Enhancing Data Security

A Deep Dive into **API Access Control** in Salesforce

CloudKettle

In today's interconnected business landscape, robust API (Application Programming Interfaces) access control is crucial for safeguarding sensitive data within Salesforce. As a Salesforce Admin, mastering the intricacies of API access control is pivotal to maintaining a secure and efficient Salesforce environment. In this resource, we'll explore the critical aspects of API access control and how it contributes to airtight data security.

Understanding API Access Control:

API access control in Salesforce involves managing who can access your organization's data through APIs. Salesforce provides comprehensive tools and features to help administrators effectively regulate and secure these access points.

1. **Profile and Permission Sets:**
   Leverage Salesforce profiles and permission sets to define which users can access APIs. Assign appropriate API-enabled permissions to profiles and permission sets, ensuring only authorized users have API access.
2. **OAuth and Connected Apps:**
   Implement OAuth authentication to control access to your Salesforce data via external applications. Configure connected apps with granular permissions, specifying which data and operations external apps can access.
3. **IP Allowlisting:**
   Strengthen API access control by setting up IP allowlisting. Restrict access to specific IP addresses, minimizing the risk of unauthorized API access from unsecured locations.
4. **Session Policies:**
   Fine-tune session policies to regulate API session behavior. Define session timeout settings to automatically terminate idle API sessions, reducing the risk of unauthorized access.

Advantages of API access control:

1. **Granular Control:**
   Salesforce provides granular control over API access, allowing administrators to define who can access which data and what actions they can perform.
2. **Enhanced Security:**
   By controlling API access, organizations can prevent unauthorized access to sensitive data, reducing the risk of data breaches.
3. **Compliance:**
   API access control helps organizations comply with regulatory requirements such as GDPR, HIPAA, etc., by ensuring that only authorized users can access and manipulate data.
4. **Integration Flexibility:**
   Salesforce APIs enable seamless integration with external systems, and access control ensures that integrations are secure and comply with organizational policies.
5. **API Usage:**
   API usage can be tracked through a report to help identify potential security threats or data misuse.
   - Admins can view a more detailed report of the API Calls made within the last 7 days. This report can be generated by switching to Salesforce Classic, selecting the reports tab, and then selecting the "API Usage Last 7 days" report within the Administrative folder. This report is broken down into the Name, Username, Email, Client ID (App), the Day of the Week, and the Call Count.
   - As a developer, you can query the Event Log for the USER_ID, CLIENT_ID, and CLIENT_NAME, as well as any other fields, to help identify API usage.

For more information on this, see our resource on Salesforce Storage Limits.

Disadvantages of API access control:

1. **Operational Complexity:**
   Managing and maintaining IP allowlists can be operationally complex, especially in organizations with dynamic IP addresses or distributed teams accessing Salesforce from various locations. The same can be said for managing the connected apps and assigning them to the right users in their profiles or creating permission sets for those connected apps.
2. **Overhead for Remote Users:**
   Remote users or those accessing Salesforce from dynamic IP addresses may encounter challenges if their IP addresses are not allowlisted, requiring additional steps to gain access.
3. **Performance Impact:**
   Overly restrictive API access controls may impact system performance, especially for integrations requiring frequent data access, necessitating a balance between security and performance. IP-based restrictions may limit flexibility for users who need to access Salesforce from different locations or devices, potentially impacting productivity if not managed properly.
4. **Potential for Misconfigurations:**
   Misconfigurations can inadvertently expose sensitive data or restrict legitimate access, leading to operational disruptions or security incidents. Misconfigurations in IP allowlists can block legitimate users or systems from accessing Salesforce, leading to operational disruptions or user frustration.

API Access Control Feature

Salesforce has a feature called “API Access Control,” which can be requested from Salesforce Customer Support. In Salesforce, API Access Control empowers administrators to tighten security by controlling access to APIs for connected apps. By implementing this feature, you can ensure that only approved connected apps can interact with Salesforce data, safeguarding your organization's sensitive information.

Here's how the Salesforce API Access Control feature works:

1. **Lock Down Access** - Initially, you lock down access to Salesforce APIs for all connected apps. This creates a secure baseline, preventing any unauthorized access attempts.
2. **Allowlist Approved Apps** - Next, you can selectively approve (allowlist) specific connected apps that meet your organization's security standards and business needs. These approved apps are granted access to Salesforce APIs, ensuring only trusted applications can interact with your data.
3. **Grant Access to Users** - Utilizing profiles and permission sets, you can grant access to the approved connected apps for specific users. This granular control allows you to tailor access permissions based on user roles and responsibilities within your organization.
4. **Secure API Access** - Once access is granted to users through the approved connected app, they can securely access Salesforce APIs, enabling them to perform authorized actions and tasks as needed.

Salesforce automatically creates connected apps for common applications and installs them in your org. These connected apps facilitate integration and collaboration with external services and platforms and would also be locked down. A connected app generates a token, and assigning the permission of the connected app to the user will open up access.

So before even switching the feature on, steps must be taken to ensure a successful outcome.

1. Salesforce administrators need to assess every connected app within their organization to ensure it meets security standards and serves a purpose for users. Any apps deemed unnecessary or insecure should be blocked. To do this, admins can follow these steps:
   - Go to Setup by clicking the gear icon in the top-right corner.
   - In the Quick Find box, type "Connected Apps OAuth Usage."
   - This will display a list of connected apps. In the Actions column, click the button to block the app.
2. Salesforce administrators need to assess every unblocked app to evaluate the users. In a large organization, you may need to be very specific by evaluating based on roles and not just profiles.
3. Once you have a list of which users should have access to which apps, the Admin can assign access through profiles or permissions sets.
4. Lastly, the Admin would be assigned the 'Use Any API Client' permission to open API Access for them or other users needing access to applications whose session ID does not originate from a connected app.

SOAP and REST API integrations

SOAP and REST API integrations are also impacted, as all API calls into Salesforce will be blocked if API Access Control Feature is enabled. When integrating Salesforce with external systems or applications using SOAP, developers must obtain an access token to authenticate their API requests. This token is proof of authorization and is required to access Salesforce data via the SOAP API. The authentication flow usually involves making a request to Salesforce's token endpoint, providing the necessary credentials (such as username, password, and security token) along with the SOAP request. If authentication is successful, Salesforce verifies the credentials and issues an access token. Once obtained, the access token is included in the SOAP request headers as a bearer token. This token is then used to authenticate subsequent API requests to Salesforce, allowing SOAP to access and manipulate Salesforce data on behalf of the user.

REST API access is more lightweight and flexible than SOAP API, making it well-suited for web and mobile application development and integrations with external systems. To access the REST API, developers must authenticate their requests using OAuth 2.0 or Session ID authentication. OAuth 2.0 is the preferred method for web and mobile applications, while Session ID authentication is commonly used for server-to-server integrations. Salesforce exposes a set of resource endpoints representing different objects and functionalities within the platform. Developers can interact with these endpoints to perform actions such as querying records, creating new records, updating existing records, and executing custom Apex REST endpoints. REST API requests are typically made using HTTP methods such as GET, POST, PUT, PATCH, and DELETE. Requests and responses are formatted in JSON or XML, providing a standardized and interoperable way to exchange data.

If API access becomes restricted within Salesforce, SOAP and REST APIs might face authentication challenges during access attempts. Even with the correct credentials provided, Salesforce could reject requests, potentially hindering SOAP and REST APIs from conducting vital testing and integration duties. Such occurrences could lead to setbacks or interruptions in development workflows. To mitigate these issues, administrators must adapt API access control settings to ensure uninterrupted authentication and access to the SOAP API.

Furthermore, specific endpoints or actions should remain unrestricted for the REST API to enable developers to execute particular operations or retrieve targeted data types. Resolving this may entail updating the CORS allowlist, adjusting connected app configurations, or modifying permission sets as required.

Something to consider is that the Chrome extension of Salesforce Inspector (a SOAP API) will be blocked due to this feature being enabled. Since administrators primarily rely on this extension, one approach could involve granting the 'Use Any API Client' permission. This permission allows for access to any application, encompassing all connected apps. However, caution should be exercised when assigning this permission set, as it effectively opens up unrestricted API access.

See the Salesforce help documentation Restrict Access to APIs with Connected Apps

Real-world Scenarios

Consider the scenario where a large company has many Salesforce users. Dataloader can be an accessible tool for all users to import, export, delete, and update data. However, with such wide access, it becomes crucial to have robust API access control to ensure that only authenticated users with appropriate permissions can manipulate the data, maintaining a secure and seamless user experience. Without proper API access control, a company could face a variety of problems.

- An unauthorized user might accidentally delete or alter important data, leading to data inconsistencies, reporting errors, and potential operational disruptions.
- Users may inadvertently (or intentionally) expose sensitive data to people who should not have access to it, leading to a serious data breach.
- An Enterprise company with thousands of users could have a user authorize many connected apps if the setting is ‘All users may self-authorize.’ That would open up the company data to these apps.

However, if the company used proper API access control, that would mitigate all of these potential issues - from accidental data deletion to unauthorized access.

Conclusion

API access control is not just a security feature; it's a strategic imperative for Salesforce administrators. By implementing robust controls and staying vigilant, organizations can harness the power of APIs while safeguarding their valuable data. Mastering API access control is pivotal to ensuring a resilient and compliant Salesforce environment in a landscape where data security is non-negotiable.