ORACLE

Risk-Driven Database Security

A practical approach to securing the Oracle Database

April 2020 | Version 20.01

PURPOSE

This technical white-paper provides an overview of securing an Oracle Database, including a discussion of features, options, and complimentary products. It is intended to help you evaluate options for reducing security risk and improving regulatory compliance for your Oracle Databases.

INTENDED AUDIENCE

If you are responsible for designing, implementing, maintaining, or operating security controls for an Oracle Database this paper is intended for you.

DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

TABLE OF CONTENTS

INTRODUCTION

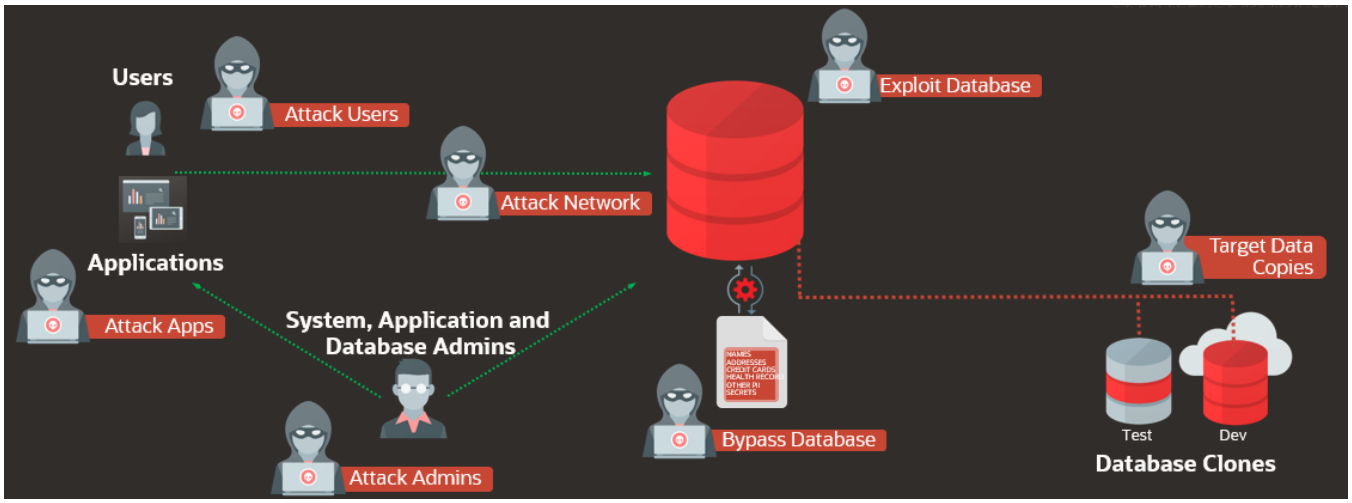
Oracle Databases contain the majority of the world's relational data, including sensitive data that is a prime target for data thieves. That sensitive data should be protected from theft or misuse. Two primary imperatives drive the need for organizations to protect data:

- Regulatory requirements – there are over 117 national laws governing data privacy (EU GDPR, Canada PIPEDA, Japan PIPA, Australian Privacy Principles, Brazilian LGPD), and a huge number of laws and regulations from state and provincial governments (California CCPA, Quebec Privacy Act), industry regulations (PCI, US FFIEC, US HIPAA, EU PSD2, BASEL). Listing all of the different laws and regulations would make this paper very long and would mean it would be out of date almost immediately. The cost of non-compliance with regulations can be severe – US HIPAA fines exceed $87M in the past three years, and individual GDPR fines have been as high €200M.
- Data Theft concerns – Data has value, and data thieves spend a lot of time stealing it. Unfortunately, they are frequently successful. Billions of personal data records have been stolen, and thousands of breaches are disclosed each year. The economic impact of this theft is incalculable – easily in the trillions of dollars

We will be focusing on risk reduction in this paper, which corresponds most closely to preventing data theft (or misuse). But despite the focus on risk reduction, the same controls discussed here play a vital role in regulatory compliance. After all, most data security-focused regulations are attempts to manage risk.

Now that we know why data should be protected, we will discuss how to secure your Oracle Databases. What the different features, options, and products are and how those combine to provide an appropriate level of protection for your sensitive data.

This paper is not intended to take the place of product documentation – any code examples used are for illustrative purposes only. I'll be discussing the use of several database features, options, and related products – there is a key to those products, along with links if you'd like more information about them, in the Appendix: **Tools – Features, Options, Products, and Packs**.



Attack Points for the Oracle Database

HOW ARE DATABASES COMPROMISED?

To better understand how to protect a database, you should first give some thought to how databases are typically breached.

- The most common point of compromise for a database starts with a valid database account. That can be a DBA account, an application service account, or even an end-user account. Most data stolen from a database leaves through a valid account that is stolen or is being used in a way that violates policy.
- Another common attack technique is to bypass the database altogether – if an attacker can access the underlying storage for the database, steal a database backup, or acquire a database export then they sidestep the database's access and monitoring controls.
- Unless you are using one of the Autonomous Database cloud services, then you're managing the application of security patches to your database to prevent database exploits. Attackers will probe the database and the underlying operating system for known exploits in hopes of finding un-patched vulnerabilities.
- Attackers who penetrate your network perimeter may simply lurk on your network and sniff for "interesting" data – these types of network attacks are attractive because the chances of being caught are so low.
- All of these attacks can be executed against the production database, but copies of production, frequently made for test and development purposes, are also good targets. In fact, those non-production copies of the database are BETTER targets, because they are less likely to be closely monitored, and often lack the security controls used in production. If it is the same data, does it matter which copy of the database it came from?

Keep in mind that a solution that only protects against one of these attacks just redirects the attacker to another weak point - you need to cover all the avenues of attack to truly secure your data.

WHAT IS RISK?

Risk is a product of threat, value, vulnerability, and impact.

- Threat: how likely is it that someone will attempt to steal, destroy, or misuse your data?
- Value: what is the value of the data? Both to the attacker and to your organization?
- Vulnerability: how exposed is the data? What are the chances that an attempt to compromise the data will succeed?
- Impact: how much damage would a breach cause to the organization. This could be in terms of fines, lost opportunity costs, or damage to customer confidence

You have little ability to impact threat or impact, but your efforts can reduce the vulnerability and, in the case of non-production database copies, may be able to reduce the value of the data. You can only do three things with risk – mitigate it, insure it, or accept it. We will focus on mitigating risk – reducing risk to an acceptable level.

ESTABLISH A SECURITY BASELINE

There are certain controls that should be implemented in each of your databases – these create a minimum security baseline. The baseline should reflect your organizations policies and risk tolerance. A good way to think about your database security baseline is that this is what you expect to see in ALL databases, regardless of what data they contain or what their

usage is. There should be few exceptions (if any) granted to the baseline, and when there IS an exception that exception should be periodically reviewed to ensure it is still valid. Here are the procedures and controls we think should be part of all database security baselines.

Assess your database

Before you make any changes, it is a good idea to assess your database to understand the current state of security. Review the database configuration, basic security policies, user entitlements, and password policies. Scan your database for sensitive data to understand what additional protections are appropriate for this database.

Validate Database Configuration

A security assessment of your database checks initialization parameters, listener settings, missing security patches, and more. Oracle Databases are extremely flexible, configurable to meet almost any business need with hundreds of parameters. A few of those parameters change the level of security risk in the database, and it is important to be aware of configuration settings that don't follow best practices for reducing risk. Use a security assessment to identify configuration choices that introduce unnecessary risk and, where possible, reconfigure the system to remove that risk. Your baseline security posture should reflect configurations that support your minimum-security standard. If you do not already have an organizational standard for secure database configuration, consider adopting either the Center for Internet Security (CIS) benchmark configuration, or the United States Defense Information Systems Agency (DISA) Secure Technical Implementation Guide (STIG) for Oracle Database. Oracle provides several tools to help you assess your database security, including the Database Security Assessment Tool (DBSAT), Oracle Data Safe, and Oracle Enterprise Manager Lifecycle Management. All three tools map findings to the CIS Benchmark and STIG.

Review User Authentication

Oracle Database supports authentication via username and password, PKI certificate, Kerberos, and RADIUS. By far the most common authentication mechanism is still the simple username and password.

If your database accounts are still authenticated by password, ensure that you are enforcing good password discipline. In most cases your database password policies should reflect your standard organizational policies – that includes requirements for password length, lifetime, and complexity. Your Oracle Database is a business-critical system containing sensitive information – why would you accept weaker password policies than you allow for less critical systems like laptops?

For database service accounts, where expiring passwords are often impractical because of the downstream impact on availability, consider mitigating the risk of those passwords not expiring with additional controls that closely limit how those accounts are allowed to connect. We will talk more about this type of mitigation later in this paper when we discuss access controls. Locking accounts after a certain number of failed logins is also a good practice and reduces the chance of a brute-force password attack succeeding.

Consider strong authentication for interactive database accounts – the most common strong authentication is Kerberos, usually using a Microsoft Active Directory Domain Controller as the Kerberos key distribution center. Moving authentication outside of the database into Active Directory has several advantages, including centralized control of authentication tokens, single sign-on with the Windows desktop, and the ability to disable database logins for ALL databases with a single action.

A less common strong authentication mechanism that is rapidly increasing in popularity is RADIUS. RADIUS is a venerable, well known authentication mechanism that has been supported for over a decade with the Oracle Database. What is driving recent adoption is the embrace of RADIUS by cloud-based authentication services like Oracle Identity Cloud Service and Okta.

User Entitlements

Since compromised accounts are the most common source of database breaches, review assigned privileges and remove any that are not necessary to reduce the threat those accounts pose if compromised. Both Data Safe and DBSAT help with entitlement reviews by reporting on the privileges an account has been granted. Additionally, Oracle provides *privilege analysis* to assist in assessing the privileges an account actually *uses*. Knowing which privileges an application account requires is valuable because it helps you identify candidate privileges to be revoked as you drive towards accounts with only the privileges required to complete their assigned tasks.

A good practice with privilege analysis is to identify candidate privileges and roles for removal, and then audit the use of those privileges and roles for a period of time to ensure you don't accidentally revoke a privilege that is required

infrequently. Caution in revoking privileges is especially appropriate for non-human accounts (application accounts, batch processes).

Encrypt Data in Motion

Data is at risk as it traverses the network between the database and database client or application. Skilled attackers will frequently monitor network traffic, sniffing for sensitive data. This type of passive attack is extremely difficult to detect because the attackers don't actually try to penetrate the database or database server.

Fortunately, Oracle Database offers a few options for encrypting data in motion.

Oracle Native Network Encryption

Oracle Native Network Encryption (NNE) is the easiest way to encrypt data in motion. It requires a single line added to the database's network configuration file (sqlnet.ora), and in most cases no changes to database clients. Oracle Database can either request or require encryption. If the database requests encryption, clients that support encryption will automatically default to encryption and clients that do not support encryption will fall back to an unencrypted connection. If the database requires encryption, clients that do not support encryption will fail to connect. The encrypted connection will use the strongest mutually-supported encryption algorithm, most commonly AES-256.

Transport Layer Security

Transport Layer Security (TLS) also encrypts data in motion. Unlike NNE, TLS requires a certificate for the database server, and allows a certificate at the database client. If only the server uses a certificate, then the connection is referred to as "server authenticated" and the client still requires user authentication to connect. If the database client is also issued a certificate, then the connection is mutually authenticated, and the client certificate is used to authenticate the user.

Which is better, NNE or TLS?

That depends on your use case. The encryption quality used for both is equivalent in terms of strength, but TLS adds server authentication to encryption, and can be used to also authenticate the client. TLS is an industry standard protocol, well known by most security teams – but TLS almost always requires changes to the client configuration and uses certificates that eventually expire and need to be maintained. NNE is easier to set up and seldom requires any changes to the client. In cases where operational efficiency is most important, we will usually see NNE used. In cases where the highest level of security is required – even if it means some compromises for operational efficiency, TLS is the most common choice.

Manage Database Users and Roles

Database users and roles can be managed locally, within the database. They can also be centrally managed in an LDAP directory. If you have few users that connect to your databases, and few databases for them to connect to, then local user management is probably the right choice for you. If you have LOTS of database users, or are managing lots of different databases, then centrally managing your users is probably a better choice. There are two options for centrally managing database users.

Centrally Managed Users

Centrally Managed Users connects Oracle Databases to Microsoft Active Directory, with database schemas mapped to Active Directory users or groups, and database roles mapped to Active Directory groups. Authentication can be with password, PKI certificate, or Kerberos, with Kerberos authentication being the most common authentication method because it is the least intrusive on the Active Directory environment. The minimum database version with support for Centrally Managed Users is Oracle Database 18c. Centrally Managed Users lets you manage database accounts for multiple databases in a single place (Active Directory) using common credentials across all connected databases.

Enterprise User Security

Enterprise User Security connects Oracle Databases to Oracle Internet Directory, with database schemas mapped to Internet Directory users or organizational units, and database roles mapped to Internet Directory groups. Authentication can be with password, PKI certificate, or Kerberos, with password authentication being the most common authentication method. In many cases, the Oracle Directory serves as a proxy for Microsoft Active Directory – in these cases, Kerberos is the most common authentication mechanism. Enterprise User Security is supported for all current Database Versions. Enterprise User

Security lets you manage database accounts for multiple databases in a single place (Oracle Internet Directory) using common credentials across all connected databases.

Audit User Activity

Database auditing generally serves three purposes:

- Provide a record of database activity to support forensic investigation following a security incident
- Provide a record of database activity to support development and operations
- Identify access anomalies in order to halt or prevent a security breach

After a security incident, you will need to determine what happened, who did it, where the attack originated from, when it occurred, how it happened, and what data was impacted. Your audit trail is a crucial part of that post-incident investigation. Having a record of database logins, changes to users, alterations of database objects, and access to sensitive data lets you support those investigations and provides evidence of the extent of the incident.

Database development and operations frequently require a record of what has occurred – who shut the database down, when was a table dropped, when did application service accounts stop successfully authenticating – your audit trail can be a valuable aid in troubleshooting and root cause analysis. Fortunately, the same audit rules required to support forensic investigation are usually sufficient to support development and operations.

Anomaly detection and incident prevention are one of the most sought-after goals for auditing – we'll talk more about anomaly detection and incident prevention later in this paper as we move beyond baseline security towards a maximum security architecture.

Oracle Database provides a rich set of features for auditing user activity, including traditional auditing, unified auditing, and fine-grained auditing. Auditing helps with several security goals:

- Assist during a post-incident investigation
- Support compliance reporting
- Detect anomalies in system usage

Auditing always impacts performance and storage to some degree, so your audit policies should consider the value of the audit data being collected. "Audit everything" is not usually practical because of the significant additional load it places on the system. Here are a few activities that are normally low frequency, and of high security value. Auditing policies should capture these activities as part of the auditing baseline:

- Changes to user accounts (create, alter, drop)
- Database logins – especially login failures
- Grants of privileges and roles
- Create, alter, drop of database objects including tables, views, database links, and stored procedures
- All database administrator actions
- Database exports and backups

Beyond the baseline, audit policies should capture attempts to access data outside of policy. For example, if policy does not allow access to data from outside the application, then audit should capture attempts to circumvent the policy.

If your database version supports unified auditing, that should be your first choice in creating audit policies. Unified auditing offers significant advantages over traditional audit, including the ability to better focus your audit policies to capture only the data that is needed. The Oracle Database includes a number of unified audit policies out-of-the-box to meet common audit requirements. Unified auditing was first available in Oracle Database 12.1.

Sensitive Data Discovery

For most databases, the decision to go beyond the security baseline will be determined by the data stored within the database. Are there regulatory requirements for additional security measures? Does the data present a business risk that justifies an investment in additional security controls? For some databases, the answer will be obvious – if this is your HCM or CRM database, then you know it contains sensitive personal data and can act accordingly. If it is a database that hosts run-time sensor data for your production shop floor, then perhaps it does not need additional protection (unless that data exposes sensitive intellectual property).

For other databases, you may not know the appropriate level of protection – and that's where sensitive data discovery comes into the picture. Sensitive data discovery scans your database for sensitive data "types" – I use the quotes because this is not data types the way we normally talk about them in a database sense – char, number, blob, etc. This is "types" as in

email addresses, taxpayer identifiers, account numbers. Sensitive Data Discovery tells you what types of data are in your database and how much of it there is. Use this information to make a data-driven decision about the risk a database contains and the protections required to mitigate that risk.

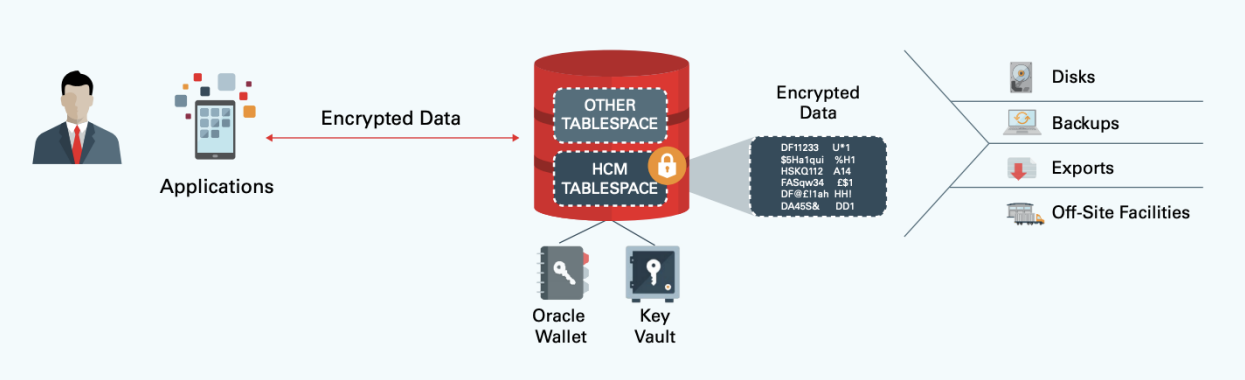
Oracle provides three tools that can help with sensitive data discovery. Database Security Assessment Tool (DBSAT), Data Safe, and Enterprise Manager Application Data Modeling (ADM), part of the Data Masking and Subsetting Pack. If you have access to it, Data Safe should be your first choice. Data Safe offers sensitive data discovery, scanning database metadata (column names and comments) and actual data within tables for over 125 different types of sensitive data. Data Safe's sensitive data format library is extensible, allowing both create and create-like to easily model organizational or regionally unique data patterns.

If you already use Oracle Enterprise Manager, and have the required license for ADM, then ADM is another good choice for sensitive data discovery.

If you are working with non-English databases, consider DBSAT - DBSAT is the only sensitive data discovery tool from Oracle that supports scanning in different languages, and as of the time I'm writing this DBSAT ships with discovery pattern files in English, Spanish, German, Portuguese, Italian, French, Dutch, and Greek. Unlike Data Safe or ADM, DBSAT does not actually scan table data, only metadata including column names and column comments.

BEYOND THE BASELINE – MAXIMUM SECURITY ARCHITECTURE

The Maximum-Security Architecture (MSA) applies a defense-in-depth approach to database security, extending the security baseline to further reduce risk in systems that contain sensitive data or are subject to regulatory constraints. Different components of the MSA are used to mitigate various attack vectors. Very few databases will use all features of the MSA, but most databases containing sensitive data will use one or more components of the MSA to reduce risk, improve security, and strengthen regulatory compliance.



Encrypting Data at Rest with Transparent Data Encryption

Encrypt Data At-Rest

At-rest data encryption mitigates the risk of database bypass. If an attacker gains access to the underlying database storage, database backups, or database exports, then encryption prevents the attacker from simply reading the data directly without using a database API. That type of bypass attack would circumvent access controls and would not trigger audit records – a worst-case situation that it is critical to avoid. Encryption is a common requirement for data privacy and protection regulations. Failure to encrypt data is typically considered a failure to exercise due care in protecting data. Oracle's primary encryption solution is Transparent Data Encryption, a feature of Oracle Advanced Security (ASO).

Transparent Data Encryption (TDE) is the most common control above the baseline – implemented by more Oracle customers than any of the other non-baseline security features. In many organizations, TDE is part of the baseline. A good example of this is the Oracle Cloud – TDE is the default there because Oracle does not wish to accept the risk of hosting your data in unencrypted form. Most TDE implementations are trouble free – TDE is a mature feature, with over a decade of successful deployments by many thousands of customers. Here are a few things to consider.

Do your own performance benchmarks on real data workloads – not on artificial queries. It is important to use actual workloads to accurately judge the impact on production. Encryption always adds some performance overhead because you are asking the system to do more work, especially the CPU. Combining TDE with Advanced Compression is a great way to reduce that overhead, since Advanced Compression lowers the number of data blocks that need to be decrypted. Tablespace encryption tends to perform better with most workloads than column level encryption, and with tablespace

encryption you lessen the chance that you'll forget to encrypt a sensitive column, or that someone will insert sensitive data into a column that shouldn't contain it. Use the strongest encryption supported by your database version – as of the time this is written, that is Advanced Encryption Standard (AES) with a 256-bit key. Lower levels of encryption and shorter key lengths may have slightly less performance impact, but in most cases the difference is not significant

Remember to consider the impact of encryption on your backup system – most modern backup storage compress and de-duplicate backups. When a database is first encrypted, all of the encrypted data blocks look new to the storage system, so there is no de-duplication and you will see an initial sharp rise in storage requirements. As time goes on and un-encrypted backups age out of the system, you will see de-duplication return to normal levels. Encrypted data tends not to compress well, and as a result there will be some permanent loss of compression. Here again, Advanced Compression can help mitigate the impact because with Advanced Compression the data is compressed before being encrypted. This subject of encryption's impact on backup storage is one of the most commonly overlooked areas in deployment plans for Transparent Data Encryption.

Manage and Protect Encryption Keys

Data at-rest encryption requires a persistent key, and the security of the encryption is no better than the security of the encryption key. Transparent Data Encryption includes automated key management, and Oracle Key Vault provides secure key storage, centralized administration, and secure distribution of keys to support advanced data architectures like Real Application Clusters and Oracle Data Guard.

Ensure your keys are backed up (separately from the database backups). Use Oracle Key Vault instead of the default Oracle Wallet for key storage. For enhanced security, consider chaining Key Vault with a Hardware Security Module (HSM) as a root of trust.

Enforce Separation of Duties

In databases containing sensitive data, separating administrative duties is a common security goal. The administrator who can create an account and manage authentication credentials cannot grant access to data. The data administrator does not have the ability to the ability to create user or change their credentials. When Database Vault is enabled it implements separation of duties (SOD) for user account management by default.

The appropriate level of SOD for an organization depends on a lot of factors – not just the sensitivity of the system or the data within it. If an organization only has one DBA, then a complex SOD setup is probably less important. On the other hand, if an organization has dozens or hundreds of DBAs, then further separating responsibilities is a good way to limit the damage if an administrator account is compromised. Other area's we have seen separated (in just about every combination imaginable) include backup and recovery management, performance management, data integration, security administration, data administration, and patching. Adopt a model that makes sense in the context of your organization, but wherever possible drive towards using accounts with the least privileges required for a job function. If your DBAs are specialized, then they should not have the privileges contained in the generic DBA role. They should have privileges tailored for their job function.

Control Administrator Access to Sensitive Data

Database Administrator accounts are one of a hacker's favorite targets, and most database breaches involve the use of compromised credentials. For that reason, compromised Database Administrator accounts can be dangerous to the organization. Fortunately, database administrators rarely need access to sensitive data – or to any application data at all. Block database administrator access to sensitive data with Database Vault and you significantly reduce the amount of damage a compromised administrator account can inflict.

Enforce Trusted Path Access to Sensitive Data

For a data thief, compromising the service account used by the application to access sensitive data is almost as good as compromising the DBA account. The application service account typically has full access to data, and in many cases the application service account credentials are difficult to update without negative operational impact. This can lead to widespread dissemination of the credentials (especially within the developer/DevOps team) and that opens the potential for misuse and an increased risk of compromise for the account. Lockdown those application service accounts with Database Vault, limiting their use to just the application servers, running the application server programs, started as the application server's operating system user. Use multiple factors to force use of the application service account to that trusted path and deny an attacker or curious team member the opportunity to misuse the account.

Centrally Manage Audit Data

You may recall from our discussion of auditing in the baseline security section that audit data can be used for supporting investigations, both forensic and operational. Analyzing that audit data for any individual database can be done manually with simple SQL queries, but if you have multiple databases it may be more practical to collect audit data from those databases into a central repository where it can be more easily analyzed and reported on. Plus an attacker with privileged user credentials may be able alter/delete locally stored audit data – especially if you have not switched to unified audit yet.

Oracle Audit Vault and Database Firewall collects audit data from Oracle Databases (as well as many other databases including Oracle MySQL, Microsoft SQL Server, IBM DB2, PostgreSQL, MongoDB, and SAP Sybase) and places that audit data in a managed data warehouse where it can be reviewed, analyzed, reported on, and (when appropriate) generate alerts.

Monitor Database Activity to Detect Anomalies

Anomaly detection and incident prevention are one of the most sought-after goals for auditing - the same audit trail that you can use to support an investigation might have been used to detect an attack in progress – or even to detect an attack at the earliest stages before successful access to the database occurred. Your audit facility should alert you when anomalous activity occurs – for example, multiple failed login attempts, or unauthorized creation of new user accounts.

There is no substitute for database auditing, but auditing is usually not well suited for anything more that the most basic anomaly detection. This is because you can very rarely audit ALL activity (the performance impact of auditing all activity would be high, and the storage required to maintain all of those audit records would also be significant). Advanced anomaly detection means that all activity needs to be inspected. We work around the limitations of auditing by supplementing the audit data with network-based activity monitoring. Monitoring differs from auditing in several ways – there is no session context involved with monitoring, nor is there access to underlying database metadata. Network based monitoring doesn't give audit-quality data, but network monitoring is very good at providing enough data to spot deviations from normal patterns – someone using a new client program to access the database, someone connecting as an OS user who we've never seen before, an application that suddenly starts issuing SQL statements that don't follow the normal pattern for the application – these are all anomalies that network-based monitoring is very good at capturing.

Oracle Audit Vault and Database Firewall performs network monitoring, helping you quickly profile normal database activity and begin detecting deviations from that norm. The combination of native database audit and network-based monitoring gives you a complete picture of database activity.

Prevent Anomalies

For some systems it makes sense to block anomalies from occurring instead of just monitoring them – here you are making a conscious choice to prevent activity that looks suspicious, potentially blocking legitimate activity. This type of control is usually applied to your most sensitive systems, or systems that are at extremely high levels of risk due to factors other than just the data within them.

Oracle Audit Vault and Database Firewall can be used to block anomalies when the firewall is placed in-line with database traffic, so that activity has to actually pass *through* the firewall before it reaches the database.

Minimize Sensitive Data – Remove Risk from Non-Production Databases

The controls discussed so far limit and monitor data access – but there are times when that limitation defeats the purpose of the system. A good example of this is non-production test or development systems. By their very nature these systems tend to have more relaxed access controls and more variation in how they are operated. If your non-production database is created with artificial data, then this is not a problem because there is no risk inherent in the artificial data. But if you create your non-production system by the common practice of just cloning production then you duplicate the risk in the production system, increasing the chances of a security incident.

Minimize sensitive data in non-production database copies with Data Masking

For these non-production copies of production systems, a good technique is to mask the sensitive data – replace it with artificial values that remove the sensitivity of the data while still providing an environment suitable for testing and development. Masking differs from the other controls we have discussed in that it doesn't mitigate risk; it actually removes the risk. Oracle offers two solutions for data masking, Oracle Data Safe, and Oracle Enterprise Manager Data Masking and Subsetting. Both solutions let you discover sensitive data, identify referential integrity constraints, and mask the data to remove security risk. Choose whichever best fits your architecture.

TAKE A RISK-BASED APPROACH

Remember that we started with the idea of a security baseline, applied to ALL your databases. Next, we talked about sensitive data discovery and basing the use of controls beyond the baseline on the risk level of the system. Here are a few sample systems with associated risk levels and controls.

System	Risk Level	Controls above baseline	Remarks
Human Resources	High	Encryption, key management, separation of duties, trusted path	Privacy concerns/regulations
Financial Reporting	High	Encryption, key management, separation of duties	Sarbanes-Oxley or similar
Customer Management	Very High	Encryption, key management, separation of duties, trusted path, anomaly detection	Privacy concerns/regulations
Order Fulfillment/ Shipping	Very High	Encryption, key management, separation of duties, trusted path, anomaly detection	Privacy concerns/regulations
Web Store	Very High	Encryption, key management, separation of duties trusted path, anomaly prevention	Internet facing application, privacy concerns/ regulations
Bug Database/ source code	Highest	Encryption, key management, separation of duties, trusted path, anomaly prevention	Intellectual property
Test and Development	High	Data Masking	Remove sensitive data from these systems

START HERE

We have covered a lot, and each of the areas I have briefly introduced could be the subject of its own individual white paper. The amount of work to be done can be daunting – I've seen more than one organization lapse into analysis-paralysis, spending inordinate amounts of time trying to put together the perfect implementation plan, and never actually accomplishing anything that reduces their risk. In one very memorable case, about six months into that analysis phase I saw a very good customer – some of the nicest people you could ever hope to work with – breached with very public disclosure, causing over $175M in breach costs and significant organization upset. Here is what I suggest in hopes of helping you avoid that trap.

Do Something!

There is no perfect security in a usable system. It is a balancing act between the drive to secure data and the need to support operations and use the data. The best you can usually do is chip away at risk until you reduce it to a level that your organization can live with. Any of the security controls I outlined above should help with risk reduction, getting you closer to that "acceptable risk" level.

A smart place to start is the first thing I covered in this paper – begin improving security with an assessment of your system and configuration. Identify your current state and decide what state you would like to get to. Which controls make the most sense to adopt. I outlined a baseline level of security in the first part of this paper – does that baseline make sense to you? If so, then adopt it and begin to apply it to systems during maintenance periods. If you have systems that you KNOW are sensitive, focus on them first. Remember, the risk you remove today may be the risk an attacker would have exploited tomorrow.

PARTING THOUGHTS

As you begin to reduce risk and improve security, you are virtually certain to face organizational inertia – do not be discouraged. The stakes are too high to continue to accept the status quo. Enlist the help of your organization's security group – they may have resources to help plan and organize this effort – allowing you to focus on the technical implementation.

Remember where we started, your databases contain some of your most valuable information – it is a safe bet that someone would be happy to take that information from you and find ways to profit from it. The controls I've outlined in this paper reduce the chances of their being successful, with each control contributing its part to a more secure system. Best of luck on your project!

APPENDIX: TOOLS – FEATURES, OPTIONS, PRODUCTS, AND PACKS

Any discussion of this nature can easily devolve into a listing of products and features – I've tried to minimize that in the main portion of this paper. Below is a list of the different features, options, products, and packs mentioned in the paper, along with links to documentation. The features are listed in the order in which they were mentioned above. One thing to consider about those documentation links – they are current as of the time I write this, but white papers tend to linger for a long time, while database versions and documentation are fluid and frequently updated. Check docs.oracle.com to find the latest version of the documentation – the links provided here will at least let you know which manual (and usually chapter or appendix) to look for.

Database Security Assessment Tool (DBSAT)

DBSAT is a standalone utility included with your database support. DBSAT helps with security assessment, user assessment, and sensitive data discovery. There is no additional fee for the use of DBSAT. DBSAT works with all supported versions of the Oracle Database, on all supported operating systems, and can be run for databases on-premises or in the cloud – including non-Oracle clouds. The utility may be downloaded from My Oracle Support – check MOS note 2138254.1 for more information on downloading the tool. DBSAT documentation is available here.

Oracle Data Safe

Data Safe is an Oracle Cloud service, included with Oracle Database as a Service offerings and available for use with on-premises databases and Oracle Databases running on Oracle Cloud Compute. Data Safe includes several database security capabilities, including security assessment, user assessment, sensitive data discovery, sensitive data masking, unified audit policy control, audit data retrieval, reporting, and alert generation. Data Safe is Oracle's newest database security service and is rapidly evolving (two-week development sprints) new features and capabilities. Click "What's New" in the documentation for updates on recent changes. Documentation for Data Safe is included here.

Enterprise Manager Database Lifecycle Management

Database Lifecycle Management is a management pack for Oracle Enterprise Manager Cloud Control. Database Lifecycle Management provides numerous functions for managing the lifecycle of your database, including configuration management, and can play a valuable part in security assessments. Information about configuration management using Enterprise Manager Database Lifecycle Management is available here.

Privilege Analysis

Privilege analysis is a database feature included with all databases and database services except for standard edition. It helps with user assessment, particularly with determining which privileges a user account has, but is not using. Privilege Analysis was introduced in Oracle Database 12c Release 1. Privilege Analysis documentation is available here.

Native Network Encryption

Native Network Encryption (NNE) is a database feature included with all databases and database services with the exception of Autonomous Database (Autonomous Database uses TLS instead of NNE). NNE encrypts data as it travels between the database and database client or application. NNE documentation is available here.

Transport Layer Security

Transport Layer Security (TLS) is a database feature included with all databases and database services. It is configured by default for Autonomous Databases. TLS encrypts data as it travels between database and database client or application. TLS documentation is available here.

Centrally Managed Users

Centrally Managed Users (CMU) is a database feature included with Oracle Database Enterprise Edition. CMU was introduced with Oracle Database 18c. CMU allows Oracle Databases to connect directly to Microsoft Active Directory. With CMU, users are created in Active Directory and mapped to database schemas. Optionally, database roles can be associated with Active Directory groups, and database role membership controlled by Active Directory group membership. CMU documentation is available here.

Enterprise User Security

Enterprise User Security (EUS) is a database feature included with Oracle Database Enterprise Edition. EUS was introduced with Oracle Database 8.1 (Oracle 8i). EUS allows Oracle Databases to connect to Oracle Internet Directory. With EUS, users are created in Internet Directory and database schemas are mapped to users or to collections of users within an LDAP organizational unit. Database roles can be associated with Internet Directory groups, and database role membership controlled by LDAP group membership. EUS documentation is available here. Internet Directory documentation is available here.

Traditional Auditing

Traditional auditing was first introduced in Oracle Database 7 and was the primary auditing mechanism for Oracle Database until the release of Oracle Database 12c. Traditional auditing is being replaced by unified audit. Traditional auditing is deprecated starting with Oracle Database 20c, and will be obsolete in Oracle Database 22c. Traditional audit documentation is available here.

Fine-Grained Auditing

Fine-grained auditing was introduced in Oracle Database 9.0 (9i Release 1). As the name suggests, Fine-grained auditing allows audit policies to be focused more narrowly than traditional auditing, allowing audit policies based on columns. Fine-grained auditing also introduced the concept of conditional auditing, where audit records would only be generated if a certain condition evaluated to true. Documentation for fine-grained auditing is available here.

Unified Auditing

Unified auditing was introduced in Oracle Database 12.1 (12c Release 1). Unified auditing consolidates audit records into a single location, combining audit data from Database Vault, Label Security, Data Pump, SQL*Loader, Recovery Manager (RMAN), fine-grained auditing, and audit records generated from unified audit policies. Unlike traditional auditing, unified audit policies may be conditional, may choose to audit only top-level statements, and are extensible to include context information not in the default audit trail. Documentation for unified auditing is available here.

Enterprise Manager Application Data Model

Enterprise Manager Application Data Model (ADM) is available in Enterprise Manager Cloud Control. ADM scans databases to locate sensitive data and helps drive Data Masking and Subsetting (DMS), Audit Vault and Database Firewall (AVDF) sensitive data audit reporting, and Transparent Sensitive Data Protection (TSDP) policies. Enterprise Manager Application Data Model (ADM) stores the list of applications, tables, and relationships between table columns that are either declared in the data dictionary, imported from application metadata, or user specified. ADM maintains sensitive data types and their associated columns, and is used by test data operations, such as data subsetting and data masking, to securely produce test data. Unlike DBSAT, ADM scans data contained within tables to find sensitive data.

ADM is included at no additional cost with Oracle Advanced Security, Oracle Database Vault, Oracle Label Security, Oracle Data Masking and Subsetting, and Oracle Audit Vault and Database Firewall. Use ADM to understand a database schema or schemas, including how columns relate to one another and which columns contain sensitive data. Documentation for ADM is available here.

Oracle Advanced Security

Oracle Advanced Security (ASO) is a database option that includes Transparent Data Encryption, RMAN backup encryption, Data Pump export encryption, encrypted Database File System (DBFS), encrypted SecureFile LOBs, and Data Redaction. Advanced Security is one of the oldest database options, tracing its roots to the Advanced Networking Option introduced in Oracle 7. Documentation for Advanced Security is available here.

Oracle Key Vault

Oracle Key Vault is a key management system supporting the Oracle infrastructure, optimized for use with Transparent Data Encryption. Key Vault provides continuous access to encryption keys with a multi-master fault-tolerant cluster architecture. Documentation for Key Vault is available here.

Oracle Database Vault

Oracle Database Vault is a database option that provides advanced access control capabilities. Database Vault is commonly used to enforce separation of duties, block administrator access to sensitive data, and enforce trusted path access to data. Documentation for Database Vault is available here.

Oracle Audit Vault and Database Firewall

Oracle Audit Vault and Database Firewall (AVDF) is a database activity monitor with heterogeneous capabilities – covering Oracle Database, Oracle MySQL, Microsoft SQL Server, PostgreSQL, MongoDB, IBM DB2, and SAP Sybase. Documentation for Audit Vault and Database Firewall is available here.

Oracle Data Masking and Subsetting

Oracle Data Masking and Subsetting (DMS) is a management pack for Oracle Enterprise Manager. DMS removes risk from databases by replacing sensitive data with artificial values. DMS can also be used to create subsets of a database – smaller copies with only a portion of the original data. Documentation for Data Masking and Subsetting is available here.

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Database Security and Regulatory Compliance
May, 2020
Russ Lowenthal, Database Security Product Management