snowflake®

SNOWFLAKE PATTERN
Authentication Scenarios

Choosing the appropriate authentication method pattern with security as a focus

Version 1.0

Latest Revision: Oct 2020

OVERVIEW

Pattern Status	ACTIVE
Pattern Superseded by	NONE

Snowflake supports authentication methods that cover a number of scenarios, ranging from human interactive scenarios, to programmatic service-account use cases. Client applications that connect to data sources like Snowflake typically have their own specifically supported authentication methods that vary from application to application. Consider BI tools, for example: Some support SAML 2.0 for single sign-on, while others don't.

This document spotlights authentication patterns that support the following scenarios:

1. **Interactive, SSO authentication for humans**.
2. **Non-interactive authentication for non-human users, such as programmatic accounts and service accounts**.

Security Patterns

This pattern is part of a series of Architecture Patterns covering Security. The patterns are:

- Access to Sensitive Objects
- Authentication
- Network Architecture

INTENDED AUDIENCE

This document is for Enterprise and Solution Architects who want to understand the connectivity capabilities and best practices of Snowflake and Snowflake Partner technologies. This document *is not* intended for use by implementation teams, although an implementation example may be provided.

WHEN TO USE THESE PATTERNS

The patterns in this document satisfy one or more of the following requirements:

A. The organization has both cloud and on-prem tools that need to authenticate to Snowflake.

B. The organization uses an IdP to manage authentication, which eliminates the need for users to maintain multiple passwords for different systems.

C. The organization has service accounts that require an authentication method that's more secure than username and password.

D. Legal or contractual agreements require the organization to implement specific authentication methods.

What You'll Learn

You will learn how to apply two techniques for authentication to Snowflake:

- Federated Authentication
- Key pair Authentication

PATTERN DETAILS

Across the three authentication patterns, there are five ways to authenticate to Snowflake:

1. **Built-in username/password authentication** in which the password is stored in the Snowflake USER object and the user authenticates with Snowflake. The USER object is delivered as a string, or the password is typed in by the user. This option is not as secure as alternative options.
2. **Built-in username/password authentication with multi-factor authentication (MFA).** This is option one with the addition of multi-factor authentication for security. Note that this option only supports Duo MFA.
3. **SSO powered by SAMLv2** for human, interactive use cases.
4. **Key Pair authentication** for non-human users, such as programmatic access or service accounts.
5. **OAuth 2.0 code grant flow** for secure, programmatic access to Snowflake data.

Snowflake does not recommend basic, built-in username/password authentication (option 1) because the alternatives offer better security. In situations where the only method to define a connection to an application is the username and password on the connection screen, Snowflake recommends option 2, multi-factor authentication implemented through Duo.

There are human, interactive use cases where federated authentication is the best supported method. Any SAML 2 compatible IdP can achieve this. Some partner applications also deliver federated SSO experiences leveraging OAuth 2.0, however, client application support for federated authentication varies. SAML 2 is an option in some cases through Snowflake's "External Browser" mode on the desktop. When a desktop application is configured to use External Browser mode, a Snowflake-provided driver opens a new browser tab/window so the user can authenticate with their IdP credentials.

Programmatic (non-human) use cases can use built-in service account passwords to authenticate, but only use this method as a last resort. Instead, consider key pair authentication combined with a secrets management solution where the client uses its private key, and Snowflake uses public keys to decrypt it and authenticate. A third option is External OAuth, which is the only method that allows for an SSO-based user credential in the programmatic scenario.

PATTERN EXAMPLE 1

Now let's look at two applications, Tableau and Microsoft Power BI, that support OAuth. The type of OAuth supported differs and in each case the client application determines which technology you should use. We will start with Tableau.

When connecting to Snowflake, Tableau Desktop/Server/Online supports an SSO-like experience through Snowflake OAuth. This example is appropriate when customers want to provide an SSO-like experience to their Tableau user population when accessing Snowflake data. This method prompts user intervention to grant Tableau access to Snowflake. An external authorization server is not required in this scenario. Snowflake acts as both the Authorization Server and the resource server. Customers that manage identity centrally and wish to provide a more secure method of accessing Snowflake other than username and password will want to use this pattern.

When connecting to Snowflake, Microsoft Power BI supports an SSO-like experience through External OAuth. This scenario is appropriate when customers want to allow Microsoft Power BI users to connect to Snowflake using Identity Provider credentials and an OAuth 2.0 implementation. Customers should take into account that this approach requires Azure AD, because that is what issues the token for access to Snowflake on behalf of the user. This approach is appropriate to consider if a customer has an IdP other than Azure AD. For example, if the customer IdP is Okta or Active Directory Federation Services (ADFS), Azure AD takes the user through the Security Assertion Markup Language (SAML) authentication process with the IdP before logging the user into the Power BI service. This scenario is appropriate when a customer wants to manage identity centrally and provide a more secure method of accessing Snowflake other than username and password.

SAML SSO may not be appropriate in cases that involve a customer's Snowflake administrators. Outages with an IdP may prevent Snowflake administrators whose passwords are stored in the IdP from logging in to Snowflake. In this case SAML SSO is not recommended and customers opt to maintain an administrator with a Snowflake password to manage federated authentication and troubleshoot any issues that occur.

SAML SSO is also not appropriate if the client applications do not support that method. Consider evaluating if the application supports an SSO-like experience similar to Power BI and Tableau.

Some customers use Cloud Service Provider (CSP) private networking technologies such as AWS PrivateLink and Azure Private Link with the use of SAML SSO. This is an appropriate scenario but

requires a choice. Currently customers can only configure single sign-on to either work with their regular, non-PrivateLink URL, or with the PrivateLink URL.

GUIDANCE

INCOMPATIBILITIES

1. At the time of this writing, SAML-based SSO can only be used on either public or private Snowflake endpoints at one time. This will be addressed in future releases.
2. At the time of this writing, Snowflake only supports a single IdP at a time for each Snowflake Account for SSO.
3. For SSO the web UI only supports SAML 2.

OTHER IMPLICATIONS

Not applicable at the time of this writing.

DESIGN PRINCIPLES & BENEFITS ENABLED BY THIS PATTERN

The benefit of configuring Snowflake for federated authentication is users need to log in just once with one set of credentials to get access to all corporate apps, websites, and data for which they have permission. This benefits users in the form of simplicity as they do not have to manage multiple passwords for access to SaaS applications, data, or websites required to perform their job duties.

The unification of user access management means there is a central directory to provision and deprovision users. Configuring SSO for Snowflake with any SAML 2.0 compliant IdP helps customers think of Snowflake as many other SaaS applications that use this common protocol.

SSO for human interactive use cases must consider the capabilities supported by Snowflake in concert with the authentication capabilities of the systems users need to authenticate to Snowflake. This matrix needs to be considered along with the three scenarios described to enable SSO for as many systems as possible.

PATTERN EXAMPLE 2

This pattern example compares when you should use key pair authentication for non-human users versus when you should use external OAuth for secure, programmatic access to Snowflake data. This example is relevant to programmatic or service account requirements that require access to Snowflake. An evaluation of supported authentication methods of service accounts and programmatic access requirements helps to determine which non-human user authentication methods to use with Snowflake.

Key pair for authentication of service accounts to Snowflake is an appropriate example when customers have requirements to not rely on a third party or for the secret to travel over the wire

as part of authenticating service accounts. The private key can be managed internally by a customer without relying on cloud-based IdPs, such as Azure AD or Okta. Secrets not traveling over the wire is a great benefit of key pair authentication where the private key stays with the customer and the public key lives in Snowflake.

An additional example appropriate for key pair authentication is if the customer wants to remove the management of the secret from the service account authenticating to Snowflake. With key pair authentication, the private key does not need to be in the possession of the user. The key is managed by code and therefore the service account itself is not in possession of the key.

This example allows for the aggressive rotation of keys without disrupting connectivity. Since Snowflake allows for two active public key values at any time, consider this as part of the pattern. This example is best used with a secrets management platform like Hashicorp Vault, AWS Secrets Manager, or Azure Key Vault to manage the private key.

Key pair authentication is not appropriate in scenarios where existing key infrastructure is not in place to provide for the protection of private keys. This method may not be appropriate in large environments where the ability to distribute and manage keys becomes more administrative overhead than what the customer is willing to deal with.

External OAuth 2.0 is a supported method for non-human users to access Snowflake. Customers that seek to allow for SSO-based user credentials in the programmatic scenario should consider this option. OAuth 2.0 is appropriate for customers that want to centralize the management of tokens issued to Snowflake by service accounts to ensure that programmatic access or access by the service account to Snowflake data has to go through the External OAuth configured service. Customers with this requirement may have additional requirements to centralize the monitoring of authorizations across a number of applications. Customers that do not wish to pass credentials over the wire or store secrets in Snowflake will find this method useful.

More specific examples where OAuth is appropriate include embedding Snowflake into your application where the application requires access to Snowflake, or if SAML cannot be accomplished because there is a program that requires access to Snowflake then OAuth is an appropriate pattern.

Customers with centralized monitoring requirements should consider OAuth. With OAuth, customers can see delegated access to Snowflake and other applications such as Salesforce in one place. This differs from key pair in that, if I want to audit users authenticating through key pair, that answer lives in Snowflake. This model also supports examples where customers want to deprovision service identities from a centralized place.

GUIDANCE

INCOMPATIBILITIES

1. Snowflake OAuth is not applicable in the programmatic scenario. External OAuth should be used.

OTHER IMPLICATIONS

Not applicable at the time of this writing.

DESIGN PRINCIPLES & BENEFITS ENABLED BY THIS PATTERN

The benefits of configuring Snowflake for Key Pair Authentication include:

1. The secret does not travel over the network
2. The user does not need the private key.
3. Snowflake allows for the aggressive rotation of key pairs.
   The outcome of key pair authentication is that it is more secure than username and password.

The result of External OAuth authentication is centralized management of tokens issued to Snowflake, and service accounts or users used exclusively for programmatic access will only ever be able to use Snowflake data when going through the External OAuth configured service. Customers benefit from sessions initiated with Snowflake do not require a password and only initiate their sessions through external OAuth.

RELATED RESOURCES

The following related information is available.

Snowflake Related Patterns	NA
Snowflake Community Posts	Using SSO between Power BI and Snowflake Using OAuth 2.0 with Snowflake Snowflake Service Account Security Part 1 Snowflake Service Account Security Part 2
Snowflake Documentation	Snowflake Password Policy Federated Authentication & SSO Using Key Pair Authentication Snowflake OAuth External OAuth Summary of Security Features
Partner Documentation	Configure SSO - Azure AD and Snowflake Configure SSO - Okta and Snowflake