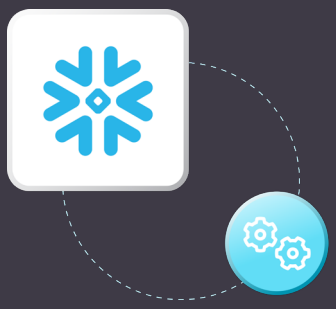
ConductorOne

Best Practices for Configuring Snowflake Access Control



Cloud administration carries the weighty responsibility of safeguarding a company's cloud infrastructure. This can be especially challenging when user data is involved, and compliance with stringent data protection regulations (such as HIPAA, GDPR, PCI DSS, and ISO 27001) is mandatory due to the serious consequences of breaches and data leaks.

In collaborative environments like Snowflake, you need to find the right balance between accessibility and security. Snowflake addresses this by embedding best practices like the principle of least privilege into its architecture. This ensures that each user has access only to what they absolutely need.

This article explores how you can use Snowflake's security framework to effectively minimize your attack surface and mitigate data loss risks.

Understanding Snowflake Access Control

Snowflake's access control system is built around four key concepts that ensure data security and controlled access:

- **Securable objects:** These are entities like tables and schemas that you can grant or deny access to.
- **Roles:** These are groups to which you can assign privileges that define specific actions on objects. Roles can be assigned to users or other roles, creating a hierarchical structure.
- **Privileges:** Distinct from roles and securable objects, privileges dictate the granularity of access control by specifying actions a role can perform on a securable object—for example, reading, writing, or modifying them.
- **Users:** Associated with roles, users are recognized entities in Snowflake that let you ensure that only authorized individuals or applications can interact with the data.

This framework of roles, privileges, and securable objects provides a granular and flexible approach to access control in Snowflake.

For instance, consider a scenario where you have two user groups: data analysts and data scientists. You can create a role for each group, granting the data analysts SELECT privileges on sales data tables and the data scientists SELECT, INSERT, and UPDATE role privileges on the same tables as well as access to more sensitive financial models.

As mentioned, roles are the foundation of access control and management. This is true for both Snowflake and any cloud providers it integrates with. However, Snowflake has a distinct feature called role hierarchy and privilege inheritance that allows you to create a hierarchical structure for organizing roles. Simply put, roles can inherit privileges from other roles in a parent-child fashion. For example, a company's "MarketingAdmin" role may have permissions to create, update, and delete campaign data, while the "MarketingAnalyst" role, which typically has only read access, could inherit the update privilege from the "MarketingAdmin" role. This setup allows for flexible and secure data management and ensures analysts have the necessary access without excess privileges.

Snowflake Roles

Snowflake provides a set of built-in, system-defined roles. These are preconfigured with specific privileges to perform common tasks within the platform and are designed to simplify access control management so that users have an appropriate level of access according to their responsibilities.

The following are Snowflake's system-defined roles:

- **ORGADMIN (organization administrator):** Manages operations at the organization level, including creating accounts, viewing all accounts and regions, and accessing usage information across the organization.
- **ACCOUNTADMIN (account administrator):** The top-level role in an individual account, encapsulating the SYSADMIN and SECURITYADMIN roles. It has full administrative permissions, including user and role management and access to all objects.
- **SECURITYADMIN (security administrator):** Manages global object grants, user and role creation, monitoring, and management. Inherits USERADMIN privileges and has the MANAGE GRANTS security privilege.
- **USERADMIN (user and role administrator):** Dedicated to user and role management only, with privileges to create and manage the users and roles that it owns.
- **SYSADMIN (system administrator):** Has privileges to create warehouses, databases, and other objects. This role should be at the top of the custom role hierarchy so it can grant other roles privileges related to specific objects.
- **PUBLIC:** A pseudo-role automatically granted to every user and role, typically used when explicit access control is not needed. Objects owned by PUBLIC are accessible to all users and roles in the account.

User-Defined Roles vs. Object-Level Privileges

Aside from system-defined roles, Snowflake also enables you to create your own set of user-defined roles, which allow for tailored access control that aligns with your organization's specific needs. Unlike object-level privileges, which define specific actions that can be performed on individual database objects (like tables or views), user-defined roles determine who can execute those actions. That is, object-level privileges specify the "what" and roles determine the "who."

You can leverage user-defined roles to assign a bundle of object-level privileges to specific users or groups. This streamlines permission management across the organization and ensures that access rights are consistently applied according to job functions or departmental requirements.

User-defined roles are tailored to meet specific organizational needs by grouping sets of privileges. For instance, a custom "DataAnalyst" role could be configured to inherit privileges from the "SYSADMIN" role. This doesn't mean it simply duplicates the "SYSADMIN" role; rather, it selectively inherits certain privileges that are appropriate for data analysis tasks, such as accessing and querying databases.

Additionally, this role can be further customized to include specific object-level privileges that are not inherently part of the "SYSADMIN" role. For example, while the "DataAnalyst" role inherits general access privileges from "SYSADMIN," it would need explicit object-level permissions like SELECT on specific tables to perform data analysis. This ensures that the "DataAnalyst" role has the necessary tools to operate within their scope without accessing broader system administrative functions, reinforcing security and adherence to the principle of least privilege.

Best Practices for Configuring Snowflake Access Control

Access control in Snowflake can ensure that only authorized users have the necessary permissions. However, it will only be effective if it's configured correctly. The following sections explore some best practices for configuring Snowflake access control and ensuring that all your data is only accessible by authorized users.

Role Creation Principles, Hierarchy, Maintenance, and Review

You need to think carefully about the roles that you create. You should follow the principle of least privilege and ensure that users are granted only the minimum number of privileges they need for their tasks. In addition, users with similar access requirements should be grouped into dedicated roles to simplify management and allow for consistent access across users with the same needs.

Snowflake's role inheritance allows for streamlined management of access rights by creating a clear hierarchical structure. For example, an "employee" role might inherit specific read privileges from a "manager" role, enabling them to view but not alter data in certain areas where the manager has broader edit and delete capabilities. This delineation ensures that each role accesses only the data necessary for their tasks, adhering to the principle of least privilege.

You need to keep the role hierarchy straightforward to avoid complex inheritance, which can lead to confusion and management difficulties. Rather than creating deeply nested levels of roles, maintain a simple structure where roles inherit common privileges that are easy to track and modify. This approach not only simplifies understanding and management of role-based access but also reduces administrative overhead by making it easier to update roles as organizational needs change. Instead of adjusting individual user permissions, changes can be made at the role level that efficiently propagate to all users assigned to that role.

As a company changes, so do its access control requirements. For instance, computing assets escalate as a company grows, and roles and access control need to account for those changes accordingly. So, like any other cybersecurity configuration, roles should be regularly reviewed and updated. This involves revoking unused privileges and identifying instances of overprivilege. Additionally, you should maintain comprehensive and updated documentation on roles and their associated privileges for future reference and auditability. This documentation should include details about the privileges granted to each role and any changes made over time, providing a clear record of access control measures within the Snowflake environment.

Secure Authentication and Access

Roles and privileges come with responsibility, and the consequences of a breached account with a high level of permissions can be severe. For example, if you handle personally identifiable information (PII), you may be required to report the breach to authorities, which can result in hefty fines, irreparable damage to your reputation, and a loss of trust from your users.

Your system administrators can use various tools and practices to enhance user experience while keeping access as secure as possible:

- → **Multifactor authentication (MFA):** This security measure requires users to provide two or more verification factors to gain access to a resource. All users, not just admins, should be MFA-enabled to reduce the risk of unauthorized access.
- → **Single sign-on (SSO):** This system allows users to access multiple applications with one set of login credentials, enhancing security while making access more convenient and efficient—even for less technical users.
- → **Strong passwords and regular rotation:** While tools like MFA and SSO can reduce reliance on passwords, they are only the first line of defense. Passwords should be strong, incorporating a mix of special characters, numbers, and capital letters, and should be rotated periodically to maintain security.
- → **IP address restrictions:** For the most sensitive access points, IP whitelisting or network restrictions can be employed to allow access only from trusted locations or networks.

Properly securing authentication and authorization is key when ensuring that only legitimate users access an organization's resources. Aside from enforcing a login, further access policies are needed to make sure the data is protected. While a legitimate user might rightfully log in to the organization, they should only have access to the resources they need and nothing more.

Object-Level Privileges and Access Control Lists

In addition to securing access, in Snowflake, it's also essential to grant object-level privileges based on specific user needs and responsibilities. This ensures that each user or role has only the necessary access to perform their tasks. Avoid granting excessive permissions to prevent security vulnerabilities. Leverage access control lists (ACLs) for granular control over individual object access, and regularly review and update ACLs to reflect changes in ownership and access requirements. This approach maintains security and compliance by ensuring that data is accessed *only* by authorized users. This approach also prevents insider threats, where a disgruntled user might attempt to disrupt company operations or steal massive amounts of data.

Monitoring and Auditing Access Controls

Effective access control in Snowflake requires robust logging and audit trails for all access activities. You need to be able to monitor user activity and identify potential identity security threats in real time. Audit logs are invaluable for forensic analysis and incident response, as they provide detailed records of who accessed what and when. Additionally, automated tools for reviewing and reporting significantly enhance efficiency and enable proactive access control management. These tools can help ensure that any anomalies or security breaches are quickly detected and addressed.

A Quick Recap in Managing Snowflake Accounts

Managing a Snowflake account effectively involves adhering to security best practices that not only safeguard account credentials but also enhance the overall security posture:

- → **Implement strong password policies:** Ensure strong, complex passwords and enforce regular rotation.
- → **Enable multi factor authentication (MFA):** All users, especially account administrators, should use MFA for an added layer of security.
- → **Limit and monitor the ACCOUNTADMIN Role:** Due to its extensive privileges, access to this role should be closely controlled and monitored.
- → **Conduct regular reviews and updates:** Continuously align account permissions with current needs and security policies.
- → **Leverage integration with security providers:** Utilize Snowflake's capabilities with security solutions like SIEM systems for improved monitoring and incident response.

These measures will not render your data lake impervious to threats but will significantly increase the difficulty for potential attackers. Implementing these practices will deter cybercriminals, making it more likely they will look for less secure targets.

Conclusion

This article explored the intricacies of access control in Snowflake, covering the importance of roles and privileges, role inheritance, secure authentication, object-level privileges, monitoring, auditing, and best practices for account management. A well-defined, crafted, and managed access control system is vital for maintaining the security and integrity of your data, reducing the risk of unauthorized access and ensuring compliance with regulations.

If you want to enhance the efficiency and effectiveness of these security measures, ConductorOne provides a seamless access control management solution that integrates with Snowflake. By offering automated access reviews, provisioning, and deprovisioning, ConductorOne amplifies the benefits of robust role management and secure practices discussed in this article. Leveraging these tools and principles enables you and your team to build a robust access control system that not only protects your data but also empowers your organization to securely harness its full potential, thus aligning with Snowflake's capability to securely scale and manage complex data environments.

Want to learn more about our identity security platform for modern workforces?

Get a demo

ConductorOne | team@conductorone.com

AICPA SOC