ORACLE DATABASE 12c

Encryption and Redaction in Oracle Database 12c with Oracle Advanced Security

ORACLE WHITE PAPER | MARCH 2017

ORACLE

Table of Contents

Introduction 1

Preventing Database Bypass with Encryption 2

- Oracle Advanced Security Transparent Data Encryption 2
- Protecting Sensitive Data Using TDE Column Encryption 3
- Protecting Entire Applications Using TDE Tablespace Encryption 3
- Protecting the Database Using TDE Database Encryption 4
- Performance Characteristics 4
- Built-In Key Management 4
- Encryption Impact for Common Operational Activities 5

Limiting Sensitive Data Exposure with Data Redaction 6

- Oracle Advanced Security Data Redaction 6
- Policies and Transformations 7
- Performance Characteristics 8
- Security Considerations 8
- Easy to Deploy Data Redaction 8
- Comparison to Alternative Approaches 9

Applying Encryption and Redaction in Oracle Multitenant Architecture 10

Conclusion 10

Introduction

Rising security threats, expanding compliance requirements, consolidation, and cloud computing are just a few of the reasons why data security has become critical. Nearly 10 years after the first U.S. breach notification law, the need for strong preventive controls continues to increase as access to data expands. Initiatives such as the European Union's General Data Protection Regulation (GDPR) help ensure data security remains a top priority for organizations. Stolen client devices, including tablets and smartphones, have the potential to easily expose sensitive information as users move beyond the laptop. Outsourcing, offshoring, corporate mergers, and nearly continuous organizational change create additional risks by making it easier for malicious insiders to obtain sensitive data and for outside hackers to gain access to servers using social engineering attacks. These growing trends are just one reason why centralized and efficient protection of sensitive data, regardless of the applications being used, is more important than ever. Implementing security measures that consistently protect sensitive data at the source becomes a critical control as stored data continues to proliferate and access to data expands beyond traditional boundaries. Protecting data requires a defense in depth, multi-layered approach that encompasses controls to evaluate security postures, prevent data loss, detect suspicious activities and apply data access controls at the source through data-driven security. Oracle Database 12c Release 2 strengthens Oracle's industry leading database security solution by providing important new security measures in each of these areas.

Oracle Advanced Security option with Oracle Database 12c delivers two essential preventive controls covering encryption of data-at-rest and redaction of sensitive data displayed by applications. These controls help protect sensitive data from being exposed directly from storage or through applications. Oracle Advanced Security Transparent Data Encryption (TDE) helps prevent attacks that attempt to bypass the database and read sensitive information from data files at the operating system level, from database backups, or from database exports. Oracle Advanced Security Data Redaction complements TDE by redacting sensitive data in query results before the data leaves the database, thus reducing the risk of unauthorized data exposure in applications. This white paper describes TDE and Data Redaction and explains how these valuable preventive controls can work together to help secure your sensitive data.

Preventing Database Bypass with Encryption

Data-at-rest encryption is an important control for blocking unauthorized access to sensitive data using methods that circumvent the database. Privileged operating system accounts are just one of the vehicles used by attackers and malicious insiders to gain access to sensitive information directly in physical storage.

Oracle Advanced Security Transparent Data Encryption (TDE) stops attackers from bypassing the database and reading sensitive information from storage by encrypting data in the database layer. Applications and users authenticated to the database continue to have access to application data transparently, while unauthenticated users attempting to circumvent the database are denied access to clear text data. To understand this better, consider the fact that privileged operating system users can access database tablespace files and extract sensitive data using simple shell commands. In addition, consider the possibility of attacks that read sensitive data from lost, stolen, or improperly decommissioned disks or backups. Figure 1 shows an example of extracting customer credit card numbers directly from storage using the common Linux "strings" command and a search pattern.

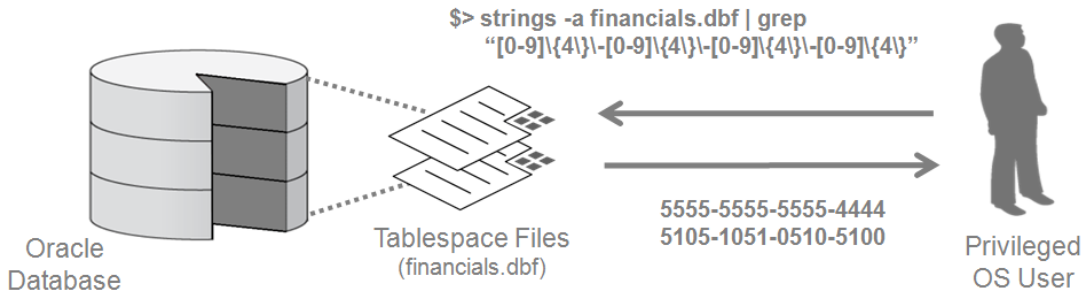

Figure 1. Extracting customer credit card numbers from Oracle database tablespace files

Oracle Advanced Security Transparent Data Encryption

Transparent Data Encryption resides at an optimal layer within the database to prevent database bypass while maintaining application transparency. TDE deploys quickly and encrypts individual application table columns, application tablespaces, or entire databases. It is transparent to applications because the encryption and decryption processes do not require any application changes, and the application users do not have to directly deal with encrypted data. Most importantly, TDE's built-in two-tier encryption key management provides full key lifecycle management, tracking the keys across their lifetime with helpful meta-data attributes, and assisted encryption key rotation, switching to a new master key with no downtime. Figure 2 shows how encrypting an Oracle database using TDE prevents database bypass.

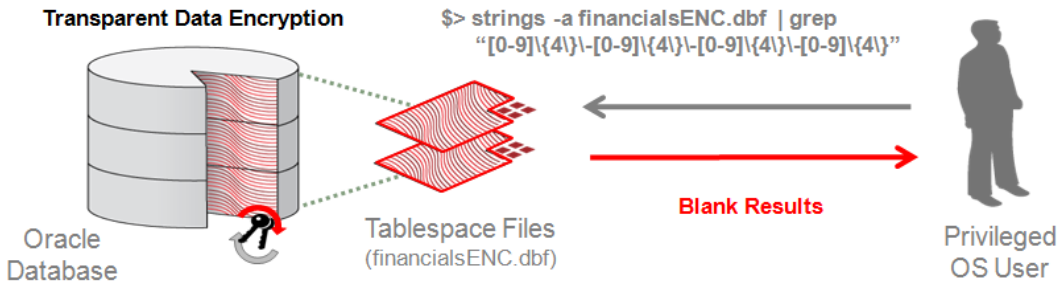

Figure 2. Encrypting with Transparent Data Encryption to prevent database bypass

TDE is unique when compared to alternative approaches that encrypt entire storage volumes or require new toolkits and programming APIs. These approaches do not protect against many bypass attacks, may require significant application changes, have complex key management, and are not integrated with complementary technologies such as Oracle Advanced Compression, Oracle Real Application Cluster (RAC), Oracle Recovery Manager (RMAN), Oracle Multitenant, Oracle GoldenGate, and Oracle Active Data Guard.

The high level of protection provided by TDE follows common standards for strong encryption as described in the figure below. With Oracle 12c Release 2, TDE supports operation with a FIPS 140-2 Level 1 cryptographic module, using approved encryption suites for SSL/TLS and TDE encryption.

Encryption Algorithms	**Hashing Algorithms (optional)**
Advanced Encryption Standard (AES) Key length: 128, 192, 256 bits	Secure Hash Algorithm 1 (SHA-1) Digest length: 160 bits
Triple Data Encryption Standard (TDES) Key length: 168 bits	
Regional encryption algorithms ARIA and SEED GOST	

Figure 3. Standard encryption and hashing algorithms used by TDE

Protecting Sensitive Data Using TDE Column Encryption

Oracle Advanced Security TDE column encryption can be used to encrypt specific data in application tables such as credit card numbers and U.S. Social Security numbers. Customers identify columns within their application schema containing sensitive or regulated data, and then encrypt only those columns. This approach is useful when the database tables are large, only a small number of columns must be encrypted, and the columns are known. TDE column encryption also is useful for warehouse applications where each query is likely to return a very different set of data. Oracle Enterprise Manager Sensitive Data Discovery searches for and identifies sensitive columns quickly. Data encrypted using TDE column encryption remains encrypted on backup media and discarded disk drives, helping prevent unauthorized access and potential data breaches that bypass the database.

Protecting Entire Applications Using TDE Tablespace Encryption

Oracle Advanced Security TDE tablespace encryption protects entire application tables by encrypting the underlying tablespaces. It encrypts application tablespaces regardless of the data's sensitivity and irrespective of its data type. Tablespace encryption simplifies the encryption process because there is no need to identify specific database columns. It is useful when the database contains a large amount of sensitive data to be encrypted and the columns reside in many different locations. TDE tablespace encryption and TDE column encryption can be used independently of one another or together within the same database. As is the case with both TDE column encryption and TDE tablespace encryption, data remains protected on backup media as a measure against potential bypass attacks.

Protecting the Database Using TDE Database Encryption

Oracle Advanced Security TDE database encryption protects entire databases including Oracle-supplied tablespaces SYS, SYSAUX, TEMP and UNDO. A new capability with Oracle 12c Release 2, this approach ensures that sensitive system and metadata information remain protected through encryption as well as application data.

Performance Characteristics

TDE's cryptographic operations are extremely fast and well integrated with related Oracle Database features. TDE leverages CPU-based hardware cryptographic acceleration available in Intel® AES-NI and Oracle SPARC T4/T5 platforms to increase performance by up to 5x or more. The block-level operations of TDE tablespace encryption receive an additional performance boost from database buffering and caching. Tablespace encryption integrates seamlessly with Oracle Advanced Compression, ensuring that compression occurs before encryption. Tablespace encryption also integrates with the advanced technologies in Oracle Exadata such as Exadata Hybrid Columnar Compression (EHCC) and Smart Scans, which offload certain cryptographic processing to storage cells for fast parallel execution.

Built-In Key Management

Key management is critical to the security of the encryption solution. Oracle Advanced Security TDE provides an out-of-the-box, two-tier key management architecture consisting of data encryption keys and a master encryption key. The data encryption keys are managed automatically by the database and are in-turn encrypted by the master encryption key. The master encryption key is stored and managed outside of the database within an Oracle Wallet, a standards-based PKCS12 file that protects keys, or in Oracle Key Vault, a centralized key management platform. Keeping the master key separate from the encrypted data mitigates attacks because both the keys and the encrypted data must be separately compromised to gain access to clear data. The two-tier key architecture also enables rotation of master keys without having to re-encrypt all of the sensitive data. Oracle Advanced Security defines a dedicated SYSKM role that may run all key management operations including rotating master keys and changing the keystore password. This role can be optionally delegated to a designated user account to enable separation of duty for these functions. Oracle Enterprise Manager provides a convenient graphical user interface for creating, rotating, and managing TDE master keys as shown in the figure below.

Oracle Key Vault is a full-stack, security-hardened software appliance which provides centralized management of encryption keys, Oracle Wallets, Java Keystores, and credential files. Oracle Key Vault works with TDE to automate the management of TDE master keys including creation, rotation, and expiration. Oracle Key Vault itemizes and stores the contents of Oracle Wallets in a master repository where they can be recovered back to servers if their local copies are mistakenly deleted or their passwords are forgotten. In addition, Oracle Key Vault can centrally manage TDE master keys over a direct network connection as an alternative to using local wallet files, eliminating operational challenges of wallet file management such as periodic password rotation, wallet file backups, and wallet file recovery. Using Oracle Key Vault with TDE enables sites to scale their TDE deployments to hundreds or

thousands of databases while improving operational efficiencies, reducing TCO, and enabling consistent key management policies. Oracle Key Vault supports hybrid cloud deployments, so organizations migrating to the Oracle Cloud can use it to support TDE deployments in both their cloud and on premises databases.

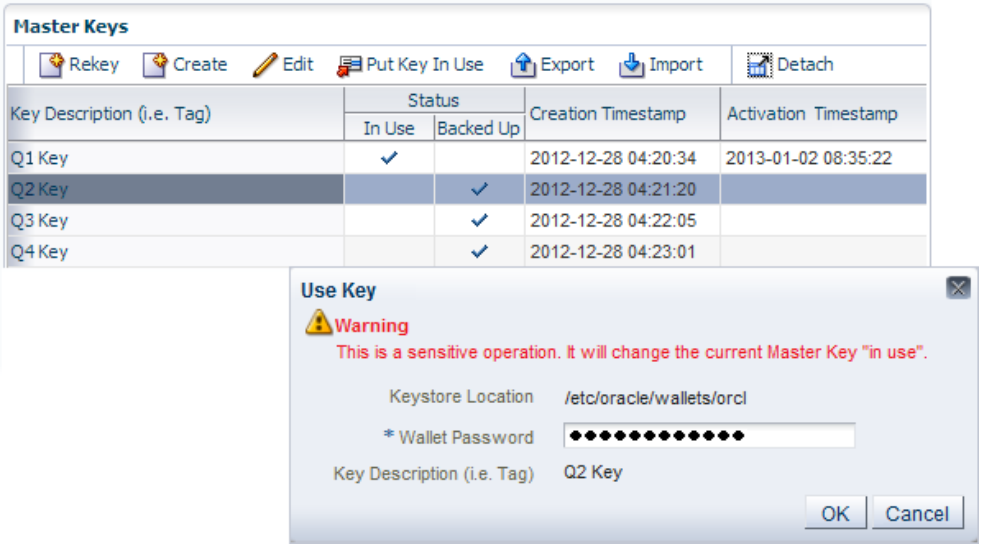

Figure 4. Managing and rotating TDE master keys using Oracle Enterprise Manager

Encryption Impact for Common Operational Activities

Essential day-to-day database operational activities can potentially leak sensitive data when not performed properly, making bypass easy. Examples of these activities include database backup and restore, data movement, high-availability clustering, and replication.

Database Technologies	Example Points of Integration	TDE Support
High-Availability Clusters	Oracle Real Application Clusters (RAC), Data Guard, Active Data Guard	✓
Backup and Restore	Oracle Recovery Manager (RMAN), Oracle Secure Backup	✓
Export and Import	Oracle Data Pump Export and Import	✓
Database Replication	Oracle Golden Gate	✓
Pluggable Databases	Oracle Multitenant Option	✓
Engineered Systems	Oracle Exadata Smart Scans	✓
Storage Management	Oracle Automatic Storage Management (ASM)	✓
Data Compression	Oracle Standard, Advanced , and Hybrid Columnar Compression	✓

Figure 5. Example integrations with Oracle Advanced Security TDE

Oracle Advanced Security TDE supports these critical database operational activities and helps ensure that the data remains encrypted. Tablespace encryption integrates with Oracle Recovery Manager (backup and restore), Oracle

Data Pump (data movement), Oracle Active Data Guard (redundancy and failover), and Oracle Golden Gate (replication). TDE also integrates with internal features of the database such as redo to prevent possible data leakage in logs. This fully integrated approach to database encryption makes the solution easy to deploy in complex real-world environments, while protecting against bypass attacks that attempt to take advantage of gaps in operational processes.

Oracle Database 12c Release 2 TDE provides two options for performing tablespace conversions from clear-text to encrypted tablespaces. For deployments which require conversion to be performed with no downtime, online tablespace encryption runs in the background to convert tablespaces from clear text to encrypted text while systems remain operational. TDE also offers an offline tablespace conversion mode which efficiently converts tablespaces with no storage overhead.

Limiting Sensitive Data Exposure with Data Redaction

Privacy and compliance require a cost effective approach to managing data exposure in applications. The embrace of smartphone and tablet devices make the issue of sensitive data exposure even more urgent as data access beyond the traditional office environment becomes commonplace. Even traditional applications require a more comprehensive solution for reducing exposure to sensitive data, for example a call center application with a screen that exposes customer credit card information and personally identifiable information to call center operators. Exposing that information, even to valid application users, may violate privacy regulations and put the data at unnecessary risk.

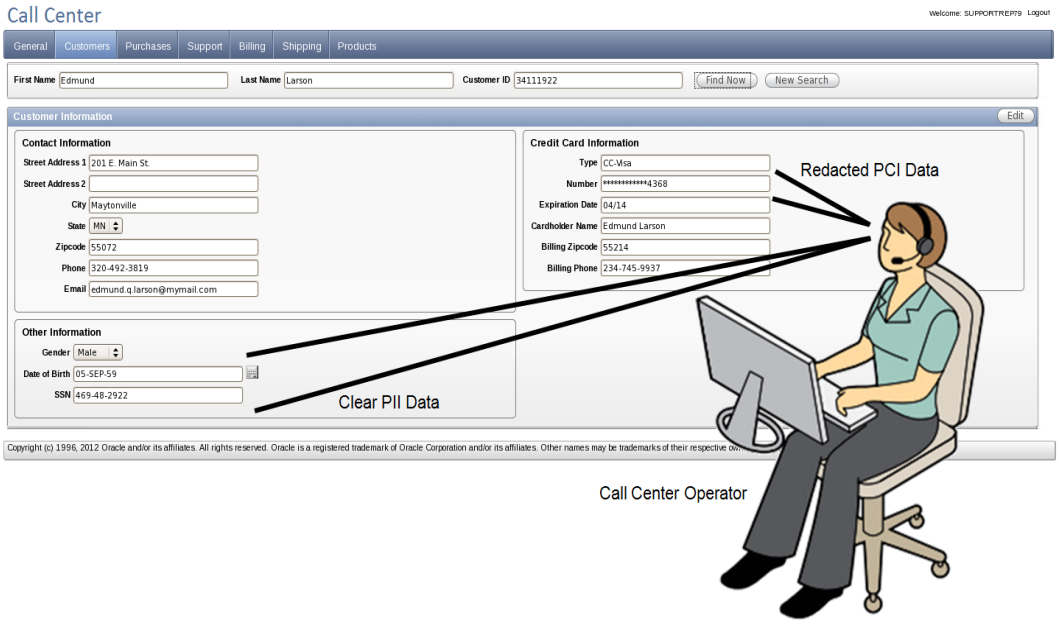

Figure 6. Clear and redacted information displayed in a call center application

Oracle Advanced Security Data Redaction

Oracle Advanced Security Data Redaction provides selective, on-the-fly redaction of sensitive data in database query results prior to display by applications so that unauthorized users cannot view the sensitive data. The stored

data remains unaltered, while displayed data is transformed and redacted on-the-fly before it leaves the database. Data Redaction reduces exposure of sensitive information and helps prevent exploitation of application flaws that may disclose sensitive data in application pages. It is well suited for both new and legacy applications that need to limit exposure of sensitive data without invasive application changes.

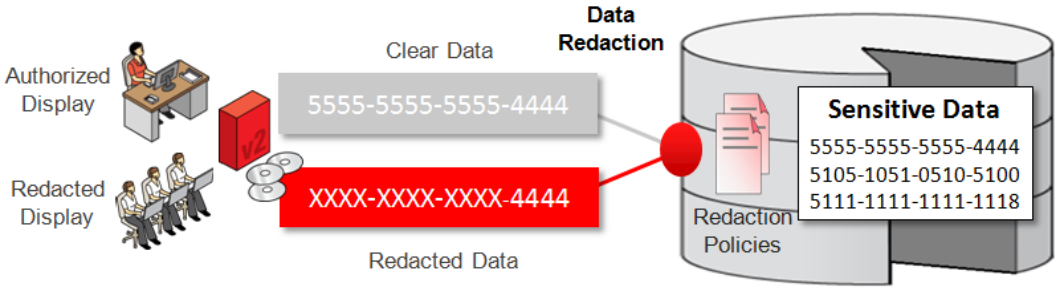

Figure 7. Redacting sensitive data displayed by applications using Data Redaction

Policies and Transformations

Oracle Advanced Security Data Redaction supports a number of different transformations that can redact all data in specified columns, preserve certain pieces of the data, or randomly generate replacement data. Examples of the supported data transformations are shown below.

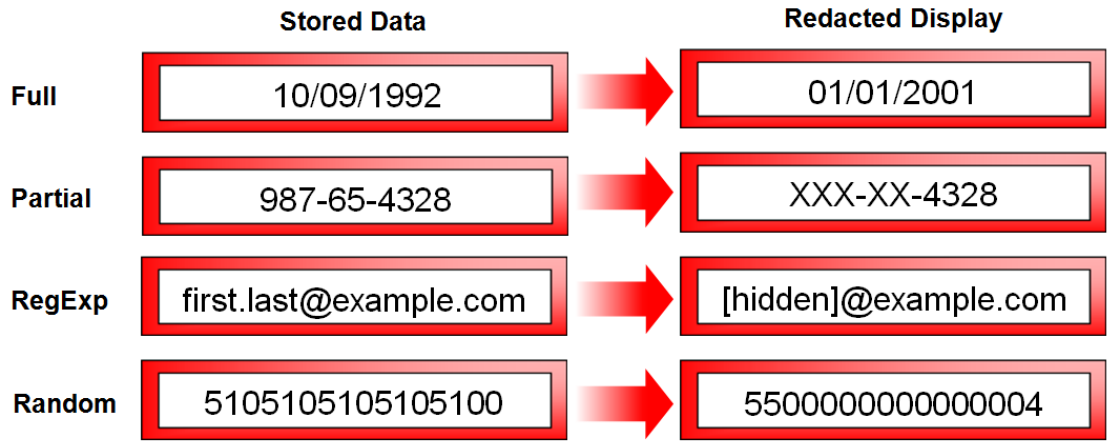

Figure 8. Example Data Redaction transformations

Data Redaction makes the business need-to-know decision based on declarative policy conditions that utilize rich runtime contexts available from the database and from the applications themselves. Examples include user identifiers, user roles, and client IP addresses. Context information available from Oracle Application Express (APEX), Oracle Real Application Security, and Oracle Label Security also can be utilized to define redaction policies. Redacting APEX applications is straightforward because policy conditions can leverage the application users and application identifiers that APEX automatically tracks. Multiple runtime conditions can be joined together within a data redaction policy for fine-grained control over when redaction occurs. The policies are stored and managed inside of the database, and they go into effect immediately upon being enabled.

Performance Characteristics

High-speed performance is crucial for Data Redaction because the target databases typically will be production systems. Data needs to be transformed on-the-fly at runtime, without altering data stored on disk or in caches and buffers. Because the transformations will execute on production environments and will be repeated frequently, the performance overhead must be small.

One important performance characteristic of Data Redaction is that it supports only data transformations with proven high performance. These are a subset of all the possible operations that could be used to transform data in non-production environments. This specific subset avoids long running and processor intensive operations.

Data Redaction also leverages performance optimizations of the Oracle Database that are only possible by being part of the database kernel. The implementation ensures that data transformations are fast in-memory computations. Policy information is cached in memory, and policy expressions are evaluated only once per execution, so there is no per row performance impact.

Security Considerations

Another benefit resulting from Data Redaction being part of the database kernel is tighter security. This implementation avoids potential vulnerabilities that plague other redaction techniques due to their dependence on proxies that can be meddled with. Additionally, Data Redaction in the kernel continues protecting sensitive data even when other security measures may be compromised. For example, runtime conditions in policies can narrow the impact of a SQL Injection attack by continuing to redact sensitive data even when the attack has bypassed other preventive controls in the application and database.

Data Redaction also avoids obvious sources of leakage where the redaction policy could be bypassed by copying data into a new table that does not have a policy. Certain mass copy operations that touch redacted data are blocked by default, and this behavior can be overridden where necessary using a Data Redaction exempt privilege.

Although Data Redaction can be used to prevent accidental viewing of sensitive data by privileged database users such as DBAs, it is intended primarily for redacting data displayed by software applications. Data Redaction does not prevent privileged users from connecting directly to the database and running ad hoc queries that back into pieces of sensitive data (i.e. it does not stop exhaustive ad hoc queries or other inference attacks). However, Data Redaction is fully compatible with Oracle Database security solutions that control and monitor privileged database user access, including DBAs. It can be deployed in tandem with other solutions such as Oracle Database Vault or Oracle Audit Vault and Database Firewall to provide defense-in-depth security. Data Redaction can also be used with database encryption as well, and it is a great complement to TDE.

Easy to Deploy Data Redaction

Data Redaction can be deployed for existing applications quickly using either a command line API or Oracle Enterprise Manager. The command line API is a PL/SQL procedure that accepts protected columns, transformation types, and conditions. Oracle Enterprise Manager provides a convenient Policy Expression Builder that enables administrators to define and apply redaction policies on existing applications. As shown below, the Policy Expression Builder dialog guides the user through creating policy conditions that use context obtained from applications, the database, the APEX framework, and other database security solutions.

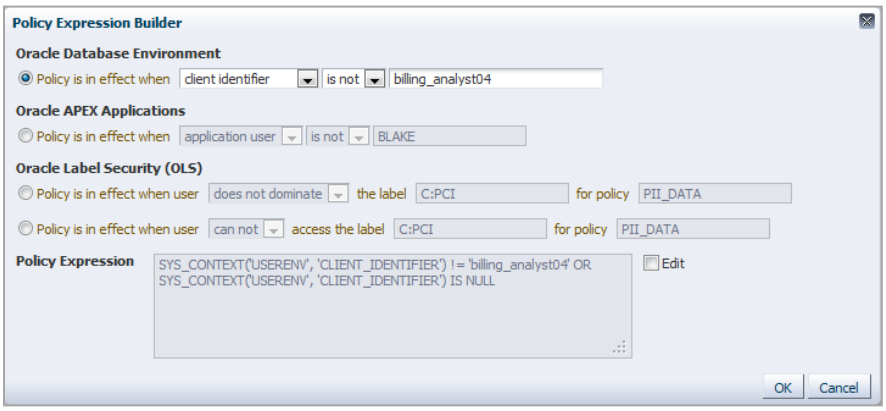

Figure 9. Using Oracle Enterprise Manager Policy Expression Builder to create Data Redaction policies

Predefined column templates also are available in Oracle Enterprise Manager for redacting common sensitive data such as credit card numbers and U.S. Social Security numbers. Oracle Enterprise Manager Sensitive Data Discovery assists in locating columns to be redacted inside of complex application schemas.

Another reason why Data Redaction is easy to deploy is its transparency to applications and the database. For application transparency, Data Redaction supports the column data types that are frequently used by applications and various database objects including tables, views, and materialized views. Redacted values retain key characteristics of the original data such as the data type and optional formatting characters. Random redaction values are drawn from data ranges defined by the existing column data. For transparency to the database, Data Redaction avoids impacting essential database operational activities. It does not affect administrative tasks such as data movement (Oracle Data Pump) or database backup and restore (Oracle Recovery Manager). It does not interfere with database cluster configurations such as Oracle Real Application Clusters, Oracle Active Data Guard, and Oracle GoldenGate. Data Redaction does not get in the way of existing database triggers or Oracle Virtual Private Database (VPD) policies. And because Data Redaction is part of the database kernel, no separate installation is required.

Comparison to Alternative Approaches

Traditional approaches to redacting sensitive data typically relied on application coding or installing third-party software on the database server to modify its behavior. These alternatives have important drawbacks compared to Data Redaction.

Approaches that require coding new application logic, modifying existing SQL statements, or authoring custom application scripts are likely to result in disparate solutions that are inconsistent across the enterprise and costly to maintain over their lifetime. In addition, strict controls must be placed on new application development to make sure that custom application code and new objects are properly accessed. The code also needs to take into consideration multiple factors under which the redaction policies are enforced, while maintaining the performance and semantics of the application.

Approaches that add new components to the Oracle Database, overwrite existing components, establish proxies, and modify basic behavior of the database also are fraught with problems. Not only do the new components introduce new attack surfaces, but they also can create performance overhead, impact operational activities of the database, and may fail when attempting to transform complex database queries that are generated by applications. In contrast, redacting directly in the Oracle Database kernel using Data Redaction has tighter security, superior performance, and better compatibility with a range of database configurations, use cases, and workloads.

Applying Encryption and Redaction in Oracle Multitenant Architecture

Oracle Advanced Security fully supports the Oracle Database 12*c* multitenant architecture. Both TDE and Data Redaction attributes automatically follow Pluggable Databases (PDB) as they move between multitenant Container Databases (CDB). When moving a PDB that has redaction policies, the policies transfer directly to the new container as part of the PDB. When moving an encrypted PDB, the TDE master keys for that PDB are transferred separately from the encrypted data to maintain proper security separation during transit. Encryption and redaction immediately resume their normal operation after the PDB has been plugged in and configured.

Conclusion

As data exposed in applications continues to rapidly expand, enterprises must have strong controls in place to protect data no matter what devices or applications are used. Oracle Database 12c Release 2, now available in the cloud and on-premises, helps organizations keep their sensitive information safe in this increasingly complex environment by delivering critical controls that enforce data security in the database.

Oracle Advanced Security with Oracle Database 12c Release 2 provides two critical preventive controls. Transparent Data Encryption encrypts data at rest to stop database bypass attacks from accessing sensitive information in storage. Data Redaction reduces exposure of sensitive information in applications by redacting database query results on-the-fly, according to defined policies. Together these two controls form the foundation of a multi-layered, defense-in-depth approach. They further establish Oracle Database 12c Release as the world's most advanced database solution

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

Encryption and Redaction in Oracle Database 12c with Oracle Advanced Security
March 2017