Deep Science Publishing, 2025
https://doi.org/10.70593/978-93-7185-191-6_4

Chapter 4: SQL security and regulatory compliance: Implementation of RBAC and audit logs

Mohanraju Muppala

1. Introduction to Security and Compliance

Diversity is the hallmark of every security and compliance strategy. Security capabilities may address data at rest or in transit; alternatively, compliance requirements may be driven by governance regulations such as Payment Card Industry Data Security Standard (PCI DSS), the Health Insurance Portability and Accountability Act (HIPAA), or the European Union's General Data Protection Regulation (GDPR) [1-3]. HIPAA compliance, for example, requires that persistence layer data be encrypted (both in transit and at rest) in addition to enforcing role-based access controls and maintaining SQL Audit Logs. GDPR assigns comparable requirements yet focuses heavily on audit reporting and the provision of easily accessible detail regarding data provenance [2,4,5].

Development of a compliance strategy centers on three main components: establishing an enterprise framework centered on meeting the technical requirements dictated in the original law or regulation; training staff on the requirements and procedures developed to address them; and, finally, ensuring adherence to governance standards through monitoring and reporting. Role-based access control (RBAC) is a widely implemented design pattern for regulating access to logical objects and is typically reinforced by supporting control structures in the physical data layer. Row-level security (RLS) is a more fine-grained approach that delivers similar protections yet operating at the record instead of the table level [6-8].

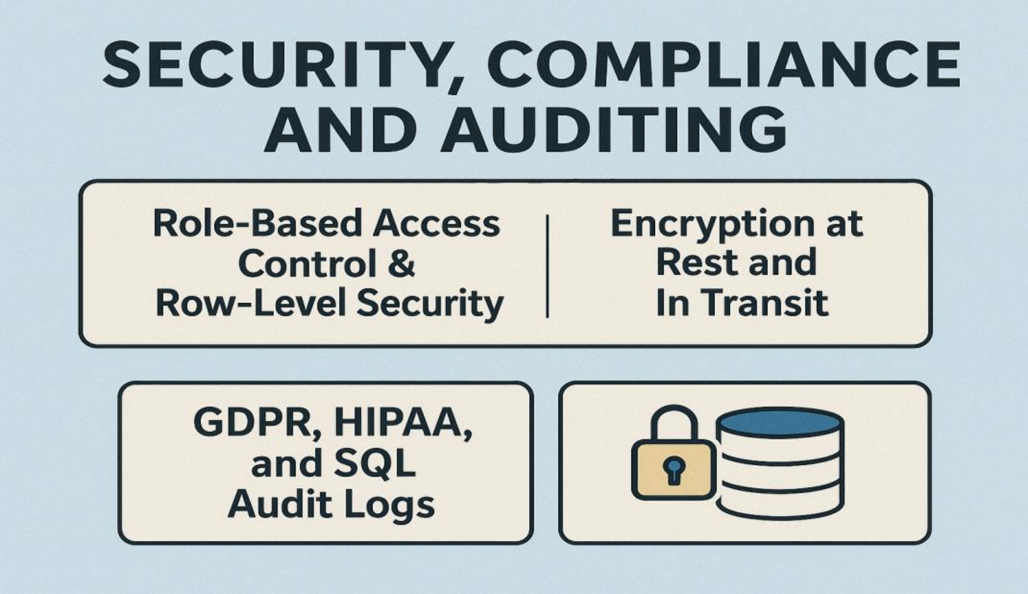
SECURITY, COMPLIANCE AND AUDITING
Role-Based Access Control & Row-Level Security
Encryption at Rest and In Transit
GDPR, HIPAA, and SQL Audit Logs

Fig1. Security, Compliance and Auditing

2. Role-Based Access Control (RBAC)

In recent years, Role-Based Access Control (RBAC) of Enterprise Content Management Systems (ECMS) has become an essential mechanism for ensuring compliance with security requirements and meeting industry regulations such as HIPAA, FDA, GLBA, CMMC, IRS 1075, GDPR, and other laws related to the security of personally identifiable information. RBAC assigns access permissions based on users' roles within an organization, thereby providing a control mechanism that prevents users without the necessary privileges from accessing sensitive data and commands.

RBAC represents a policy-neutral logical access control framework that supports fundamental security principles such as least privilege, separation of duties, and the concept of a trusted computer system. The main elements of RBAC are users, roles, permissions, and sessions. Users—persons or other machines—have access to perform specific operations. Roles are job functions or titles within an organization that define authority and responsibility. Permissions are approvable requests to perform operations on objects, and sessions are mappings between a user and an activated subset of the roles to which the user is assigned. Systems that support RBAC implement role engineering and analysis, which involve creating role hierarchies and defining how roles inherit permissions from other roles.

2.1. Overview of RBAC

Role-based access control (RBAC) is a common method for restricting system access to authorized users [9,10]. It is used by enterprise IT administrators and their customers to regulate who can view and alter resources within an organization. RBAC follows the principle of least privilege by assigning users only the permissions needed for their work, minimizing the risk of data leaks. The NIST IR 7438 report defines RBAC as a method of restricting network access based on users' roles within an organization and outlines its taxonomy, components, implementation process, and challenges. Additionally, it defines the term reuse-control for improving the security of sensitive data.

Enterprise services use RBAC to ensure a secure, user-friendly environment in which access to network data is restricted based on user rights. Secretaries can be given read-only access to documents, enabling them to disseminate information to the public while preventing them from editing the files accidentally or, intentionally. When users are assigned a role or roles, the permissions defined for that role or roles become active. Access permissions granted to a role enable transformation and action on business assets that fall within an analyst’s scope of interests. However, if a role does not include rights for it, a user cannot view or modify an object, providing the security boundaries that prevent unauthorized data modification.

2.2. Implementing RBAC

Role-Based Access Control (RBAC) is a widely implemented security principle used to protect systems from unauthorized access. In an RBAC system, roles that correspond to business functions are created and users can be assigned roles. Users inherit the privileges that are associated with their role and can carry out activities without restriction. Users who do not have a specific role are explicitly prohibited from carrying out the activities related to that role.

Setting access privileges can be a major challenge. For example, it can be difficult to determine which users require the ability to carry out a particular activity and which ones do not. It can also be complex to define appropriate roles that reflect the structure of the business and to identify users who work together so that they operate within the same role [11-13]. Because of this complexity, organizations are responsible for establishing access privileges and assuming the risk if either a violation of segregation of duties occurs or excessive numbers of users possess the ability to carry out sensitive or critical activities.

2.3. Benefits of RBAC

Role-based access control (RBAC) is a method of restricting system access to authorized users. RBAC is a dimension of enterprise security that accommodates multiple implementations of mandatory access control (MAC) and discretionary access control (DAC). Roles are created for various job functions, and members are assigned accordingly. Through those role assignments, users acquire the computer permissions of their assigned role. The principles of least privilege and separation of duties are realized through the appropriate assignment of roles [2,14-17]. RBAC is considered an efficient way to implement security in large networks with hundreds or even thousands of users. Users predefine the roles within the system and then create new users and assign them to the appropriate role(s). From there, roles can be added to and removed from users according to the system administrator's discretion.

In Multilevel secure (MLS) databases, RBAC is usually employed to enforce users' assigned security levels—or security roles. They are then assigned to classes of permissions, or what accesses their roles allow them to make. The user may then be granted or denied access according to the policy of the specific system. The simplest model of this category is a lattice-based access control model, where clearances are represented by elements of a lattice. RBAC describes what operations a user can perform, but it is not concerned with the decision of granting authorization to certain users because of their family ties.

2.4. Challenges in RBAC

An RBAC system faces some specific challenges. Firstly, two administrative roles are defined for performing administrative activities: one for assigning user permissions and the other for assigning roles to users. Such a setting may not be appropriate for large systems where a number of administrators are responsible for handling user permissions and role assignments. Secondly, support for user hierarchy is not available. Finally, permission assignment to roles cannot be restricted to a certain set of objects within the system.

The first challenge of the RBAC model affects the Object-Oriented RBAC model. In some real organizations, multiple administrators might be responsible for setting user permissions and role assignments for a number of users. However, in the Object-Oriented RBAC model, only two super-users classified as SU-U (super-user user) and SU-P (super-user permission) can assign users to roles and permissions to roles, respectively.

3. Row-Level Security (RLS)

Access control is a fundamental part of any framework for security and compliance in modern organizations. Within data analytics, Row Level Security (RLS) is the technique that restricts table rows to only a subset of users. It is implemented in Microsoft Power BI, a data analytics software for visualization and business intelligence. The Power BI Service is the cloud platform for sharing and collaborating on Power BI content within and outside organizations. Power BI has a dynamic, built-in RLS capability to limit data visibility in reports and dashboards that are shared with users and groups.

The built-in RLS feature imposes restrictions on individual tables and supports filtering on individual columns. The filters are used for granular, cell-level security only as a public feature from Power BI premium. Furthermore, the accessibility feature supports filtering on individual table rows only when the relationship between tables is a direct one, i.e., without any intermediate tables between the table with the role and filter.

3.1. Understanding RLS

Understanding of RLS

Row-level security (RLS) allows a business to control access to rows in a database table based on the characteristics of the user executing a query. This makes it easier to maintain a single copy of data, rather than one for each user, because users will only see the rows they are allowed to see.

3.2. Implementing RLS

Role-level security (RLS) provides access control over rows in tables and views. Row-level access filters can optionally block direct access to selected rows for any REST API action, for example, via data virtualization.

Cloud-level security including cloud identity federation is supported through single sign-on with Microsoft 365 and Azure Active Directory (AD). Azure AD identities also provide optional data access controls via role assignments at the resource, workspace, or collection level, offering an additional layer above RLS roles.

3.3. Use Cases for RLS

Row-Level Security (RLS) is driven by real-world needs, such as compliance with regulatory frameworks mandating specific types of data segregation. Another common scenario is when organizations possess sensitive information and want to restrict access accordingly; for example, people managers may be granted access to view only data concerning their direct reports [9,18-21]. Direct scenarios of this kind often apply RLS on user attributes, which can be derived either from the database itself or from an external Identity Provider (e.g., Azure AD).

Beyond these direct settings, many organizations favor the creation of custom roles in the Identity Provider. This approach yields advantages such as tighter security controls—like Multi-Factor Authentication (MFA) enforcement—and simplifies membership management through established organizational processes. In these situations, roles and RLS are defined based on groups or applications. Unlike user attributes, group names are not retrieved through direct mapping but by querying the user membership of the groups.

3.4. Limitations of RLS

A limitation of RLS is that policies are tied to individual tables, and it is not possible to apply policies spanning multiple tables. For instance, consider the expense report form and the employee table [22,23]. Employees who are managers may only view the form when the reporting employee is a direct subordinate. No RLS function can be associated with these two tables to perform this check; it needs additional logic to traverse the hierarchy of employees and their direct subordinates. This can be performed through table-level triggers, but the performance of the associated form suffers as a consequence.

Another limitation is that RLS policies cannot reference the pg_shadow system catalog; for example, they cannot use Current_User. The built-in functions Current_User and Session_User cannot be used in policy definitions to associate users with records [24-26]. This is because these functions deterministically change the output of these policies as the records are being scanned. This constraint arises because internally the system uses set containment on the output of the RLS policy definition. Finally, the system does not permit the use of the WITH CHECK clause with a USING clause.

4. Encryption in Data Management

Data encryption converts data into a format that is unreadable by an unintended recipient. Therefore, encryption safeguards data by scrambling it, and it uses an algorithm to dictate how the original information becomes scrambled so that only someone with the key can unscramble the ciphertext and obtain the original data. The algorithms that encrypt and decrypt data are called ciphers. The transformation performed by a cipher depends on the values of the encryption key, which are used by the encryption algorithm in both the processes of scrambling and unscrambling the data.

Although it is necessary to share the encryption key with the recipient to unscramble the ciphertext, exposing the encryption key to other people should be avoided because anyone who has the key can also decrypt the original data. Symmetric key encryption encrypts and decrypts plaintext with the same key, whereas asymmetric key encryption uses two keys: the public key encrypts, and the private key decrypts the information.

When data is at rest, it can be stored in a list or file on a hard drive or SSD. Encryption protects data at rest by stopping unauthorized access to the drive, even if it is physically removed. When data is transmitted across a network, the sender can use encryption to ensure privacy and prevent unauthorized changes. For data in use, encryption provides such protection by placing data inside a homomorphic encryption scheme.

4.1. Encryption at Rest

Encryption at Rest is the practice of Securing Sensitive Data on Secondary Storage using an Encapsulating Wrapper that Ensures Confidentiality. Encrypted Persistent Disks in Google Cloud Platform (GCP) safeguard data at rest using cryptographic methods, offering an additional layer of defense for data stored on persistent disks. Data written to the persistent disk, snapshot, or image is encrypted before being written to the physical media.

Data in the persistent disk cache and snapshots that are stored in Cloud Storage are also encrypted. This encryption process occurs transparently, without any behavioral changes or additional operational steps from the user's side. Encryption and decryption activities happen automatically during data write to or read from persistent disks. Each persistent disk is encrypted with a distinct key that is further protected by a regularly rotated set of master keys.

4.2. Encryption in Transit

Encryption methods disguise data by converting it into a non-human-readable form, called ciphertext, using an encryption key [27,28]. This ciphertext is only decipherable using the appropriate decryption key. Encrypting data in transit protects it from a man-in-the-middle attack where a malicious actor intercepts the data stream between two communicating parties.

Data in transit is particularly [vulnerable to passive and active attacks]:HTTPS, SSH, SFTP and other similar protocols use data-in-transit encryption techniques such as TLS (Transport Layer Security) to secure the connection. Other protocols such as SSL (Secure Sockets Layer) and WPA2 (Wi-Fi Protected Access) were used historically but have been deprecated in favour of more secure protocols like TLS.

4.3. Best Practices for Encryption

Encryption adds a layer of security to stored data. It can be used alone or in conjunction with a combination of other best practices mentioned in the preceding paragraphs to provide a strong line of defense against data breaches. Because data breaches frequently involve the unauthorized collection of personally identifiable information (PII)—such as credit card numbers, social security numbers, and birth dates—the United States government requires all public and private organizations—regardless of industry—to comply with data encryption regulations.

Sensitive data stored within a relational database is best protected by disk-level, file-level, column-level, or data-level encryption. With disk- or file-level encryption, data is encrypted at the operating system level and does not require any modification of the application. Database passwords and passwords for application access should be encrypted. Passphrases, which are often used as passwords for system users, should also be encrypted. And it is particularly important to encrypt column-level data when encryption is only needed for specific columns in a table. This is accomplished by defining the encrypted columns as binary data and providing encrypt and decrypt application programming interface (API) functions as appropriate. Finally, when disks are frequently shared (such as within a cluster or cloud-based backup), data-level encryption provides adequate protection for sensitive information.

4.4. Regulatory Compliance and Encryption

In the current age of digital transformation, nearly all business sectors have become digitalized [19,29-31]. As a result, companies accumulate and store massive amounts of data in cloud services and remote databases for analysis and decision-making. However, users whose data has been uploaded to these public cloud services face the inherent risk of privacy exposure. Despite being secured with access control measures, the cloud remains a publicly accessible environment vulnerable to data attacks by cloud administrators, foreign users, and hackers. To mitigate the risk of privacy leakage, data owners often encrypt their sensitive data before outsourcing it to the cloud. Nevertheless, a fundamental requirement for many cloud services is to allow the cloud to operate directly on encrypted data and return encrypted results to the users.

Cloud data security and privacy are not isolated concerns; law and regulation also govern these issues. Many countries and regions have enacted legal frameworks, such as the General Data Protection Regulation (GDPR) in the European Union and the Gramm-Leach-Bliley Act (GLBA) in the United States, to ensure enhanced protection of citizen data. These laws aim to control and prevent data breaches and provide legal remedies for the affected individuals. The subsequent sections explore how these legal regulations and requirements influence data protection and privacy, delving deeper into the scope of security concerns.

5. Regulatory Frameworks

Several laws have been drafted to regulate the management of information and data. The Federal Information Security Management Act (FISMA) requires agencies to develop, document, and implement controls for information systems, requiring third-party evaluations of information security formal certifications and accreditations for all federal IT systems. The Gramm-Leach-Bliley Act (GLBA) requires financial institutions to explain their information-sharing practices to their customers and to safeguard sensitive data. It has two principal rules, the Financial Privacy Rule requires financial institutions to provide their customers with privacy notices that explain what kinds of information the institution collects and how it is shared; the Safeguards Rule requires financial institutions to implement security programs to protect customer information. The Health Insurance Portability and Accountability Act (HIPAA) stipulates that healthcare and healthcare-related organizations must protect patient information and protected health information. Similar to FISMA, HIPAA requires regular third-party evaluations, formal certification, and accreditation of information system security.

The PCI Data Security Standard (PCI DSS) is a proprietary information security standard for organizations handling branded credit cards from the major card schemes [32,33]. The Sarbanes-Oxley Act (SOX) requires public companies to guarantee the accuracy and reliability of their financial information systems, requiring the evaluation of internal controls and risk management procedures with annual report disclosures. The Gramm-Leach-Bliley Act (GLBA)—requires financial institutions to explain their information-sharing practices to their customers and to safeguard sensitive data under the GLBA privacy and security rules. Additional examples include the Control Objectives for Information and Related Technology (COBIT) from the Information Systems Audit and Control Association; also, comprehensive laws protecting consumer information such as the EU Data Directive, the Canadian Personal Information Protection and Electronic Documents Act (PIPEDA), the California Online Privacy Protection Act (CalOPPA), and New Zealand's Privacy Amendment. The General Data Protection Regulation (GDPR) is a legal framework that sets guidelines for the collection and processing of personal information from individuals who live in the European Union (EU). Information governance is an outcome of the requirements imposed by these laws and regulations.

5.1. General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) is a regulation on data protection and privacy for all individuals within the European Union (EU) and the European Economic Area (EEA). The regulation additionally addresses the export of personal data outside the EU and EEA. Its primary aim is to empower all EU citizens by giving them control over their personal data while simplifying the regulatory environment for international business through a unified regulation. The GDPR became enforceable on 25 May 2018. It replaces the 1995 Data Protection Directive, marked as DP95.

Upon determining that maintenance of adequate data protections is necessary in areas cited under GDPR, Amazon Personalize invokes additional controls based on the GDPR service model. These controls provide the supervisory authorities with audit and complaint handling features, and the data subjects with rights to access, rectification, erasure, and restriction of processing. Furthermore, the GDPR supports the correctness and limitation of data, and provisions for transfer to third countries. Access to personal data is provided with a full audit trail, and charge-free measures are in place. As an example, designating a particular dataset as "customer PII" will activate the associated GDPR controls and responsibilities for Amazon Personalize.

5.2. Health Insurance Portability and Accountability Act (HIPAA)

The Health Insurance Portability and Accountability Act of 1996, known as HIPAA, deals with electronic protected health information (EPHI) and includes several important rules. The Privacy Rule focuses on the protection of EPHI and establishes controls on who can access and distribute it. The Security Rule prescribes specific technical, physical and administrative safeguards for organizations handling EPHI and defines additional mechanisms for unauthorized personal access. HIPAA is a difficult law to satisfy since it requires controls within the organizational environment, such as background checks, that are not possible to enforce in an Infrastructure as a Service (IaaS) or even a Platform as a Service (PaaS) cloud environment.

One difference between HIPAA and other regulations is that it not only categorizes the governance characteristics required for regulatory compliance, but also requires that the regulation is applied across a continuum of business roles, organizational classifications, geographic locations and services. The consequence of this is that an Information System must be able to satisfy a particular requirement depending on the business context under which it operates.

5.3. Comparison of GDPR and HIPAA

Both GDPR and HIPAA were designed to protect individuals against the theft and misuse represent data breaches. They provide specific security and facility requirements, such as HIPAA's Security Rule for Data, Device, Network and Facility Management and the GDPR's encryption and pseudonymisation requirements under the Security of Processing Articles. Nevertheless, security requirements, such as encrypting personal data in transit and at rest, hold strong under both regulations. \protect\hyperlink{ref-AcsSzappanosCentaur202011}{Acs and Szappanos (2020)} propose a GDPR and HIPAA comparison based on the Hospital Data Transfer Protocol to identify compatible healthcare security technologies complying with both regulations.

In the United States, the federal Health Insurance Portability and Accountability Act (HIPAA) safeguards individuals' medical data during creation, use and disclosure. As regulations on the use of individuals' personal data, the Health Information Technology for Economic and Clinical Health Act (HITECH) and the American Recovery and Reinvestment Act (ARRA) extended protection to data privacy and security requirements.

5.4. Implications for Data Management

The General Data Protection Regulation (GDPR) imposes stringent requirements on the processing and handling of personal data. While GDPR's recent introduction together with its non-technical nature inhibit the adoption of privacy-preserving technologies, the regulation serves as a catalyzing factor by raising awareness of privacy risks and encouraging the construction of dedicated privacy-preserving shields. The different provisions enshrined in the regulation, either explicitly expressed or implied for respecting the fair treatment of the data subject, span a wide spectrum that touches virtually every part of a company's architecture, from governance and staff training to the underlying security architecture.

Respecting that palimpsest of constraints in the design of a dedicated security architecture is a prescriptive approach, yet employing the collection of requirements as a checklist for assessing an architecture is an instructive exercise. Categorizing GDPR's canons according to security properties not only facilitates this architectural assessment but can also shed light on inconsistencies among the regulation's directives. The data subject's right for transparency and the company's duty to guarantee security appear as security properties themselves. Strikingly, contrasting requirements are implied: whereas the right to be forgotten demands the permanent erasure of the data subject's personal data, the accountability principle—in combination with the duty to maintain records of processing activities—implies their retention.

6. SQL Audit Logs

Oracle Cloud also logs SQL commands including creation of Database users. Audit records of Data Management Activity are recorded by Cloud Audit which can be viewed in the OCI Console. A set of predefined view commands in the DB Management Service lets the user query the DB Management Activity, indicating who accessed the database and when. These views can be accessed using the Service Console or the OCI CLI.

Examples of SQL commands for setting auditing at the user or DB level include CREATE SESSION and ALTER USER statements. Oracle Cloud Audit Logs also track Database account activity. They contain service details including whether the management activity was successful. Users can check for successful and failed login attempts. For greater control, auditing can be enabled at the user level with the DB Management Service, allowing monitoring of specific users who created or modified other user accounts. The set of commands that can be audited at the Service level includes DISPLAY, UNDISPLAY, CREATE USER, ALTER USER, DROP USER, and GRANT/REVOKE.

6.1. Importance of SQL Audit Logs

The audit logs generated by the MySQL Audit Plugin provide a wealth of information, including events such as database operations (e.g., DELETE, SELECT) and general server information (e.g., server start and stop). They also include details about SQL operations and the general login and logout of users to the data source. Two basic challenges exist in creating a repository for these logs: deciding how to project an audit log and how to store audit logs according to the need for further analysis.

Audit logs are usually stored in a format that is capable of being queried; this depends on the format of the operation such as a DELETE. These logs can be secured by encrypting or digitally signing the entries to ensure the integrity and confidentiality of the audit data stored. The audit data and its format depend on the general security level of the application scenarios. The amount of stored audit should depend on the requirements of securing the application and be considered over the profitability of a potential attack on the data source. A deep analysis of the audit data can help to generate access control policies.

6.2. Implementing SQL Audit Logs

Auditing SQL Server logs would allow you to control and review the information related to activity on the database server. SQL Server offers a dedicated audit feature, called SQL Audit, which enables the auditing infrastructure at the server level. Once enabled, you can create audit specifications—at the server and database levels—by including specific groups of actions or individual actions. When SQL Audit is enabled, it creates a trail of information that you can filter by the actions performed.

The SQL Server Audit architecture consists of three components:
- Audit: Defined at the server level and used to specify the audit destinations and the name of the audit.
- Server audit specification: Associated with the audit component and used to select the server audit action groups.
- Database audit specification: Associated with the audit component and method used for audit granular control.

To implement auditing, you must first create an audit. You then need to create one or more server audit specifications or database audit specifications that are associated with the audit, which can be configured to record groups of actions or specific actions. Finally, the audit is enabled and starts capturing information, which can be either logged to a file, the Security event log, or the Windows Application event log. These elements can be opened, and their details reviewed using the SQL Server Management Studio GUI through Server Objects ▶ Audits, Server Audit Specifications, and Database Audit Specifications.

6.3. Analyzing SQL Audit Logs

Querying individual SQL files on the command line or filtering them by importing them into a text editor is a tedious task. Moreover, using the .csv files is not very effective either, because the csv format is effective for static data, but less so for streaming data, because it does not support timestamps, and it lacks compliance features such as denial or acceptance of a violation.

Instead, it is better to use a proper stream processor to analyze streaming data. Such a tool could be Apache Kafka, Apache Samza, or Apache Flink. Given that the .csv files are currently the authoritative source, the first step is to convert the csv data into a data stream. This can be achieved by implementing a Data Pipeline that reads the data from SQL audit file log and publishes it to a data streaming topic, such as Apache Kafka. The pipeline reads the new logs every five minutes from the .csv file and publishes them, preserving the time information of the actual events.

6.4. Best Practices for SQL Auditing

Auditing SQL Server databases is a vital security task that reveals valuable insights about operations impacting sensitive information. SQL Server 2014 offers a broad set of auditing capabilities that empower users to monitor instances and their data. Together with Windows Server 2012, it supports the generation of audit logs within the Windows event log. Management can configure audit settings at the database level or across the instance by explicitly specifying the actions and grouping those that must be audited. Auditing SQL activity and reviewing audit security logs are crucial activities for maintaining control over environment changes and determining the timeline of server activities.

When planning and implementing auditing tasks for SQL Server installations, certain best practices should be considered. Auditing is enabled at the SQL Server instance level and saved in defined destinations, including files, Windows Security event logs, or Windows Application event logs. Establishing server-level auditing first allows operations to be audited on a granular level within the server, whether for database creation or modification or for particular operations on the database data itself. Such auditing provides not only enterprise-level security but also granular security for specific entries in the database. Most auditing functionality can be controlled at the database level; consequently, it is recommended to leverage the database audit specification.

7. Compliance Strategies

The Preparatory Phase of ISO 27001 is straightforward. Even in large organizations, mapping existing controls to Annex A controls is rarely an insurmountable challenge. ISO 27002 easily achieves the required assurance for internal audit purposes [34-36]. For a thorough internal audit, more evidence, sometimes supported by other sources such as ISO 9001, is used. When organizations are mature in aspects related to the control of classified information in the various disciplines in the organization, the Internal Audit is seen as the best way to maintain the level of assurance.

When an organization is mature, the implementation of type-2 reports is also feasible. The observation is that the Public Service schedules tend to repeat the specific controls of companies audited by service providers. The ISO standard's Annex A sometimes becomes a guarantee for the requested levels of confidentiality at a service provider.

7.1. Developing a Compliance Framework

The difficulty of verifying the security of a program stems from the fundamental challenge that programmers are not always aware of the paths that the program can take, especially in concurrent or distributed systems, or even in smart contracts. In formal terms, they are unaware of the specification of the implementation.

Security clearly requires a formal framework, and every formal framework requires a logic. To ensure systematic accountability in the database process of a leading financial management company, it is imperative to first establish a compliance framework. The symbolic logic required for such a framework is implemented by a formal language for database operations, together with a set of logical rules that assert the legality of the actions.

The formal language defines the basic operations that mid-level or executive-level employees can perform on the database, as well as the contextual parameters in which these actions can be carried out. For example, it specifies the properties that the table must have, the format that the input must respect, or the identity of the employee. The rules govern the legality of the operation considering the current state of the database and the previous operations, such as allowing insertions that do not violate the uniqueness of an item or dismissals for different reasons.

7.2. Training and Awareness Programs

Training and Awareness Programs play a crucial role in addressing cybersecurity risks by providing bespoke training courses and continuous awareness communications, all designed using the latest cyber-threat intel. The Remediation Fleet enjoys full access to these offerings. Every Member is encouraged to actively participate and leverage available resources effectively.

Through an innovative and evolving training program, personnel gain the ability to immediately take decisive mitigation actions against newly identified threats. Daily awareness bulletins and dedicated 'Deep Dives' keep everyone informed about specific cyber-attack methods and Tactics, Techniques, and Procedures (TTPs) that have been observed in the wild. The visibility provided by the Training and Awareness Platform supports Members in building knowledge, confidence, and a proper security mindset.

7.3. Monitoring and Reporting Compliance

With Cloud, data breaches have become both highly sophisticated and rampantly distributed across the planet. It is even more difficult on Cloud as few organizations control the entire data security stack. Most organizations struggle with monitoring data access and usage. Product engineers lack the context to generate usage reports because data and consumer of data often reside in different platforms. As a result, business stakeholders—such as CISO, CCO, and CEO—are dissatisfied with vendor usage reports and their lack of meaningful information. Without such monitoring reports, it is difficult to corroborate or dispute breach notification or breach impact reports.

Cloud and digital marketing providers sometimes over-expose sensitive data in breach notifications, while Cloud infrastructure providers tend to under-expose breach details. Without monitoring reports, it is challenging to dispute or verify claims made by either party. The Data Security Law defines explicit requirements for access and usage monitoring. It also mandates in-depth reporting to business stakeholders regarding potential unauthorized access, which can be utilized for conducting forensic investigations.

8. Future Trends in Security and Compliance

As data volumes grow and organizations grapple with an ever-expanding compliance landscape, major initiatives increasingly focus on the efficient collection and analysis of large quantities of metadata. However, merely gathering metadata is insufficient—it also requires the integration of external data sources to provide context and guide future operations. For example, involving the public by applying data science to health organizations' COVID-19 dashboards deepened community engagement and enhanced compliance supervision. Similarly, organizations are pursuing broader business integration, combining data classification and risk scoring with internal controls and broader governance efforts. ThyssenKrupp's Oil & Gas information governance journey involved a broader risk management initiative that integrated data discovery and controls in alignment with their risk framework.

Subject matter experts tackle various technical, procedural, and governance challenges related to security and compliance. These efforts span optimizing classification results and evaluating testing methods; integrating encryption and tokenization with data classification, user behavior analytics, and permissions management; unifying diverse data sources—online, offline, and external; employing automation to address the tedium and repetition of compliance procedures; and providing clear decision support to counteract decision fatigue and maintain a consistent compliance posture. Looking ahead, controls that boost productivity for both business and IT users will emerge as vital elements of data protection and compliance solutions.

8.1. Emerging Technologies

The emerging technologies responsible for the information explosion are the key technology in all sectors of industry. To manage the large volume of data generated and provide security and compliance governing the practices of formulation, collection, storage, integration, cross-referencing, processing, and dissemination of data becomes an important issue. Long-term data management for the emerging technologies requires highly scalable storage architecture, extremely fast retrieval of data in heterogeneous environment, and interoperability in a highly distributed fashion. In catering to the requirements of electronic data, various types of data management systems are being proposed, developed, and implemented.

Built-to-purpose transactions and workflow of the transactions inherent in the system are implemented and operated usually using workflow management software. Data classification is implemented by the systems as defined in the data lifecycle. Appropriate procedures are prescribed and practiced to ensure the protection of sensitive and proprietary information including encryption, authentication, and audit trails. Data collection methodologies and controls are implemented and practiced to ensure privacy and conformance to assigned processing requirements and procedures. Data management functions are integrated with the system to ensure the responsibility for data management, security, and confidentiality is covered by appropriate management support and review.

8.2. Evolving Regulatory Landscape

A key trend influencing the data management landscape is the growing role of regulation. As concern with cybersecurity rises, new regulations and guidelines are being released by the European Union and governments worldwide. Within the EU, the General Data Protection Regulation is designed to protect EU citizens' personal data, while the Network and Information Systems Directive focuses on the security of network information systems, including telecommunications providers and crucial infrastructure. Governments around the world—such as in Australia, Canada, China, Japan, South Korea, and the United States—are introducing regulations to address cybersecurity risks and promote best practices in the financial sector.

Alliances such as the Financial Services Information Sharing and Analysis Center provide guidelines for information sharing among banks and security assessments. Moreover, many countries enact laws governing the use and protection of personal data. For example, China's Personal Information Protection Law outlines principles and regulations for handling personal information. The United States has enacted sector-specific state laws such as the California Consumer Privacy Act, while the European Union's Directive on Privacy and Electronic Communications protects individuals' privacy and personal data regarding electronic communications. These regulations collectively aim to safeguard the privacy, security, and integrity of personal information in an increasingly connected world.

9. Conclusion

Throughout this work, we have strategized the deployment of Vault - a robust secrets management system - within the dynamically evolving and distributed Loki platform. Technological transitions influenced these strategies, especially the move from a monolithic logistics data backend to a decoupled, distributed system. Vault was initially employed to manage integration and authentication secrets of the centralized backend. The migration distributed logistics data components across varied cloud providers and remote data centers, reinforcing the need for a centralized, secure repository for newly generated API keys and access credentials. The logistical platforms continue to leverage Vault for encryption and decryption of Rustlet secrets and client authorization to system components.

Securing containerized workloads with VPNs introduced additional requirements for managing OpenVPN credentials and keys in a centralized store. Loki components rely on Vault during startup to retrieve OpenVPN credentials and keys for establishing connections. Expanded authorizations for business services in Loki's ecosystem created a need for a centralized authorization platform. Consequently, integration and authentication secrets for new business services are stored in Vault. This guarantees secure and authorized communication among these services.

Across all phases of this architectural evolution, administrative roles in Vault assume critical responsibility. Efficient management of Vault entails enabling its audit logs to record every HTTP call, fostering a comprehensive trail of operational and security events.

References:

[1] Bauskar S. A review on database security challenges in cloud computing environment. Available at SSRN 4988780. 2024 Sep 5.

[2] Rahul S, Kumaran U, Sai TT, Pramodh T, Balasubramanian S. Preventing SQL Injection Attacks on Web Applications for Enhanced Security and CIA Triad Compliance. InInternational Conference on Advances in Information Communication Technology & Computing 2024 Apr 29 (pp. 99-110). Singapore: Springer Nature Singapore.

[3] Thilakraj M, Anupriya S, Cibi MM, Divya A. Detection of SQL injection attacks. In2024 International Conference on Inventive Computation Technologies (ICICT) 2024 Apr 24 (pp. 1515-1520). IEEE.

[4] Panda SP, Muppala M, Koneti SB. The Contribution of AI in Climate Modeling and Sustainable Decision-Making. Available at SSRN 5283619. 2025 Jun 1.

[5] Shivadekar S. Artificial Intelligence for Cognitive Systems: Deep Learning, Neuro-symbolic Integration, and Human-Centric Intelligence. Deep Science Publishing; 2025 Jun 30.

[6] Davidson L. Pro SQL Server Relational Database Design and Implementation: Best Practices for Scalability and Performance. Apress; 2021.

[7] Zhang W, Li Y, Li X, Shao M, Mi Y, Zhang H, Zhi G. Deep Neural Network-Based SQL Injection Detection Method. Security and Communication Networks. 2022;2022(1):4836289.

[8] Roy P, Kumar R, Rani P. SQL injection attack detection by machine learning classifier. In2022 International conference on applied artificial intelligence and computing (ICAAIC) 2022 May 9 (pp. 394-400). IEEE.

[9] Katsogiannis-Meimarakis G, Koutrika G. A survey on deep learning approaches for text-to-SQL. The VLDB Journal. 2023 Jul;32(4):905-36.

[10] Fotache M, Munteanu A, Strîmbei C, Hrubaru I. Framework for the assessment of data masking performance penalties in SQL database servers. Case Study: Oracle. IEEE Access. 2023 Feb 22;11:18520-41.

[11] Panda SP. Augmented and Virtual Reality in Intelligent Systems. Available at SSRN. 2021 Apr 16.

[12] Karwin B. SQL Antipatterns, Volume 1: Avoiding the Pitfalls of Database Programming. The Pragmatic Programmers LLC; 2022 Oct 24.

[13] Nasereddin M, ALKhamaiseh A, Qasaimeh M, Al-Qassas R. A systematic review of detection and prevention techniques of SQL injection attacks. Information Security Journal: A Global Perspective. 2023 Jul 4;32(4):252-65.

[14] Chakraborty S, Aithal PS. CRUD Operation on WordPress Database Using C# SQL Client. International Journal of Case Studies in Business, IT, and Education (IJCSBE). 2023 Nov 28;7(4):138-49.

[15] Panda SP. The Evolution and Defense Against Social Engineering and Phishing Attacks. International Journal of Science and Research (IJSR). 2025 Jan 1.

[16] Choi H, Lee S, Jeong D. Forensic recovery of SQL server database: Practical approach. IEEE Access. 2021 Jan 18;9:14564-75.

[17] Thalji N, Raza A, Islam MS, Samee NA, Jamjoom MM. Ae-net: Novel autoencoder-based deep features for sql injection attack detection. IEEE access. 2023 Nov 28;11:135507-16.

[18] Chakraborty S, Paul S, Hasan KA. Performance comparison for data retrieval from nosql and sql databases: a case study for covid-19 genome sequence dataset. In2021 2nd International Conference on Robotics, electrical and signal processing techniques (ICREST) 2021 Jan 5 (pp. 324-328). IEEE.

[19] Crespo-Martínez IS, Campazas-Vega A, Guerrero-Higueras ÁM, Riego-DelCastillo V, Álvarez-Aparicio C, Fernández-Llamas C. SQL injection attack detection in network flow data. Computers & Security. 2023 Apr 1;127:103093.

[20] Antas J, Rocha Silva R, Bernardino J. Assessment of SQL and NoSQL systems to store and mine COVID-19 data. Computers. 2022 Feb 21;11(2):29.

[21] Shivadekar S, Kataria DB, Hundekar S, Wanjale K, Balpande VP, Suryawanshi R. Deep learning based image classification of lungs radiography for detecting covid-19 using a

deep cnn and resnet 50. International Journal of Intelligent Systems and Applications in Engineering. 2023;11:241-50.

[22] Ashlam AA, Badii A, Stahl F. Multi-phase algorithmic framework to prevent SQL injection attacks using improved machine learning and deep learning to enhance database security in real-time. In2022 15th International Conference on Security of Information and Networks (SIN) 2022 Nov 11 (pp. 01-04). IEEE.

[23] Tanimura C. SQL for Data Analysis: Advanced Techniques for Transforming Data Into Insights. " O'Reilly Media, Inc."; 2021 Sep 9.

[24] Brunner U, Stockinger K. Valuenet: A natural language-to-sql system that learns from database information. In2021 IEEE 37th International Conference on Data Engineering (ICDE) 2021 Apr 19 (pp. 2177-2182). IEEE.

[25] Panda SP. Relational, NoSQL, and Artificial Intelligence-Integrated Database Architectures: Foundations, Cloud Platforms, and Regulatory-Compliant Systems. Deep Science Publishing; 2025 Jun 22.

[26] Khan W, Kumar T, Zhang C, Raj K, Roy AM, Luo B. SQL and NoSQL database software architecture performance analysis and assessments—a systematic literature review. Big Data and Cognitive Computing. 2023 May 12;7(2):97.

[27] Hong Z, Yuan Z, Zhang Q, Chen H, Dong J, Huang F, Huang X. Next-generation database interfaces: A survey of llm-based text-to-sql. arXiv preprint arXiv:2406.08426. 2024 Jun 12.

[28] Islam S. Future trends in SQL databases and big data analytics: Impact of machine learning and artificial intelligence. Available at SSRN 5064781. 2024 Aug 6.

[29] de Oliveira VF, Pessoa MA, Junqueira F, Miyagi PE. SQL and NoSQL Databases in the Context of Industry 4.0. Machines. 2021 Dec 27;10(1):20.

[30] Rockoff L. The language of SQL. Addison-Wesley Professional; 2021 Nov 4.

[31] Shivadekar S, Halem M, Yeah Y, Vibhute S. Edge AI cosmos blockchain distributed network for precise ablh detection. Multimedia tools and applications. 2024 Aug;83(27):69083-109.

[32] Lawson JG, Street DA. Detecting dirty data using SQL: Rigorous house insurance case. Journal of Accounting Education. 2021 Jun 1;55:100714.

[33] Zhang B, Ren R, Liu J, Jiang M, Ren J, Li J. SQLPsdem: A proxy-based mechanism towards detecting, locating and preventing second-order SQL injections. IEEE Transactions on Software Engineering. 2024 May 14;50(7):1807-26.

[34] Panda SP. Artificial Intelligence Across Borders: Transforming Industries Through Intelligent Innovation. Deep Science Publishing; 2025 Jun 6.

[35] Gandhi N, Patel J, Sisodiya R, Doshi N, Mishra S. A CNN-BiLSTM based approach for detection of SQL injection attacks. In2021 International conference on computational intelligence and knowledge economy (ICCIKE) 2021 Mar 17 (pp. 378-383). IEEE.

[36] Dhanaraj RK, Ramakrishnan V, Poongodi M, Krishnasamy L, Hamdi M, Kotecha K, Vijayakumar V. Random forest bagging and x-means clustered antipattern detection from SQL query log for accessing secure mobile data. Wireless communications and mobile computing. 2021;2021(1):2730246.