International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 5(1), September - October 2015 ISSN 2278-6856

Authentication Playground for SaaS Connection

Prakash Hiremath , Dr P Jayarekha

Student M.tech 4th Sem, BMS College of Engineering. Department of Information Science, Bangalore 560019, India

Associate Professor, BMS College of Engineering. Department of Information Science, Bangalore 560019, India

Abstract

The Cloud has increased the demand for integrating between services. Companies want to connect from on-premises apps to cloud services and from cloud services to cloud services and these connections need to be secure and governed for performance. In a Cloudstreams environment developing of an Enterprise connector for any service provider involves understanding of the authentication method used in order to connect and make API calls to the Software as a Service (SaaS) provider. To develop multiple connectors, it is good to have a single place to generate authentication data for standard authentication mechanism like OAuth 1.0, OAuth 2.0 so that authentication data can be generated easily without much overhead. Calculating authentication data for customized authentication, involves extra steps which will be time consuming for end user and also for connector developer to follow the steps. For REST based connector, the basic authentication mechanisms that often used are OAuth 1.0, OAuth2.0 , Customized OAuth with provider specific User Id or credentials. For SOAP based connector, the authentication schema is usually basic credentials. The credentials will be passed in the header for each operation or a Session Id(generated from the login operation) will be passed in the header information for the operation. Here a new approach has been proposed to have single authentication playground to generate authentication data in order to connect and make an API call through connector. In Cloudstreams, this playground helps to build the connector faster without the overhead of Authentication data. The same can be referred to customer since it eases the authentication data generation to connect and use the APIs.

Keywords: OAuth, Cloudstreams, Connector, SOAP, REST.

1. INTRODUCTION

Every SaaS application provider will be using different authentication mechanism as per the requirement. API Developer should be aware of most widely used authentication scheme and should be flexible to generate the authentication data to connect to the SaaS provider and make API calls. It is tedious to generate authentication data manually every time for the providers like TOA-Technologies. A new approach is proposed to have a single playground with which the authentication data generation method can be chosen and the authentication data will be calculated automatically and can be used flexibly for API call or operation.

SOAP (Simple Object Access Protocol): SOAP is a platform, language independent, XML based communication protocol used for communicating between applications.

REST (Representational State Transfer): REST is an architecture style which relies on HTTPS and has guidelines and best practices for creating scalable web services.

OAuth: It is an open standard for authentication and a simple way to publish and interact with protected data. It provides client applications a 'Secured Access' to server resources on behalf of resource owner without sharing their credentials. It includes a Consumer Id and corresponding Consumer Secret[1] that together authenticate the consumer to the service provider.

Connector: Connector is a package that contains connections and services used to integrate with Software as a service (SaaS) provider and provide access to the instances simpler than adapters using SOAP or REST based APIs.

Cloudstreams is a packaged integration template which provides the details required to secure, govern and manage the communications between the applications at API level.

2. BACKGROUND STUDY

OAuth 1.0: For OAuth 1.0 the customer sets the Request URL, Authorization URL and Access Token URL for the application and uses the Consumer Id, Consumer Secret for SaaS application for Token request. The service provider sends the oauth_token and oauth_token_secret in the response after authorization. The oauth_signature is calculated using the Consumer Id, Consumer Secret, Access Token and Token Secret then passed to the Authorization header in the request for every resource/service call to the service provider. If the signature is not calculated properly, the request to the resource is rejected with authentication failure. Below is an example of authentication header data for with OAuth 1.0.

Authorization:
OAuth realm=https://quickbooks.api.intuit.com
oauth_consumer_key="73253d91qwe84q22",
oauth_token=" nnch734d(0)0sl2jdk",
oauth_signature_method="HMAC-SHA1",
oauth_signature="KSlgnzu41ntJp8UxlQhtjA4nfs4%3D",
oauth_timestamp=" 1433049644",
oauth_nonce="4572616e48616d6d65724c61686176",
oauth_version="1.0"

OAuth 2.0: OAuth 2.0 includes scope and grant_type and code [3] along with Client Id and Client Secret. The authorization code obtained after authorization is shared for getting the short term Access Token and optional Refresh Token. It avoids the overhead of cryptographic requirements of signing requests with Client Id and Client Secret involved in OAuth 1.0.

Below is the format of authorization header for OAuth 2.0 will be Authorization: Bearer $ACCESS_TOKEN

Example: Authorization: Bearer ya29.hAGvG3hNDKee2SX3tlQtBe1o0mVMjkj_n54YQs2sDrYfw3U2wOSIPtJEvrtFPA

MD5: MD5 is a message digest algorithm which is used as a hashing function producing 128 bit hash value and expressed as 32 digit hexadecimal value. It is typically used to hash the fields like password or Company ID which should not be passed as plain text.

3. DIFFERENT AUTHENTICATION MECHANISM USED BY REST-BASED SAAS PROVIDER.

Even though most of the REST based SaaS providers use OAuth, but they differ in the way they enclose the Authorization token. This authorization token will be passed in order to authorize the API call to backend.

A. Facebook

Facebook uses OAuth2.0 and allows the end client to create an app[4] on the Development environment. On app creation you will be provided with unique APP ID and its associated App Secret. The Facebook provides Graph API tool in Facebook developer environment which itself will fetch the APP Id and generate OAuth 2.0 Authorization token[5] without having to enter the Client Id and Client Secret again.

B.LinkedIn

LinkedIn follows OAuth 2.0 [6] with below steps.

Step 1. Asks user to make a below call in order to get the Authorization Code.

https://www.linkedin.com/uas/oauth2/authorization?response_type=code&client_id=123456789&redirect_uri=https%3A%2F%2Fwww.example.com%2Fauth%2Flinkedin&state=987654321&scope=r_basicprofile

Step 2.Allow the app to access the user account and get the access token.

Step 3: Get the Access token in the response of the below call

POST /uas/oauth2/accessToken HTTP/1.1
Host: www.linkedin.com
Content-Type: application/x-www-form-urlencoded
grant_type=authorization_code&code=987654321&redirect_uri=https%3A%2F%2Fwww.myapp.com%2Fauth%2Flinkedin&client_id=123456789&client_secret=shhdonottell

C.Twitter

Twitter uses OAuth 1.0 for authentication. The user has to pass the consumer id, consumer secret, Access Token, Token secret to generate signature and will be passed in the authorization header for each API call.

D.Amazon :

Amazon has its proprietary authentication mechanism Amazon Signing Version [7] in a customized way . It allows user to get the authentication detail through IAM console .Which has the class implementation like

```
static byte[] HmacSHA256(String data, byte[] key) throws Exception {
    String algorithm="HmacSHA256";
    Mac mac = Mac.getInstance(algorithm);
mac.init(new SecretKeySpec(key, algorithm));
returnmac.doFinal(data.getBytes("UTF8"));
}
static byte[] getSignatureKey(String key, String dateStamp, String regionName, String serviceName) throws Exception {
byte[] kSecret = ("AWS4" + key).getBytes("UTF8");
byte[] kDate = HmacSHA256(dateStamp, kSecret);
byte[] kRegion = HmacSHA256(regionName, kDate);
byte[] kService = HmacSHA256(serviceName, kRegion);
byte[] kSigning = HmacSHA256("aws4_request", kService);
return kSigning;
}
```

E. Salesforce Chatter

Even though Salesforce uses the OAuth 2.0, it authenticates the user through basic credentials but it expects password to have login credential appended with security token[8].

4. DIFFERENT AUTHENTICATION MECHANISM USED BY SOAP-BASED SAAS PROVIDER

The SOAP based provider will be using credentials for most of the connector authentication scheme through login operation. The credential will be used for subsequent call or the Session Id in the login response will be used for subsequent SOAP operation.

A.Zuora

Zuora authenticates the user by the registered Email Id and password as basic credentials through login call[9] and a Session Id will be received for the login call. The user can pass Session Id or credentials in the header for subsequent SOAP operation.

B.Melissa Data

Melissa Data provides a valid customer Id to the user and Customer Id will be the only data used to authenticate by Melissa Data for any operations,

C. TOA Technologies

TOA Technologies uses the registered company name, login name and auth_string[10] as authentication data.

auth_string will be md5(now+md5(password) where

now is different from current server and difference exceeds the predefined time-window

password is the login password set by the user for TOA Technologies.

5. NEED OF AUTHENTICATION PLAYGROUND

In the current scenario, the end user of the Integration provider has to follow the back ends authentication process. Authentication data is generated using

- The proprietary tool (if provided) by SaaS provider like Facebook provides Graph API Explorer.
- Third party tool like Google OAuth playground[2] (only for OAuth)
- The manual steps mentioned by service provider for authentication.

The calculated authentication data is entered in the connection details every time and which is tedious and time consuming. This playground can be easily integrated within the Cloudstreams for connections which in turn simplifies the API call for customer.

6. APPROACH

With the above observations it's possible to have single authentication playground in which the developer adds the authentication method he wants to have and can integrate the same in the connector development for connection.

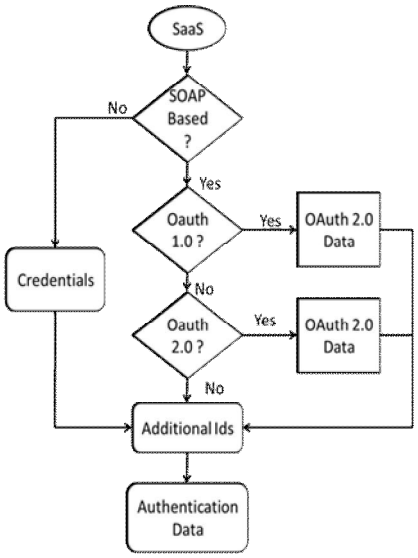

Figure I Authentication Playground flow chart

The current proposed model allows user to select the Authentication mechanism based on the requirement for the SaaS application.

If the connector is REST based, follows the OAuth 1.0 for authentication then an App is created for the user. The corresponding Client Id, Client Secret will be obtained from the app and with the app authorization from the account, Access Token and Token Secret will be generated by Auth Playground.

If the connector is REST based and follows the OAuth 2.0 then an App is created by the user. The associated Client Id, Client Secret will be obtained and corresponding Access Token and Refresh Token will be generated to use for API calls.

If the connector is REST based and neither follows neither OAuth 1.0 nor OAuth 2.0 but follows some custom IDs for authentication which can be plain or to be hashed data. It can be added through Additional Ids button. User can enter required number of unique Ids with key value pair. Also can use the utilities concatenate, append, hashing and other utilities like current time, date.

If the Authentication involves the customized Additional Ids along with OAuth 1.0 /2.0, those IDs can be supported by selecting Additional Ids after OAuth data generation.

If the service provider wants to authenticate with basic credentials then the Basic credential can be used and can be passed in the SOAP-Envelope.

If the authentication requires Additional Id parameter along with or without any other authentication data then parameters can be added through Additional Id. Multiple customized parameters for the connections can be added without overhead.

If the Authentication credential are not simple as in TOA Technologies and Salesforce then we can use Customized credentials which allows to add, current time, append string and hashing utilities if required.

The proprietary authentication mechanism like Amazon Signing Version-4 has also been included.

7. CONCLUSION

With this new approach it is easier to build the connector without having to spend more time for authentication. Since it allows the user to have customized credential where in user can add, append string, add current time and hash the data. It avoids the overhead involved in calculation. For customized hash based credentials, it saves 60% time and effort involved in calculation of Authentication data for back-end like TOA Technologies. The overall data generation time for time reduces by 30% with the new approach.

References

[1] http://hueniverse.com/oauth/guide/
[2] https://developers.google.com/oauthplayground
[3] http://tools.ietf.org/html/rfc6749#appendix-A.10
[4] https://developers.facebook.com/apps
[5] https://developers.facebook.com/docs/facebook-login/access-tokens#authClientServer
[6] https://developer.linkedin.com/docs/oauth2
[7] http://docs.aws.amazon.com/general/latest/gr/sigv4_signing.html
[8] https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/sforce_api_concepts_security.htm
[9] https://knowledgecenter.zuora.com/BC_Developers/SOAP_API/E_SOAP_API_Calls/login_call
[10] http://docs.oracle.com/cloud/latest/servicecs_gs/FAGPS.pdf page7-8 , chapter 2.2

AUTHORS

PRAKASH HIREMATH completed his B.E in Computer Science from Sir MVIT under Visvesvaraya Technological University in 2010 and is about to receive M.Tech degree in Computer Networks Engineering from BMSCE under VTU. He had worked as Software Engineer in Wipro and Intern in Software AG R&D in Bangalore.

Dr. P Jayarekha holds M.Tech (VTU Belgaum) in Computer Science securing 2nd rank and Ph.D. in Computer Science. She has nearly two decades of experience in teaching field. She has published more than 15 research papers in referred International Journals and also few in national and international conferences. Research Scholars are working on various fields like cloud computing, WSN and related areas under her guidance. Currently, she is working as Associate Professor in the department of Information Science.