specifying TLS Protocol and TLS Cipher Suites
  Oracle Database 23ai supports TLS protocol versions 1.2 and 1.3 and their associated cipher suites for Transport Layer Security (TLS).

Enhancements to RADIUS Configuration

Starting with Oracle Database 23ai, Oracle Database supports the Requests for Comments (RFC) 6613 and 6614 guidelines, and updates to RADIUS security with the latest standards.

This enhancement introduces the following new RADIUS-related `sqlnet.ora` parameters:

- `SQLNET.RADIUS_ALTERNATE_TLS_HOST`
- `SQLNET.RADIUS_ALTERNATE_TLS_PORT`
- `SQLNET.RADIUS_AUTHENTICATION_TLS_HOST`
- `SQLNET.RADIUS_AUTHENTICATION_TLS_PORT`
- `SQLNET.RADIUS_TRANSPORT_PROTOCOL`

The following existing RADIUS `sqlnet.ora` parameters have been updated:

- `SQLNET.RADIUS_ALTERNATE_PORT`
- `SQLNET.RADIUS_AUTHENTICATION_PORT`
- `SQLNET.RADIUS_SECRET`

The older RADIUS standards are blocked by default in Oracle Database 23ai. If you need to enable pre-release 23ai clients to connect using the older protocol, then set one or both of the following parameters, new to release 23ai, in the `sqlnet.ora` file.

- `SQLNET.RADIUS_ALLOW_WEAK_CLIENTS` enables pre-release 23ai database clients to connect RADIUS users using the older standard.
- `SQLNET.RADIUS_ALLOW_WEAK_PROTOCOL` enables the pre-release 23ai database server to connect to the RADIUS server using the older standard.

This enhancement is beneficial in that Oracle Database RADIUS API implements TCP over Transport Layer Security (TLS) and provides other security improvements, such as support for AES256 and SHA512.

Related Topics

- About Configuring RADIUS Authentication
  Oracle Database supports the RADIUS standard for user authentication.
- Enabling RADIUS Authentication, Authorization, and Accounting
  You can enable RADIUS authentication, authorization, and accounting from the command line.
- *Oracle Database Upgrade Guide*
- *Oracle Database Upgrade Guide*

Enhancements to the DBMS_CRYPTO PL/SQL Package

Starting with Oracle Database 23ai, the `DBMS_CRYPTO` PL/SQL package has APIs to support several customer needs, such as elliptic-curve Diffie–Hellman (ECDH) operations, updated signature and verification algorithms, and other enhancements.

These enhancements are as follows:

- New APIs for elliptic-curve Diffie–Hellman (ECDH) operations
  - `ECDH_GENKEYPAIR`: This function generates an EC public/private key pair
  - `ECDHDERIVE_SHAREDSECRET`: This function derives shared secret using private key of local application and public key from the remote application.
- New `PKENCRYPT`/`PKDECRYPT` algorithm: `PKENCRYPT_RSA_PKCS1_OAEP_SHA2`
- New chain modes `GCM`, `CCM`, and `XTS`
- New `DBMS_CRYPTO` block cipher suites `AES_CCM_NONE` and `AES_GCM_NONE`
- New signature and verification algorithms:
  - `SIGN_SHA224_ECDSA`
  - `SIGN_SHA256_ECDSA`
  - `SIGN_SHA384_ECDSA`
  - `SIGN_SHA512_ECDSA`
  - `SIGN_ECDSA`

Related Topics

- On-Demand Encryption of Data
  You can use the `DBMS_CRYPTO` PL/SQL package to perform on-demand encryption of data.
- *Oracle Database PL/SQL Packages and Types Reference*

Authenticating and Authorizing IAM Users to Oracle Autonomous Database on Dedicated Exadata Infrastructure

Starting with Oracle Database 23ai, users can authenticate and authorize IAM users to Oracle Autonomous Database on Dedicated Exadata Infrastructure.

Additional enhancements are as follows:

- Applications can now connect to an Autonomous Database instance by using end-user, instance, and resource principals.
- IAM users can now proxy to an Autonomous Database by using a database user schema.
- Database links are supported for IAM connections.

Related Topics

- Authenticating and Authorizing IAM Users for Oracle DBaaS Databases
  Identity and Access Management (IAM) users can be configured to connect to an Oracle Database as a service (Oracle DBaaS) instance.

Ability of Azure Users to Log in to Oracle Database with Their Azure AD OAAuth2 Access Token

Available initially for the Oracle Autonomous Database in June 2022, Microsoft Azure Active Directory (Azure AD) users can now log in to Oracle Databases on-premises and in the cloud.

You can use Azure AD `OAuth2` tokens to access the database. Azure AD users can access the database directly using their Azure AD token, and applications can use their service tokens to access the database.

Related Topics

- Authenticating and Authorizing Microsoft Azure Users for Oracle Databases
  An Oracle database can be configured for Microsoft Azure users of Microsoft Entra ID (previously called Microsoft Azure AD) to connect using single sign-on authentication.

Ability to Audit Object Actions at the Column Level for Tables and Views

Starting with Oracle Database 23ai, you can create unified audit policies to audit individual columns in tables and views.

The `ACTIONS` clause of the `CREATE AUDIT POLICY` and `ALTER AUDIT POLICY` procedures allows you to specify the list of columns whose access is to be audited. For example, to audit `UPDATE` statements on the `SALARY` column of a table, you would specify `ACTIONS UPDATE(SALARY)`.

The feature enables you to configure more granular and focused audit policies, and ensures that auditing is selective enough to reduce the creation of unnecessary audit records, and effective enough to let you meet your compliance requirements.

Related Topics

- Example: Auditing an Action on a Table Column
  The `CREATE AUDIT POLICY` statement can audit actions on table or view columns.
- Object Actions That Can Be Audited
  Auditing object actions can be broad or focused (for example, auditing all user actions or only a select list of user actions).

Consolidation of the FIPS_140 Parameter

Starting with Oracle Database 23ai, you can use the `FIPS_140` parameter to configure FIPS in a uniform way with multiple Oracle Database environments and features.

These environments and features are as follows:

- Transparent Data Encryption (TDE)
- `DBMS_CRYPTO` PL/SQL package
- Transport Layer Security (TLS)
- Network native encryption

You can still use the legacy FIPS 140-2 configurations for these environments, but Oracle recommends that you use the consolidated `FIPS_140` parameter instead.

Related Topics

- Configuration of FIPS 140-2 Using the Consolidated FIPS_140 Parameter
  The consolidated `FIPS_140` parameter can be set for several different Oracle Database environments.

Desupport of Case Insensitive Passwords

Starting with Oracle Database 23ai, case-insensitive passwords are no longer supported.

Users whose passwords are case-insensitive will be unable to log in to the database after upgrading to Oracle Database 23ai. Before upgrading, an administrator must use the following query to find the users whose passwords are case-insensitive and notify these users to change their passwords:

```
SELECT USERNAME FROM DBA_USERS
WHERE (PASSWORD_VERSIONS = '10G '
OR PASSWORD_VERSIONS = '10G HTTP ')
AND USERNAME <> 'ANONYMOUS';
```

Changing the password enables the use of later, more secure password versions. If you have already upgraded to release 23ai and still have users whose passwords are case insensitive, then these users will not be able to log in. An administrator will need to change the password for these users. The password of any user that has only the `10G` password version remains case insensitive until it is changed, and it becomes case sensitive after it is changed.

Related Topics

- Finding and Resetting User Passwords That Use the 10G Password Version
  For better security, find and reset passwords for user accounts that use the `10G` password version so that they use later, more secure password versions.

Desupport of Traditional Auditing

Starting with Oracle Database 23ai, traditional auditing is desupported.

Unified auditing is the way forward to perform Oracle Database auditing. Unified auditing offers more flexibility to perform selective and effective auditing, which helps you focus on activities that really matter to your enterprise. Unified auditing has one single and secure unified trail, conditional policy for audit selectivity, and default predefined policies for simplicity. To improve security and compliance, Oracle strongly recommends that you use unified auditing.

Related Topics

- Handling the Desupport of Traditional Auditing
  Traditional auditing is desupported, starting in Oracle Database 23ai. Oracle recommends that you use unified auditing instead.

Updates to Oracle Database Security 23ai

Oracle Database Security Guide for Oracle Database 23ai has updates.

- New Procedure for Oracle SQL Firewall DBMS_SQL_FIREWALL PL/SQL Package
  The Oracle SQL Firewall package `DBMS_SQL_FIREWALL` now has an additional procedure, `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL`.

- DBMS_CRYPTO Support for SM2, SM3, SM4, and SHA-3 Cryptographic Algorithms
  The `DBMS_CRYPTO` PL/SQL package now supports the use of SM2, SM3, SM4, and SHA-3 cryptographic algorithms.
- orapki Enhancements
  The `orapki` command line utility has been enhanced to include `mkstore` features and new command parameters to specify wallet certificates and keys.
- Microsoft Entra ID (Azure AD) Integration Enhancements
  Oarcle Cloud Infrastructure (OCI) and Oracle Database Instant Client now can directly retrieve Microsoft Entra ID (Azure AD) `OAuth2` tokens. In addition, the Oracle Database server on AIX, Solaris, and HPUX platforms support the Entra ID integration.

New Procedure for Oracle SQL Firewall DBMS_SQL_FIREWALL PL/SQL Package

The Oracle SQL Firewall package `DBMS_SQL_FIREWALL` now has an additional procedure, `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST_SINGLE_SQL`.

This procedure enables you to individually append specific SQL records from a capture log or a violation log to an existing allow-list. While `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST` provides the flexibility to append the entire violation or capture log to the allow-list, in most common scenarios you might also need the flexibility to add just one of them instead of the entire list. In previous releases, if you wanted to append specific SQL commands to an allow-list, you had to use `DBMS_SQL_FIREWALL.APPEND_ALLOW_LIST` to append the entire violation or capture log to the allow-list, and then use `DBMS_SQL_FIREWALL.DELETE_ALLOWED_LIST` to manually delete the unwanted entries. This enhancement gives more flexibility to adjust the allow-list with specific records that you want to include.

Related Topics

-
- *Oracle Database PL/SQL Packages and Types Reference*

DBMS_CRYPTO Support for SM2, SM3, SM4, and SHA-3 Cryptographic Algorithms

The `DBMS_CRYPTO` PL/SQL package now supports the use of SM2, SM3, SM4, and SHA-3 cryptographic algorithms.

- SM2 is an asymmetric cryptographic algorithm. It is deployed for digital signatures, key exchange, and encryption.
- SM3 is a 256-bit hash algorithm. It is used for digital signatures, message authentication codes, and pseudorandom number generators.
- SM4 is a block symmetric encryption algorithm.
- SHA-3 (Secure Hash Algorithm 3) is a new cryptographic hash algorithm that supports fixed length hash, variable length hash, sign, verify, Hash-based Message Authentication Code (HMAC), and KECCAK Message Authentication Code (KMAC) functionalities.

The following `DBMS_CRYPTO` functions have been enhanced to support to the new algorithm constants:

- `DBMS_CRYPTO.ENCRYPT`
- `DBMS_CRYPTO.DECRYPT`

- DBMS_CRYPTO.HASH
- DBMS_CRYPTO.MAC
- DBMS_CRYPTO.PKENCRYPT
- DBMS_CRYPTO.PKDECRYPT
- DBMS_CRYPTO.SIGN
- DBMS_CRYPTO.VERIFY

The following DBMS_CRYPTO functions have been added to support new algorithm constants for some SHA-3 features:

- DBMS_CRYPTO.HASH_LEN (similar to the existing DBMS_CRYPTO.HASH function but it includes an extra input length)
- DBMS_CRYPTO.KMACXOF (similar to the existing DBMS_CRYPTO.MAC function but it includes an extra input length and custom string)

This new hash type can be used with DBMS_CRYPTO.ENCRYPT and DBMS_CRYPTO.DECRYPT:

- ENCRYPT_SM4

These new hash types can be used with DBMS_CRYPTO.HASH:

- HASH_SHA3_224
- HASH_SHA3_256
- HASH_SHA3_384
- HASH_SHA3_512
- HASH_SM3

These new MAC types can be used with the DBMS_CRYPTO.MAC function:

- HMAC_SHA3_224
- HMAC_SHA3_256
- HMAC_SHA3_384
- HMAC_SHA3_512

These new encryption types can be used with DBMS_CRYPTO.PKENCRYPT and DBMS_CRYPTO.PKDECRYPT:

- PKENCRYPT_SM2
- KEY_TYPE_SM2

These new algorithms can be used with DBMS_CRYPTO.SIGN and DBMS_CRYPTO.VERIFY:

- SIGN_SHA3_224_RSA
- SIGN_SHA3_256_RSA
- SIGN_SHA3_384_RSA
- SIGN_SHA3_512_RSA
- SIGN_SHA3_224_ECDSA
- SIGN_SHA3_256_ECDSA
- SIGN_SHA3_384_ECDSA

- SIGN_SHA3_512_ECDSA
- SIGN_SM3_SM2

SHA-3 provides variable-length hash functions, allowing for hash values of any desired length.

These new variable length hash types can be used with the new DBMS_CRYPTO.HASH_LEN function:

- HASH_SHAKE128
- HASH_SHAKE256

These new variable length MAC types can be used with the new DBMS_CRYPTO.KMACXOF function:

- KMACXOF_128
- KMACXOF_256

Related Topics

- *Oracle Database PL/SQL Packages and Types Reference*

orapki Enhancements

The orapki command line utility has been enhanced to include mkstore features and new command parameters to specify wallet certificates and keys.

- **mkstore features included in orapki:** mkstore features have been incorporated into the orapki command line utility to simplify the management of Oracle Database wallets, certificates, and secrets. The new commands in orapki support the following capabilities of mkstore:
  - The ability to create, modify and delete secret store credentials and entries
  - The ability to list specific secret store credentials and entries

  These capabilities are supported with the orapki secretstore command.

  The mkstore utility has been deprecated. Oracle recommends that you use orapki instead.
- **New command parameters to specify wallet certificates and keys:** The orapkicommand-line utility now enables you to store alias names in an Oracle wallet and also display and reference certificate thumbprint signatures in an Oracle wallet. These enhancements enable users to do the following:
  - Use thumbprint or alias to select the certificate in a connect string for TLS connections.
  - Use thumbprint or alias to select the certificate in the Microsoft Certificate Store (MCS) for TLS connections.
  - Store certificates with their serial numbers to simplify specifying certificates or removing certificates.

  This enhancement affects the orapki wallet add, orapki wallet display, and orapki wallet remove commands. The benefit of this feature is the simplification of managing wallets and selecting certificates through the new thumbprint, alias, and serial number parameters.

Related Topics

- orapki Utility Commands Summary
  The `orapki` commands perform a variety of wallet, certificate revocation lists (CRL), and certificate management tasks.

Microsoft Entra ID (Azure AD) Integration Enhancements

Oarcle Cloud Infrastructure (OCI) and Oracle Database Instant Client now can directly retrieve Microsoft Entra ID (Azure AD) `OAuth2` tokens. In addition, the Oracle Database server on AIX, Solaris, and HPUX platforms support the Entra ID integration.

Microsoft has renamed Azure AD to Entra ID. This terminology will be used in Oracle Database 23ai and later releases.

- **OCI and Instant Client now can directly retrieve Entra ID OAuth2 tokens.** Oracle Call Interface (OCI) and Oracle Database Instant Client can retrieve a Microsoft Entra ID `OAuth2` token directly from Entra ID instead of relying on a separate script or process to retrieve the token first. This design improves the interactive flow between the database server and the client when users connect to the database (for example, with SQL*Plus). This enhancement simplifies the configuration that an end-user must perform in order to retrieve tokens. In previous releases, the end-user had to run a script to get the token from Entra ID before starting SQL*Plus or any other OCI utilities. Now, the token retrieval is part of OCI. This enhancement is similar to recent enhancements with the JDBC-thin and ODP.NET core and managed clients.
- **The Entra ID Integration is now supported with the Oracle Database server running on the AIX, Solaris, and HPUX platforms.** Entra ID integration is now available for the Oracle Database server on all supported operating system platforms. In addition to the newly supported AIX, Solaris, and HPUX platforms, Linux and Windows are still supported. The Entra ID integration feature for the Oracle Database is supported on Windows and Linux only with the full (thick) client and the instant client.

Related Topics

- About Configuring Client Connections to Entra ID
  There are three different ways for an Oracle Database client to use an Entra ID `OAuth2` token to send to the database for access.
- About Integrating Oracle Database with Microsoft Entra ID
  Oracle Database and Microsoft Entra ID can be configured to allow users and applications to connect to the database using their Entra ID credentials.

1

Introduction to Oracle Database Security

Oracle Database provides a rich set of default security features to manage user accounts, authentication, privileges, application security, encryption, network traffic, and auditing.

It is important to secure data to help protect sensitive information from access and interception by unauthorized parties. Without the appropriate security measures in place, data can be vulnerable to many types of attack vectors, such as man-in-the-middle attacks, packet sniffing, or data tampering. Leadership across various lines of business such as, technology, information security, and legal and compliance tend to be concerned about data breaches for three reasons:

1. Since data and data-driven information elements are critical assets in a digital economy, safeguarding this asset set is paramount to staying competitive.
2. The bad press associated with data breaches does more intangible damage than direct financial damages in the form of fines, penalties, and retribution costs. Lingering impacts include missed new revenue opportunities, pipeline conversion rate drops, failed cost avoidance measures, and so on.
3. They need to comply with the requirements of national and state laws, industry regulations, contractual agreements, and organizational policies.

1.1 About Oracle Database Security

Use Oracle Database's security features to reduce risk and protect data from theft, destruction, or misuse.

A few popular areas to focus security efforts on include:

- **User accounts.** When a schema is created, it comes with a local database user account that has privileges in that schema. When you create user accounts, you can secure them in a variety of ways. You can also create password profiles and resource limits to better secure password policies for your site. Oracle Database provides a set of predefined schemas that provide database functionality and other predefined schemas with administrative privileges.

  For more information see Managing Security for Oracle Database Users.
- **Authentication methods.** Oracle Database provides several ways to configure authentication for users and database administrators. For example, you can authenticate users on the database level, from the operating system, and on the network, and for multitier, global users, and application servers. If you use Microsoft Active Directory, you can authenticate and authorize Microsoft Active Directory users with the database directly.

You can configure your databases to use strong authentication with Oracle authentication adapters that support various third-party authentication services with digital certificates. Oracle Database provides the following strong authentication support:

  - Centralized authentication and single sign-on.
  - Kerberos
  - Remote Authentication Dial-in User Service (RADIUS)
  - Certificate-based authentication

  For more information see Configuring Authentication and Configuring Centrally Managed Users with Microsoft Active Directory.

- **Privileges and roles.** You can use privileges and roles to restrict user access to data in the following ways:
  - Creating and granting privileges and roles to users or other roles.
    For more information see Configuring Privilege and Role Authorization.
  - Performing privilege analysis to find information about how privileges are used in your site
    For more information see Performing Privilege Analysis to Identify Privilege Use.
  - Configure definer's rights and invoker's rights for your applications
    For more information see Managing Security for Definer's Rights and Invoker's Rights.
  - Manage fine-grained access in PL/SQL packages and types
    For more information see Managing Fine-Grained Access in PL/SQL Packages and Types.
  - Use Enterprise Manager to manage security
    For more information see Managing Security for a Multitenant Environment in Enterprise Manager.
- **Application security.** The first step to creating a database application is to ensure that it you have properly incorporated application security into your application security policies.

  For more information see Managing Security for Application Developers.
- **User session information using application context.** An application context is a name-value pair that holds the session information. You can retrieve session information about a user, such as the user name or terminal, and restrict database and application access for that user based on this information.

  For more information see Using Application Contexts to Retrieve User Information.
- **Classify and protect data in different categories.** You can create Transparent Sensitive Data Protection policies to find all table columns in a database that hold sensitive data (such as credit card or Social Security numbers), classify this data, and then create a policy that protects this data as a whole for a given class.

  For more information see Using Transparent Sensitive Data Protection .
- **Network data encryption.** You can use Transport Layer Security (TLS) and native network encryption to encrypt data as it travels on the network to prevent unauthorized access to that data. You can configure native Oracle Net Services data encryption for both servers and clients.

  For more information see Configuring Oracle Database Native Network Encryption and Data Integrity and Configuring Transport Layer Security Encryption.
- **Thin JDBC client network configuration.** You can configure thin Java Database Connectivity (JDBC) clients to securely connect to Oracle databases.

- **Auditing database activities.** Auditing provides the most accurate record of any database activity, not just from connections that take place over the wire but also through direct local logins, recursive SQL, dynamic SQLs, and stored procedures. Database auditing involves creating and enabling unified audit policies to track activities such as user actions, schema changes, logon events. Unified auditing further enables you to audit selectively by adding various conditions including application context values and simple built-in functions. This helps you to reduce the volume of your audit data, and at the same time helping you detect malicious activities in a timely manner.

  For more information see Monitoring Database Activity with Auditing.

1.2 Additional Oracle Database Security Products

In addition to the security resources that are available in a default database installation, Oracle Database provides several other database security products.

These products are as follows:

- **Oracle Advanced Security** enables you to protect sensitive data by using Transparent Data Encryption and Oracle Data Redaction.
- **Oracle Label Security** applies classification labels to data, allowing you to filter user access to data at the row level.
- **Oracle Database Vault** provides fine-grained access control to your sensitive data, including protecting data from privileged users. For example, you can restrict database administrators from having access to employee information such as salaries.
- **Oracle Data Safe** enables you to analyze the sensitivity and risks of data in your Oracle databases, and based on these findings, create policies that mask sensitive data, create and monitor security controls, assess user security, and monitor user activity.
- **Oracle Enterprise User Security** enables you to manage user security at the enterprise level. Enterprise User Security (EUS) is deprecated with Oracle Database 23ai.

  Oracle recommends that you migrate to using Centrally Managed Users (CMU). This feature enables you to directly connect with Microsoft Active Directory without an intervening directory service for enterprise user authentication and authorization to the database. If your Oracle Database is in the cloud, you can also choose to move to one of the newer integrations with a cloud identity provider.
- **Oracle Enterprise Manager Data Masking and Subsetting Pack** can irreversibly replace the original sensitive data with fictitious data so that production data can be shared safely with IT developers or offshore business partners.
- **Oracle Audit Vault and Database Firewall** collects database audit data from sources such as Oracle Database audit trail tables, database operating system audit files, and database redo logs. Using Oracle Audit Vault and Database Firewall, you can create alerts on suspicious activities, and create reports on the history of privileged user changes, schema modifications, and even data-level access.
- **Oracle Key Vault** enables you to accelerate security and encryption deployments by centrally managing encryption keys, Oracle wallets, Java keystores, and credential files. It is optimized for Oracle wallets, Java keystores, and Oracle Advanced Security Transparent Data Encryption (TDE) master keys. Oracle Key Vault supports the OASIS KMIP standard. The full-stack, security-hardened software appliance uses Oracle Linux and Oracle Database technology for security, availability, and scalability, and can be deployed on your choice of compatible hardware.

In addition to these products, you can find the latest information about Oracle Database security, such as new products and important information about security patches and alerts, by visiting the Security Technology Center on Oracle Technology Network at

http://www.oracle.com/technetwork/topics/security/whatsnew/index.html