Managing Security Across Disparate Database Technologies

Enterprise databases present several unique challenges for security and audit professionals. They are paradoxical, as they must securely store vast amounts of sensitive data while being broadly accessible, with a typical database processing transactions across several thousand connections per second.[1] Extrapolating this across several dozen or, in some cases, hundreds of databases operating concurrently, the ability to identify valid database traffic vs. malicious data exfiltration or other attack patterns quickly can become a challenge.

Approaches to securing flagship database products such as MySQL, Microsoft SQL Server and Oracle database can be obscure and difficult to understand.

ADAM KOHNKE | CISA, CISSP, GIAC, GPEN

Is the owner of XL Security Consulting and a cybersecurity architect for a leading and sustainable plastics manufacturer. With 15 years of experience in IT and cybersecurity, Kohnke has led large security teams and conducted numerous internal and external penetration testing engagements during his career. He is a regular contributor to the *ISACA® Journal* and *Infosecurity Magazine,* with articles covering IT audit and cybersecurity subjects. When not architecting cybersecurity solutions, he enjoys traveling with his family, camping, gardening and catching an in-person game of Dungeons and Dragons.

The worst-case scenario an enterprise faces is a publicly exposed database that contains large volumes of sensitive or regulated data. Malicious actors perform network scanning using tools such as Shodan, a search engine for locating vulnerable Internet-accessible devices. Research has shown that hundreds of connections will occur against publicly accessible databases within a few hours following their initial exposure to the Internet.[2]

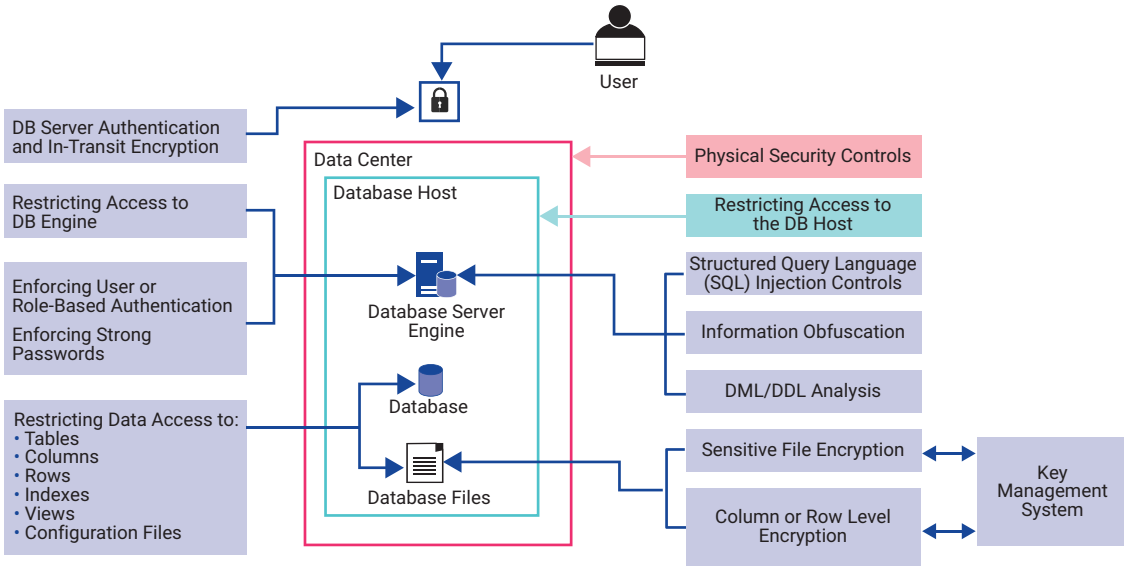
From a cybersecurity perspective, successful ransomware attacks against enterprise data occur every 11 seconds, costing enterprises on average US$283,000 per breach as of 2020.[3] Successful attacks against databases are one of the most troublesome attack vectors because administrative IT staff leave default credentials in place, ignore insecure network configurations, and leave unencrypted backups or logfiles available for abuse or theft.[4]

Enterprises must consider many aspects of database security, but initial efforts should focus on physically securing the operating environment, including the data center or network closets. They must ensure proper security of the physical server hosting enterprise databases by validating existing vulnerability patching processes and ensuring network traffic is properly inspected and filtered through a web application firewall (WAF) or similar technology that is specifically configured to inspect Internet traffic sent to web applications and their backend databases. **Figure 1** shows the foundational level of security each enterprise database deployment should strive to achieve and perpetuate.[5] Once a solid foundation is developed and established, additional controls should be assessed, selected and implemented to enhance security.

There are simplistic assurance activities enterprises may leverage to assess some of the elements within the database security architecture depicted in **figure 1**. While not exhaustive, high-level audit guidance via a linked database audit program is useful for assessing whether a database implements a secure default configuration, whether appropriate

FIGURE 1

An Example Database Security Architecture With Database Protections



Source: Adapted from Linster, M.; "Creating a Multi-Layered Security Architecture for Your Databases," *IT Ops Times*, 22 October 2019, *https://www.itopstimes.com/itsec/creating-a-multi-layered-security-architecture-for-your-databases/*.

user access management procedures are exercised if adequate encryption capabilities for data at rest and in transit are used, if adequate logging facilities are enabled, and if database replication or backup mechanisms necessary to support recovery are in place. Additional topics such as assessing data obfuscation, evaluating the effectiveness of change management and auditing the use of parameterized statements should also be covered in the audit program.

Secure Default Database Configuration

Secure default database configuration concepts can be applied during installation or postinstallation, but security and IT audit departments should seek to ensure the enterprise has instituted proper network configuration, default database configuration, expansive audit coverage, and user or system accounts are appropriately configured to optimize security. Initial audit focus for secure default configuration should ensure that databases require, or are structured to provide, input sanitization to reduce the chances of successful Structured Query Language (SQL) injection attacks. Audit focus should then expressly consider specific security topics by database type:

- **MySQL databases**—The mysql_secure_default command-line utility allows administrators to enforce security settings by default on deployed databases, requiring passwords for root users and removing insecure default databases.
- **Microsoft SQL servers**—A baseline configuration can be established using vendor-supplied tools or third-party solutions, but validation should occur to ensure that default system administrator accounts, default ports used to connect to the database instance and SQL server services such as SQL Server Browser are correctly configured. In many cases, this means default services are disabled appropriately and that default ports are changed to uncommon values that increase difficulty in footprinting or readily identifying critical database services running on network ports.
- **Oracle 19c databases**—A secure default configuration seeks to ensure that unused products and services are not installed, default user accounts (if not used) are disabled with nondefault passwords, database tables and indexes require logging when data manipulation language (DML) statements are issued, and the database is not configured to store credentials in cleartext within the database itself.

LOOKING FOR MORE?

- Read *Database Audit Program. www.isaca.org/database-audit-program*
- Learn more about, discuss and collaborate on audit and assurance management in ISACA's Online Forums. *https://engage.isaca.org/onlineforums*

Second only to data management, user access management is the most complicated aspect of database security due to the number of internal and external users who either require access or must have access restrictions in place.

The audit focus should also concentrate on how the secure default configuration is automated to the extent allowed by the associated database technology (e.g., MySQL, Oracle database) and how it is perpetuated throughout the database life cycle. Audit focus should then be concentrated on individual database services, network configurations, client configurations, and whether the most restrictive settings and configurations are enforced while concurrently allowing the business to operate within the database effectively.

User Access Management

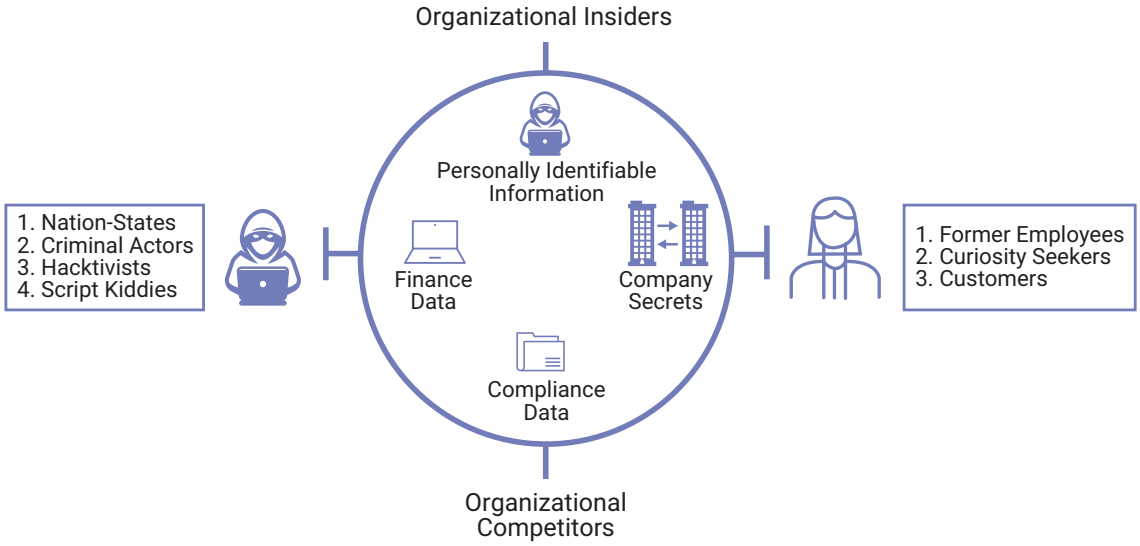
All modern versions of MySQL, Microsoft SQL and Oracle databases provide role-based access features to streamline the assignment and granting of permissions for users and applications requiring access to the database. Second only to data management, user access management is the most complicated aspect of database security due to the number of internal and external users who either require access or must have access restrictions in place, the individual permissions and combinations of access that can exist, and the tracking requirements for legal and regulatory purposes. **Figure 2** shows the typical classes of users that the enterprise security architecture must consider and around whom adequate access control must be placed.[6]

When database security is optimized, access should be restricted to trusted insiders, business partners, customers and employees with a need to access stored personal data on enterprise databases. Access controls should be effectively designed to keep inappropriate users such as competitors, cybercriminals or potentially problematic curiosity seekers out of the system. This is best achieved and simplified by instituting role-based access schemes that are defined by management based on job responsibilities and then logically controlled in each database. All access to the database should be formally requested, reviewed and approved, with routine access reviews performed to determine continuing need for access. Initial audit focus for database access should center on:

- **MySQL databases**—MySQL provides a proxy user function, which, when enabled, allows an external principal, such as a user or system, to assume an access identity defined at the MySQL database level. The status of the root-level user, who can

FIGURE 2

A High-Level Look at Enterprise Security Architecture Classes



Source: Adapted from Oracle, "What Is Data Security?" *https://www.oracle.com/security/database-security/what-is-data-security/.*

perform all actions on the database, should be determined ensuring that use of this account is heavily restricted and monitored in the instances where use is warranted. Aside from reviewing standard, named users, proxy users, root users and privileged database users should be routinely reviewed for the appropriateness of access.

- **Microsoft SQL Server**—The system administrator account is the default super administrator account and is part of the sysadmin fixed server role. Users in the fixed server and fixed database roles should be examined, as these roles provide extensive administrative access to database instances and the individual databases where role memberships are defined. The system administrator account should be renamed to obfuscate it and set to "disabled."
- **Oracle 19c databases**—Restricting database operations to nonprivileged accounts via the creation of Oracle home users is an essential aspect of default database access safety. Oracle databases running on Microsoft Windows machines can define a nonadministrative local user account on the host or within the Active Directory to manage Oracle database services. A review of the running Oracle services from Windows Task Manager will reveal the user account assigned these privileges. Further inspection of the group memberships and permissions assigned to this account will determine whether its permissions are excessive or unnecessary concerning database operations. For example, the account does not require domain administrator rights to operate Oracle databases.

Most cryptographic materials, such as a symmetric encryption key, have a set life cycle that does not exceed two calendar years following their creation before being rotated and destroyed.

The audit focus for user access management should determine whether multifactor authentication (MFA) is required for users making database connections and if strong password and lockout policies are enforced across the enterprise database estate.

Encryption

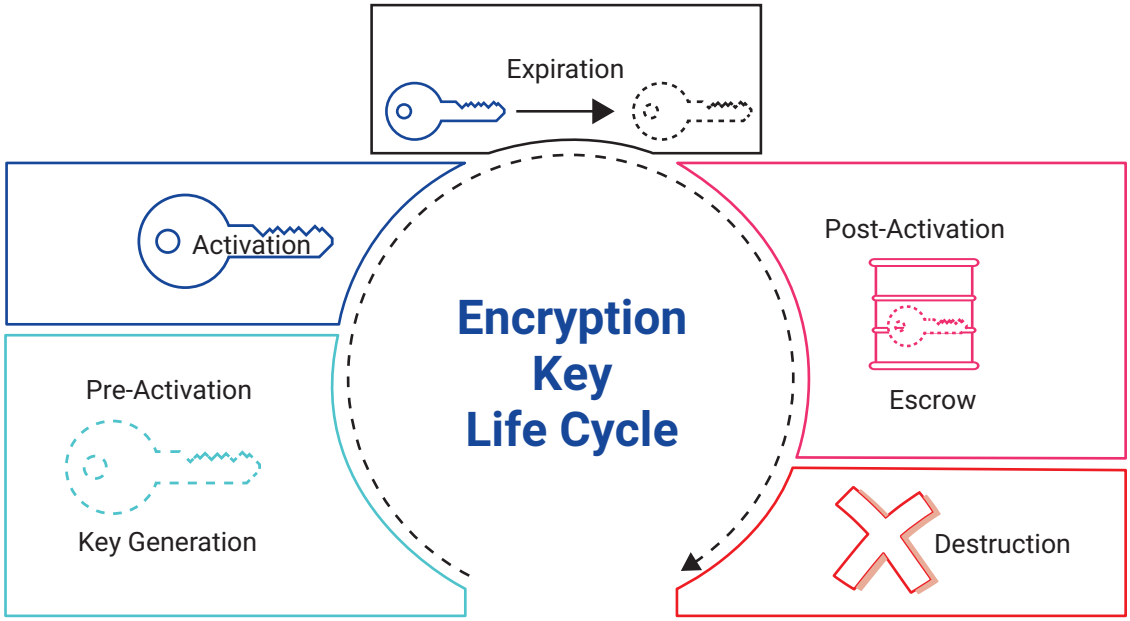
The goal of database encryption is to assure that the enterprise has configured its databases to enforce encryption for data at rest, encryption for data in transit, and column-level encryption and processes to ensure rotation of cryptographic materials such as keys or digital certificates. **Figure 3** illustrates the typical life cycle for enterprise database encryption keys, digital certificates, secrets or other cryptographic materials.[7]

Most cryptographic materials, such as a symmetric encryption key, have a set life cycle that does not exceed two calendar years following their creation before being rotated and destroyed. Initial audit focus should determine at what point in the database life cycle cryptographic protections are applied to the database or database tables and how cryptographic protections are effectively managed throughout the life cycle. This should be extensively validated for each key, certificate or secret to protect the database and its stored contents. Product-specific audit considerations for databases include:

- **MySQL databases**—Several storage engines such as Inno-DB are supported and can be deployed for each database. Through inspection of the "show engines" command, the auditor should understand which engines are active and for what purpose, how the data stored in the engine are encrypted, and if the strength of the mechanism is adequate (AES-128 vs. AES-256-bit encryption) and meeting enterprise requirements.
- **Microsoft SQL Server**—The Always Encrypted feature provides column-level encryption to restrict inappropriate users, including database administrators, from seeing sensitive data, such as credit card numbers, that may be stored in database tables. Audit procedures should either sample database table columns where the enterprise requires encryption and determine if an encrypted data value is returned or obtain and inspect screenshots from the Always Encrypted Setup wizard found within the SQL Server Configuration Management tool to determine which columns are presently encrypted and which do not meet enterprise requirements.

FIGURE 3

Recommended Steps in the Database Encryption Key Life Cycle



Source: Townsend Security, "The Definitive Guide to Encryption Key Management Fundamentals," *https://info.townsendsecurity.com/definitive-guide-to-encryption-key-management-fundamentals*. Reprinted with permission.

- **Oracle 19c databases**—The Transparent Data Encryption (TDE) mechanism is provided to encrypt database tables and resident data stored within. The TDE functions in tandem with an encryption key store called the Oracle Wallet to securely store and manage cryptographic material, such as tablespace encryption keys, certificates and user login information. To validate the status of TDE encryption for a database and its tables, the command "SELECT * FROM DBA_ENCRYPTED_COLUMNS;" can be run from SQL Plus and further inspected.

Logging

Database logging is essential for understanding operational and security events occurring within the database. In some instances, the availability and integrity of logs are also needed to perform database recovery or replication activities effectively. Segregation of duties (SoD) is also a critical audit consideration for database logging, as users responsible for database administration should not be provided access to modify or destroy database logs—these permissions should be provided to a security or storage administrator. Initial audit focus should consider whether detailed audit logs are generated supporting after-the-fact investigations, including the date and time of logged events, source and destination of logged events, success or failure of action, and principal performing actions against the database. Product-specific audit considerations for database logging include:

> Insecure log files should be offboarded to cold storage and deleted immediately upon data retention expiration schedules being reached.

- **MySQL databases**—Ensuring that the binary log is enabled via inspection of the "SHOW VARIABLES LIKE 'bin_log';" command and ensuring that binary logs are appropriately replicated to the intended database using a secure connection, with encryption at rest enabled, are essential.
- **Microsoft SQL Server**—The provided Log Ship function allows the secure offloading of critical database log files to alternate repositories. The Log Ship defines the relationship between databases in a Log Ship configuration as a publisher (source database) and a distribution (destination database).

Audit steps for Log Ship configuration should seek to determine the appropriateness of access to the destination databases and ensure that strong encryption is used in transit and at rest for the stored log data.

- **Oracle 19c databases**—The provided NOLOGGING clause can be issued against individual statements such as CREATE TABLE, ALTER TABLE or ALTER INDEX. These commands are present to reduce potential database performance scenarios, and their use cannot be efficiently restricted. Besides the obvious risk of not correctly documenting events of interest, as CREATE or ALTER statements are issued, the absence of certain logs in the Oracle redo log can have negative implications for effective database restoration efforts. The use of NOLOGGING can lead to recovery failures and errors. Audit focus on the careful use of NOLOGGING revolves around assessing user training for good scenarios or statements where NOLOGGING is acceptable and monitoring activities performed to identify inappropriate uses of this logging clause.

Additional logging controls should be assessed to determine if the enterprise achieves database log integrity via inspection of access control lists; whether immutable replication to a secondary, secure data repository is occurring; and if the log files themselves are leveraging strong encryption at rest and in transit. Audit focus for logging should also concentrate on whether the appropriate number of log files are being generated under defined scenarios that may impact logging, including if the Oracle database crashes and restarts or in instances where encryption is enabled or disabled. For the latter example, it is pertinent to ensure that previously unencrypted log files containing sensitive information such as passwords do not persist in logs containing encrypted data. Insecure log files should be offboarded to cold storage and deleted immediately upon data retention expiration schedules being reached.

Conclusion

Security risk is prevalent in enterprise database deployments and potential audit approaches to assess the state of controls. An in-depth and methodical audit approach is required to evaluate each layer of the database security architecture to ensure that enterprise, customer and business partner data remain secure while also providing users the ability to conduct efficient data transactions. IT audit departments can deliver on their promises to serve as trusted advisors by becoming more familiar with the database technology in their organization and assessing these deployments using security best practices and controls. Malicious activity is increasing, with databases serving as prime targets for hacking and advanced persistent threats. IT audit departments can help shift the threat landscape in favor of the enterprise by adopting new audit procedures to protect valuable applications and their core databases.

IT audit departments can help shift the threat landscape in favor of the enterprise by adopting new audit procedures to protect valuable applications and their core databases.

Endnotes

1. Jackson Higgins, K.; "Hacker's Choice: Top Six Database Attacks," *Dark Reading*, 8 May 2008, *https://www.darkreading.com/risk/hacker-s-choice-top-six-database-attacks*
2. Zorz, Z.; "With Database Attacks on the Rise, How Can Companies Protect Themselves?" *Help Net Security*, 14 October 2020, *https://www.helpnetsecurity.com/2020/10/14/securing-exposed-databases/*
3. Ireland, S.; "Revealed: The True Cost of Rising Cyber Attacks," *CEOWORLD Magazine*, 2 February 2022, *https://ceoworld.biz/2022/02/02/revealed-the-true-cost-of-rising-cyber-attacks/*
4. *Op cit* Jackson Higgins
5. Linster, M.; "Creating a Multi-Layered Security Architecture for Your Databases," *IT Ops Times*, 22 October 2019, *https://www.itopstimes.com/itsec/creating-a-multi-layered-security-architecture-for-your-databases/*
6. Oracle, "What Is Data Security?" *https://www.oracle.com/security/database-security/what-is-data-security/*
7. Townsend Security, "The Definitive Guide to Encryption Key Management Fundamentals," *https://info.townsendsecurity.com/definitive-guide-to-encryption-key-management-fundamentals*