THE ROLE OF APIS IN DATA SECURITY AND PRIVACY

Content produced by: Eduardo Arantes, Gibson Nascimento, Fernanda Goulart and Felipe Mercadante

Introduction

The role of APIs in Data Security is part of a solid governance strategy, focusing on controlling traffic and data access. Nowadays, they are part of the infrastructure of the most competitive companies in the market, from the smallest, such as startups, to big corporations that have been in the market for decades.
Among some points that relate APIs to data, we have:

- **Traffic of multiple data points, identifiable or not;**
- **Establishment of standards for information exchange between systems;**
- **Easy and fast integration between systems.**

The Sensedia API Platform has a series of features that guarantee information confidentiality and security. They include properties from the access rules to API deployment environments to the application of security policies for each API (what we call interceptors). In this post, we describe some of the key tools.

Privacy by Design & Privacy by Default

Privacy by Design is an approach that ensures that privacy is thought since the design of the products and incorporated throughout the life cycle. It supports the vision of positive impacts of investments in privacy to protect the customer, as well as the business, seeking to place privacy on the same level of importance as all other key projects and product requirements

In **Privacy by Default**, the most secure privacy settings available are applied in all new products launched, and data is stored only as long as necessary. This implies that privacy must be linked to the business models and solution architecture of the integrations. Thus, when an API is launched, both information security and APIs access must be considered since the design stage.

Security layers for accessing APIs

The APIs deployed on the Sensedia API Platform are accessed by partner applications and it is important to have control over this process. We provide security layers to ensure that APIs will only be accessed by authorized applications.

TLS and mTLS

It is possible to deploy the APIs in open environments, without the need for certificates that validate the identity of the client and server. For greater security, however, the Sensedia Platform supports TLS and mTLS as security options for the environments created for the APIs implementation.

Transport Layer Security (TLS) is a security protocol designed for internet communications. The basis of this protocol's security is the verification of the server's identity, using a valid and digitally signed certificate, so that the client is sure that it is sending and receiving data to the correct recipient.

In the Sensedia Platform, the TLS option stipulates that the server identity is guaranteed, by sending a certificate from the gateway to the client, but the client does not need to send a certificate for authentication. In mTLS, the client (the application that will make requests to the API deployment environments) must also send a valid certificate when making these requests

OAuth 2.0

OAuth 2.0 is an authorization protocol that allows applications to access user data without having to enter their credentials. Applications use this method to gain access to a particular user through an ID or authentication keys. There are different OAuth 2.0 flows that require different ways of having an access token, and, on the platform, it can be configured according to the need.

There are two kinds of clients in OAuth. The confidential one can guarantee that no malicious user will obtain resources to impersonate a valid application. Non-confidential allows that these resources can be obtained, but validation happens via URL to ensure that the information only returns to the real application.

Access Tokens

An Access Token is a feature that allows you to include authorization data for an application to access an API. In the Sensedia Platform, an Access Token is always used in conjunction with an application and is only required if the API is configured to require validation.

In addition, it is possible to use the same Access Token for different applications through a plan management. In practice, Access Tokens are used to validate access, such as direct logins, where the user enters a login and a password, or through identity providers, such as Facebook or Google.

There are several types of tokens that can be used in different contexts, such as opaque access token or JWT (JSON Web Token). In addition to the types, there are specific configurations, such as expiration time, which can add greater security in this kind of access. For each context, a different kind of validation can be requested, being blocked by the server if it is non-standard.

In the Sensedia Platform, there are resources available for you to configure your APIs and implement custom authentication flows, inserting tokens such as MD5, UUID or random. In all of them, it is possible to customize parameters to meet specific demands.

APIs design

Security can and should be embedded in the APIs design, mainly through the application of policies by what we call interceptors. But controlling the steps for API design and implantation also represents security requirements.

Implantation environments

The APIs design goes through several stages of development until reaching a level of maturity enough for them to be exposed in a production environment. In the meantime, the use of isolated environments ensures that API management practices are applied to the different stages of development, with quality control through tests, ensuring more safety and efficiency.

Then, it is possible to create environments separated from the production environment with different security and control requirements, such as a more unrestricted sandbox environment and a specific testing environment. The security policies applied to APIs themselves can be checked before deployment in production.

Developers, teams, roles

It is also important to control the design process by defining the team responsible for each API and/or process.

The Sensedia API Platform provides the creation of developers' teams and the definition of roles for each developer or team. With this, it is possible to guarantee that only the responsible teams have access to APIs, environments and/or other functionalities of the Platform.

Implementation of policies via interceptors

You can instruct the API Gateway to execute snippets of code in an API request or response flow. This can be used to customize the behaviour of the API in different scenarios, applying security policies and business rules.

For example, it is possible to write a script that adds a header to the request depending on the tokens being used, or modify the response body that will be sent to the client in case of internal errors in the backend (to avoid exposing internal details).

There are several interceptors available on Sensedia Platform, divided into five categories: traffic control, tracking, security, data transformation and call mediation. In addition to the interceptors that are already available, users can create custom interceptors using Java or JavaScript.

Security interceptors are our focus here. They bring a series of features to make APIs more secure, protecting them against different types of attacks and validating information that is shared in the calls. They include:

- Access control on requests, with validation of access token, client ID, IPs, and digital signature.
- Preventing CSRF and injection attacks in JSON, SQL, XML and XSS.
- Concealment and/or encryption of payload data.

However, other types of interceptors can also be used in conjunction with security interceptors to build broader secure design solutions. For example, tracking interceptors allow controlling the tracked data, with the possibility of hiding sensitive data within the generated logs.

Ensuring data security through API Governance

Adaptive Governance is a Sensedia product that enables advanced governance features to control the API lifecycle. Among these features, there is the ability to create alternative workflows that meet the needs of different business areas in a company.

Looking at security, Adaptive Governance allows policies, such as mandatory oAuth, ClientID and Data Cryptography, to be defined for each step of the workflow, which helps to ensure that an API is not available in production (or any other environment) without mandatory policies configured.

In addition to the validation of mandatory policies, it is also possible to configure policies that should not be present when deploying an API in any environment. For example, when an API transits sensitive data, it is possible to configure the system such that a no additional Log policy is enabled for that API.

These validations ensure that APIs only have policies that are actually necessary and mandatory. They are executed automatically and ensure that the process of deploying an API is assertive, without the need of human intervention.
In addition to the Sensedia Adaptive Governance module, Sensedia's consulting team developed an API Governance playbook to support their customers in defining the governance strategy.

The playbook includes defining teams and responsibilities, governance models, policies, standards, security mechanisms, KPIs, impact analysis, prioritizing APIs and configuring workflows. These definitions aim to ensure digital strategies control and evolve with APIs.

About Sensedia

Founded in 2007, Sensedia is an Application Programming Interface (API) specialist with offices in the UK, Brazil, and Peru. Sensedia works with market leaders in varying sectors and its solutions enable clients to extend their digital businesses. Sensedia is one of the main API pure players in the world, focused only on its API Management Platform and Strategy & Professional Services around the full lifecycle of API Management. Sensedia is recognized by Gartner in its Magic Quadrant as Visionary and by Forrester in its Wave as Strong Performer.

Find out more at www.sensedia.com