9 Security

Oracle Database 23ai is packed with new features that help you reduce risk, better secure your data, and achieve regulatory compliance objectives. From marquee new features like the new SQL Firewall through standards updates like TLS 1.3 support to small (but important) changes like increasing the maximum length of a password from 30 bytes to 1024 bytes, you'll find this newest release a step up from your older database versions.

SQL Firewall

Oracle SQL Firewall Included in Oracle Database

A new feature of Oracle Database Vault, SQL Firewall is built into Oracle Database. SQL Firewall inspects all incoming SQL statements and ensures that only explicitly authorized SQL is run. When licensed, 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 embedded in 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. Consult Table 1-11 of the *Oracle Audit Vault and Database Firewall Licensing Information* for more information on licensing requirements for SQL Firewall.

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).

View Documentation

Encryption

Transport Layer Security (TLS) 1.3 Now Supported in Oracle Database

Transport Layer Security (TLS) version 1.3 is supported in Database 23ai. TLS 1.3 is the latest and most secure TLS protocol to protect network connections to and from an Oracle database.

Because TLS 1.3 handles initial session setup more efficiently than prior TLS versions, users moving to TLS 1.3 should see improvements in TLS performance, particularly for applications that frequently connect and reconnect to the database. TLS 1.3 also implements newer, more secure cipher suites that improve confidentiality of data in transit.

View Documentation

Strict DN Matching with Both Listener and Server Certificates

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 he 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 considered a temporary solution to enable the behavior of `SSL_SERVER_DN_MATCH` prior to release 23ai.

View Documentation

Simplified Transport Layer Security Configuration

The Transport Layer Security (TLS) configuration between the database client and server has been simplified with streamlined parameters, performance improvements, and an additional parameter to find a wallet. Older TLS protocols have also been removed.

These changes improve security and make it easier to implement TLS.

View Documentation

Ability to Configure Transport Layer Security Connections Without Client Wallets

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, client wallet is still required. Starting in this release, you no longer need to install and configure a wallet to hold a well-known root certificate if it is already available in the local system.

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

View Documentation

New sqlnet.ora Parameter to Prevent the Use of Deprecated Cipher Suites

You can block the use of deprecated cipher suites by setting the `SSL_ENABLE_WEAK_CIPHERS sqlnet.ora` parameter to `FALSE`.

Removing the ability to use older, less secure cipher suites improves protection for data in-motion between the database.

View Documentation

AES-XTS Encryption Mode Support for TDE Tablespace Encryption

Transparent Database Encryption (TDE) tablespace encryption now supports Advanced Encryption Standard (AES) XTS (XEX-based mode with ciphertext stealing mode) in `CREATE TABLESPACE` statements. Earlier versions of Oracle Database TDE used AES-CFB cipher mode.

AES-XTS provides improved security and better performance, especially on platforms where TDE can take advantage of parallel processing and specialized instructions built into processor hardware.

View Documentation

Changes for TDE Encryption Algorithms and Modes

The default encryption algorithm for both TDE column encryption and TDE tablespace encryption is now AES256. The previous default for TDE column encryption was AES192. For TDE tablespace encryption, the default was AES128.

The decryption libraries for the GOST and SEED algorithms are deprecated. New keys cannot use these algorithms. The encryption libraries for both of these libraries are desupported.

The column encryption mode is now Galois/Counter mode (GCM) instead of cipher block chaining (CBC), and the tablespace keys are now used in tweakable block ciphertext stealing (XTS) operating mode instead of cipher feedback (CFB).

The Oracle Recovery Manager (RMAN) integrity check for column encryption keys now uses SHA512 instead of SHA1.

The keys for Oracle RMAN and column keys are now derived from SHA512/AES for key generation. In previous releases, they used SHA-1/3DES as a pseudo-random function.

These enhancements enable your Oracle Database environment to use the latest, most secure algorithms and encryption modes.

View Documentation

Improved and More Secure Local Auto-Login Wallets

A local auto-login wallet is now more tightly bound to the host where it was created or modified (both bare metal and virtual). 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 Transparent Data Encryption (TDE) local auto-login keystores.

View Documentation

Changes to DBMS_CRYPTO

The following updates have been made to the DBMS_CRYPTO package:

- Added XTS mode to AES algorithms and set it as the default mode
- Added SHA-3
- Added SM2/3/4

Customers can use the latest cryptographic features with Oracle Database.

View Documentation

New Parameter to Control the TDE Rekey Operations for Oracle Data Guard

You now can use the `DB_RECOVERY_AUTO_REKEY` initialization parameter for Oracle Data Guard environments. `DB_RECOVERY_AUTO_REKEY` controls whether an Oracle Data Guard standby database recovery operation automatically performs the corresponding tablespace rekey when it encounters a redo that says the primary database has performed a tablespace rekey operation.

This feature is useful for standby deployments with large tablespaces whose users must perform an online TDE conversion.

View Documentation

Audit

Audit Object Actions at the Column Level for Tables and Views

You can create unified audit policies to audit individual columns in tables and views.

This 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.

View Documentation

Control Authorizations for Unified Auditing and Traditional Auditing

You can control how privileged users can grant and revoke the Oracle Database `AUDIT_ADMIN` and `AUDIT_VIEWER` roles by using Oracle Database Vault APIs. Database Vault blocks direct modification of the database audit tables except through the `DBMS_AUDIT_MGMT` PL/SQL package by authorized users. A new mandatory default realm (Oracle Audit Realm) protects the `AUDSYS` schema and audit-related objects in the `SYS` schema.

This new Database Vault realms simplifies auditing database vault, consolidating the privileges required for auditing into one authorization mechanism. In addition to facilitating the granting of audit-related privileges to the user, this enhancement provides greater separation of duties for managing auditing in an Oracle Database Vault environment.

View Documentation

Authentication

Microsoft Azure Active Directory Integration

You can log into Oracle Databases using your Microsoft Azure Active Directory (Azure AD) single sign-on `OAuth2` access token. This feature has been backported to Oracle Database release 19.16 and later, but not for Oracle Database 21c.

New features for Oracle Database 23ai include support for Azure AD v2 tokens and retrieving the tokens directly with the Oracle Database clients. Use of scripts to retrieve tokens for end-users will not be necessary when using the OAuth2 interactive flow.

This multi-cloud feature integrates authentication and authorization between Azure AD and Oracle Databases.

View Documentation

ODP.NET: Azure Active Directory Single Sign-On

ODP.NET can log into Oracle databases using a Microsoft Azure Active Directory (Azure AD) OAuth 2.0 access token. Users can sign-on once with Azure AD, acquire the token, and access their on-premises and cloud-based Oracle databases. This feature is available in ODP.NET Core and managed ODP.NET.

This multicloud capability eases authentication and authorization between Azure AD and Oracle Databases by simplifying user access and management.

View Documentation

Increased Oracle Database Password Length

Oracle Database now 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.

Increasing the password length supports an industry-wide trend for stronger authentication. In cases where passwords must be used, the increased length permits passwords that are more difficult to guess.

View Documentation

JDBC-Thin Support for Longer Passwords

Passwords for database user authentication can now be as long as 1024 characters.

This feature fosters increased authentication security for Java applications in Cloud and On-premises environments.

View Documentation

Oracle Data Pump Export and Import Support for Longer Encryption Passwords

Oracle Data Pump can protect export files with encryption passwords of up to 1024 bytes long.

Oracle Data Pump enhances security by supporting encryption passwords of up to 1024 bytes long.

View Documentation

Oracle Call Interface (OCI) and Oracle C++ Call Interface (OCCI) Password Length Increase

Oracle Call Interface (OCI) and Oracle C++ Call Interface (OCCI) now support passwords for database user authentication up to 1024 bytes long.

This feature allows longer passwords to be used to improve security. It also aids database use with tools that generate long passwords.

View Documentation

Updated Kerberos Library and Other Improvements

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.

View Documentation

Enhancements to RADIUS Configuration

RADIUS is frequently used to provide multi-factor authentication (MFA) for Oracle Database. Oracle Database 23ai now supports the RFC 6613 and 6614 guidelines for RADIUS and implements TCP over Transport Layer Security (TLS) by default. This enhancement introduces new RADIUS-related `sqlnet.ora` parameters to support the new standards. The enhancement also deprecates several RADIUS-related `sqlnet.ora` parameters that are no longer needed to support the new standards.

This update to RADIUS standards support improves security for customers using RADIUS-based authentication.

View Documentation

UTL_HTTP Support for SHA-256 and Other Digest Authentication Standards

UTL_HTTP is extended to support both SHA-256 and SHA-512/256 for digest authentication, to ensure forward compatibility.

UTL_HTTP can be seen as an API for client-side HTTP access, much like a standard browser. Support for both SHA-256 and SHA-512/256 for digest authentication enables UTL_HTTP to be at par with other standard browsers.

View Documentation

XDB HTTP SHA512 Digest Authentication

Oracle XDB HTTP protocol server now supports digest authentication SHA512 authentication, which is a more secure digest algorithm than MD5.

This feature improves security when using Oracle XDB from the web.

View Documentation

Ability of OCI and Instant Client to Directly Retrieve Microsoft Entra ID (Azure AD) OAuth2 Tokens

Oracle Call Interface (OCI) and Oracle Database Instant Client now can retrieve a Microsoft Entra ID (formerly Azure AD) `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.

View Documentation

Microsoft Entra ID (Azure AD) Integration Now Supported on AIX, Solaris, and HPUX

The Microsoft Entra ID (previously Azure AD) integration is now available to all Oracle Database users regardless of the server operating system platform.

In addition to the newly supported AIX, Solaris, and HPUX platforms, Linux and Windows are still supported. This feature is supported with the Oracle Cloud Infrastructure (OCI) full client and instant clients on Windows and Linux only.

View Documentation

New Parameters to Specify Wallet Certificate and Keys

The `orapki` command line utility now enables you to store alias names and thumbprint signatures in an Oracle wallet.

These enhancements enable users to do the following:

- Specify these private keys using their thumbprint or alias in a connect string.
- Use the thumbprint to specify a private key in the Microsoft Certificate Store (MCS).
- Store certificates with their serial numbers to simplify specifying certificates or removing certificates.

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

The benefit of this feature is the simplification of managing wallets and selecting certificates through new the thumbprint, alias, and serial number parameters.

View Documentation

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
- The ability to delete a wallet

The capabilities are supported with the `orapki secretstore` command.

The `mkstore` utility has been deprecated. Oracle recommends that you use `orapki` instead.

View Documentation

Authorization

Schema Privileges to Simplify Access Control

Oracle Database supports granting privileges on schemas (in addition to the existing object, system, and administrative privileges).

This feature improves security by simplifying authorization for database objects, especially for schemas that frequently add new objects. Instead of granting broad system level (* ANY) privileges that apply to the entire database, privileges can now be granted at the individual schema level.

View Documentation

Oracle Label Security Triggers Are Now Part of the New LBAC_TRIGGER Schema

A new schema, `LBAC_TRIGGER`, is introduced to own the internal triggers that were previously owned by the `LBACSYS` schema. You can migrate existing `LBACSYS` triggers to this new schema.

Both the `LBACSYS` and `LBAC_TRIGGER` schemas are Oracle-maintained and dictionary-protected.

This feature improves security when using the Oracle Label Security option.

View Documentation

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

Oracle Database schemas can have data dictionary protection with additional separation of duties protection for SYSOPER, SYSASM, SYSBACKUP, SYSKM, SYSRAC, and SYSDG.

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

View Documentation

GoldenGate Capture and Apply User Roles

New roles OGG_CAPTURE, OGG_APPLY, OGG_APPLY_PROCREP, XSTREAM_CAPTURE, XSTREAM_APPLY have been created for granting appropriate capture and apply privileges to the GoldenGate and XStream administrators. These new roles replace the functionality in the procedures of the DBMS_GOLDENGATE_AUTH and DBMS_XSTREAM_AUTH packages, which are now de-supported.

This feature simplifies administrative tasks.

View Documentation

New Utility Functions for Finding Client Host and IP Information

You can use two new Oracle Database Vault utility functions to find information about client hosts and IPs. These new utility functions are as follows:

- DBMS_MACUTL.CONTAINS_HOST
- DBMS_MACUTL.IS_CLIENT_IP_CONTAINED

These utility functions enable you to conveniently check if an IP address (or a host) is contained in a domain (or subnet range). They are useful for configuring rules and rule sets.

View Documentation

Ability to Set Tracing Using Oracle Database Vault APIs

You now can use two Oracle Database Vault APIs to control system level tracing, which applies to all database sessions. These new APIs are as follows:

- DBMS_MACADM.SET_TRACE_LEVEL
- DBMS_MACUTL.GET_TRACE_LEVEL

This enhancement enables users who have been granted the DV_ADMIN role to enable or disable tracing for all database sessions. In previous releases, this user needed the ALTER SYSTEM and the ALTER SESSION system privileges to perform this task, in addition to the DV_ADMIN role. The ALTER SYSTEM system procedure for tracing is still supported. The enhancement also provides the DBMS_MACUTL.GET_DV_TRACE_LEVEL function, which returns the trace level that has been set for the current database session. This trace level can have been set by ALTER SYSTEM, ALTER SESSION, or DBMS_MACADM.SET_DV_TRACE_LEVEL.

View Documentation

Fewer Parameters to Specify When Creating or Updating Controls

When configuring Oracle Database Vault, you may now omit parameters in the following cases:

- If you are creating a new control, omitting the parameter specifies its default value.
- If you are updating an existing control, omitting the parameter retains the current setting.

The procedures that are affected are as follows:

- DBMS_MACADM.CREATE_COMMAND_RULE
- DBMS_MACADM.CREATE_CONNECT_COMMAND_RULE
- DBMS_MACADM.CREATE_FACTOR
- DBMS_MACADM.CREATE_POLICY
- DBMS_MACADM.CREATE_REALM
- DBMS_MACADM.CREATE_RULE
- DBMS_MACADM.CREATE_RULE_SET
- DBMS_MACADM.CREATE_SESSION_EVENT_CMD_RULE
- DBMS_MACADM.CREATE_SYSTEM_EVENT_CMD_RULE
- DBMS_MACADM.UPDATE_COMMAND_RULED
- DBMS_MACADM.UPDATE_CONNECT_COMMAND_RULE
- DBMS_MACADM.UPDATE_FACTOR
- DBMS_MACADM.UPDATE_POLICY_STATE

- `DBMS_MACADM.UPDATE_REALM`
- `DBMS_MACADM.UPDATE_RULE`
- `DBMS_MACADM.UPDATE_RULE_SET`
- `DBMS_MACADM.UPDATE_SESSION_EVENT_CMD_RULE`
- `DBMS_MACADM.UPDATE_SYSTEM_EVENT_CMD_RULE`

Omitting parameters for default behaviors while creating or updating realms, rules, command rules, factors, and policies streamlines the process, allowing administrators to complete tasks more efficiently and reducing the opportunity for errors.

View Documentation

Autonomous Database

Identity and Access Management Integration with Oracle Autonomous Cloud Databases

You can now log in to additional Oracle Database Oracle Cloud Infrastructure (OCI) DBaaS platforms by using an Identity and Access Management (IAM) password or a token-based authentication. It's possible to log in to these databases by using these IAM credentials from tools, such as SQL*Plus or SQLcl.

This feature improves security through centralized management of credentials for OCI DBaaS database instances.

View Documentation

ODP.NET: Oracle Identity and Access Management

ODP.NET supports Oracle Identity and Access Management (IAM) cloud service for unified identity across Oracle cloud services, including Oracle Cloud Database Services. ODP.NET can use the same Oracle IAM credentials for authentication and authorization to the Oracle Cloud and Oracle cloud databases, now with IAM SSO tokens. This feature is available in ODP.NET Core and managed ODP.NET.

This capability allows single sign-on and for identity to be propagated to all services Oracle IAM supports including federated users via Azure Active Directory and Microsoft Active Directory (on-premises). A unified identity makes user management and account management easier for administrators and end users.

View Documentation

Oracle Client Increased Database Password Length

Starting with this release, Oracle Database and client drivers support passwords up to 1024 bytes in length.

The Oracle Database and client password length has been increased to 1024 bytes, up from 30 bytes, to allow users to set longer passwords if needed. The maximum number of characters is based on the character set used since some characters are larger than one byte.

View Documentation

Other

Secure Distributed Transaction Recovery Background Process (RECO)

Oracle Database enables queries and DMLs on objects hosted on a different database. When objects are updated on a remote database, the transaction on the source database ends up becoming a distributed transaction. If a distributed transaction fails, a database background process (RECO) periodically tries to re-establish contact, with the yet-to-be-notified subordinates and pushes the final outcome to those remote databases.

Secure Distributed Transaction Recovery Background Process (RECO) provides additional security for the RECO process.

View Documentation

IP Rate Limit in CMAN

You can use Oracle Connection Manager (CMAN) to limit the number of new connections allowed from an IP address in the specified unit of time. This IP rate limit feature enables you to protect your database against potential denial-of-service (DoS) attacks.

Malicious clients can send excessive connection requests to the server node. This can saturate the capacity of CMAN to handle new connections per second, and thus cause DoS attacks on your database. Using this security feature, you can prevent these types of attacks by detecting such clients early and rejecting those connections.

View Documentation

OCI Attributes for Microsoft Azure Active Directory Integration with Additional Oracle Database Environments

You can log into additional Oracle Database environments using your Microsoft Azure Active Directory (Azure AD) single sign-on `OAuth2` access token. The previous release supported Azure AD integration for Oracle Cloud Infrastructure (OCI) Autonomous Database (Shared Infrastructure). This release has expanded Azure AD integration to support on-premises Oracle Database release 19.16 and later. The project adds the OCI attributes needed to supply the bearer token for connection creation.

This multi-cloud feature integrates authentication and authorization between Azure AD and Oracle Databases in Oracle Cloud Infrastructure and on-premises.

View Documentation