ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Survey on OAuth as Bulwark for Authentication and Sanction

Rahul Ravindra Vishwakarma[1], Dr. Jayalekshmi K.R [2]

PG Student, Dept. of MCA, NCRD's Sterling Institute of Management Studies, Navi Mumbai, India[1]

Head of Dept. of MCA, NCRD's Sterling Institute of Management Studies, Navi Mumbai, India[2]

ABSTRACT: Gregarious media is ignited source of information for today's generation, Authentication and sanction becomes incredible consequential. OAuth works on TLS (Transport Layer Security) for building some aspect of sanction and server authentication. OAuth provides mechanism to resource owner that the clients can securely delegated access to server resources. Logging into another application from your current authenticated application, that can engender a vulnerably susceptible port for hackers. Here OAuth comes in, OAuth keeps your passwords safe on third-party applications. OAuth does not ask for password to access the application, instead it engenders the access token and gives access to the stuff which is sanctioned to the clients. This paper presents the mechanism of OAuth, Benefits of authentication from OAuth, comparative analysis of OAuth 1.0 and OAuth 2.0.

KEYWORDS: OAuth, Access token, Authentication, Sanction

I. INTRODUCTION

What is Authentication?
Authentication is process of verifying that the utilizer who is endeavoring to authenticate into system is valid or not. Authentication is done substratum of credentials provided to each utilizer customarily it is username and password.
It checks for the information stored into database and check for it, if the credentials match the utilizer is considered as Authentic Utilizer[1].

What is Sanction?
Sanction is a mechanism of security which defines the privileges and access list for a system files, resources, application and features.
During sanction, a system defines a rules and access sanction for a authenticated utilizer.
For Example: There is system which provides an accommodation of printing and dashboard. There are two group of users one is employees who has access to printing accommodations but not dashboard and another is group of admin who has both access. This bifurcation of access is maintained utilizing access levels kenned as sanction[2].

What is OAuth?
OAuth is an open standard mechanism which is utilized for authenticates users, delegate the access of resource on behalf of resource owner. This mechanism is utilized to apportion the data of owner with the third-party applications.
For Example: Some third party applications are linked to a primary application. If utilizer signed in any linked application to third party application and a utilizer cerebrates the application utilizes the credentials from currently logged account[7].
But this is major misconception of utilizer. No application should apportion a password to other third party application, but authentication and sanction is done utilizing OAuth which engenders an access token and sanction utilizer to utilize application[7].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

II. OAUTH MEHANISM

1. A bit history of OAuth

OAuth commenced around November 2006, while Blaine Cook was working on the Twitter OpenID implementation. He got in touch with Chris Messina probing for a way to utilize OpenID together with the Twitter api to delegate authentication. A meeting was set with David Record on and others at a CitizenSpace OpenID to get subsisting solutions. After reviewing subsisting OpenID functionality, as well as other industry practices, they came to the conclusion that there was no open standard for api access delegation. The conversation perpetuated online and off for a few months.[1]

2. Simplification of OAuth

Components to be included for implementation of OAuth:

A. Roles
B. Creating an application
C. Sanction the privileges
D. Making an Authenticated Requests

A. Roles:

- The Third-Party Application: "Client"

  The client is the application that is endeavoring to get access to the utilizer's account. It requires to get sanction from the utilizer afore it can do so[3].

- The API: "Resource Server"

  The resource server is the API server used to access the utilizer's information[3].

- The Sanction Server

  This is the server that presents the interface where the utilizer approves or gainsays the request. In more minute implementations, this may be identically tantamount server as the API server, but more astronomically immense scale deployments will often build this as a separate component[3].

- The Utilizer: "Resource Owner"

  The resource owner is the person who is giving access to some portion of their account[3].

B. Creating an application:

Before begin with implementation of OAuth, first register an application which will be used to communicate to resource server. This application will provide a Client Id and Secret Key.

Let's take an example of creating outlook application. You can create your own application from https://apps.dev.microsoft.com/. Here Client Id is Application Id and Secret key is Private Key.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

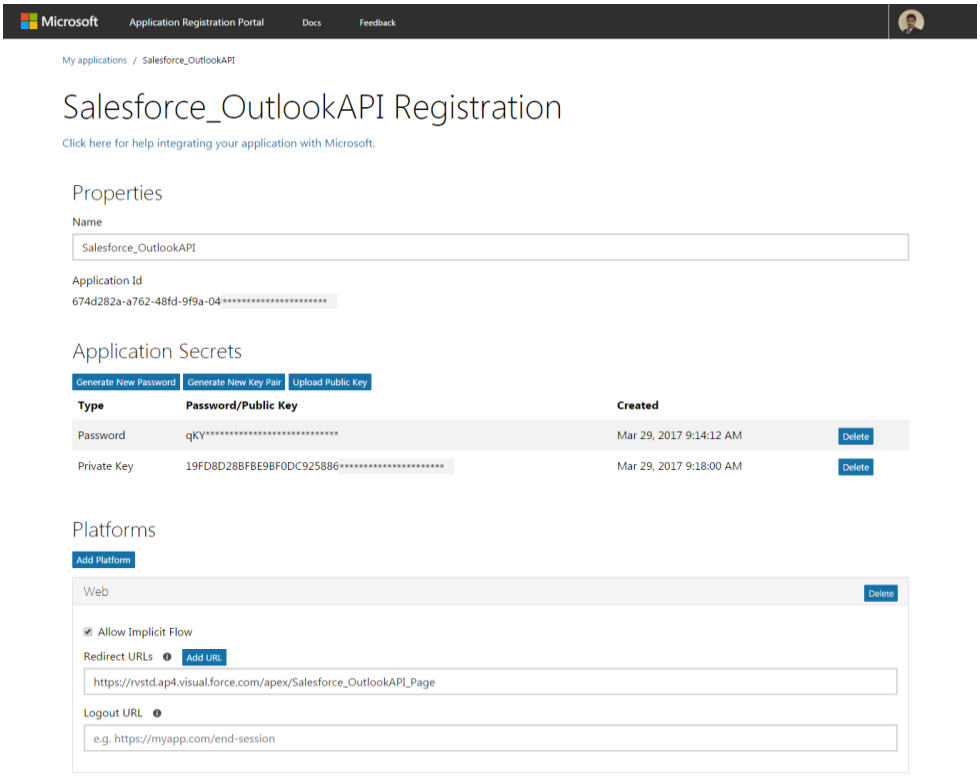

Fig. 1 Creating an application

C. Sanction the privileges:

We are going to use salesforce application and we will use outlook application for sending email using OAuth. For Authorization(Sanction) Hosted application will generate URL in such form:

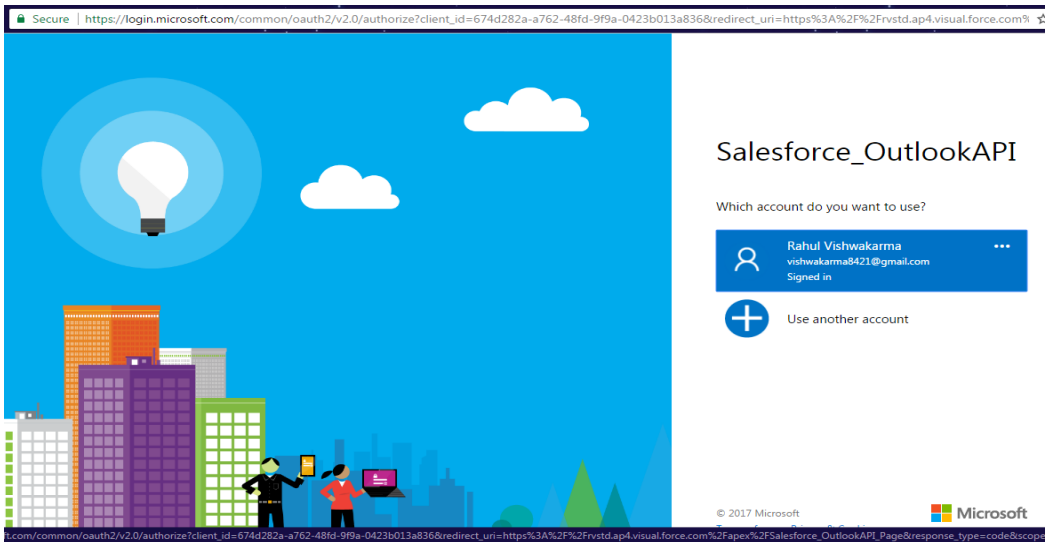

Fig 2. Sanction a user from outlook.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

https://login.microsoft.com/common/oauth2/v2.0/authorize?
client_id=674d282a-a762-48fd-9f9a-0423b013a836&
redirect_uri=https%3A%2F%2Frvstd.ap4.visual.force.com%2Fapex%2FSalesforce_OutlookAPI_Page&
response_type=code&
scope=openid+offline_access+profile+https%3A%2F%2Foutlook.office.com%2Fmail.readwrite+https%3A%2F%2Foutlook.office.com%2Fmail.readwrite.shared+https%3A%2F%2Foutlook.office.com%2Fmail.send+https%3A%2F%2Foutlook.office.com%2Fcontacts.readwrite+https%3A%2F%2Foutlook.office.com%2Ftasks.readwrite

Where,
Client_id: The client ID you received when you first engendered the application
Redirect_uri: Betokens the URI to return the utilizer to after sanction is plenary
Replication_type: betokens that your server expects to receive a sanction code
Scope: One or more scope values betokening which components of the utilizer's account you optate to access

After the authentication the endpoint of OAuth application sends a code and state as access token.

code - Designates that your server expects to receive a sanction code
state - A desultory string engendered by your application, which you'll verify tardy

D. Making an Authenticated Requests
The terminus result of all the grant types is obtaining an access token.

Now that you have an access token, you can make requests to the API. You can expeditiously make an API request.

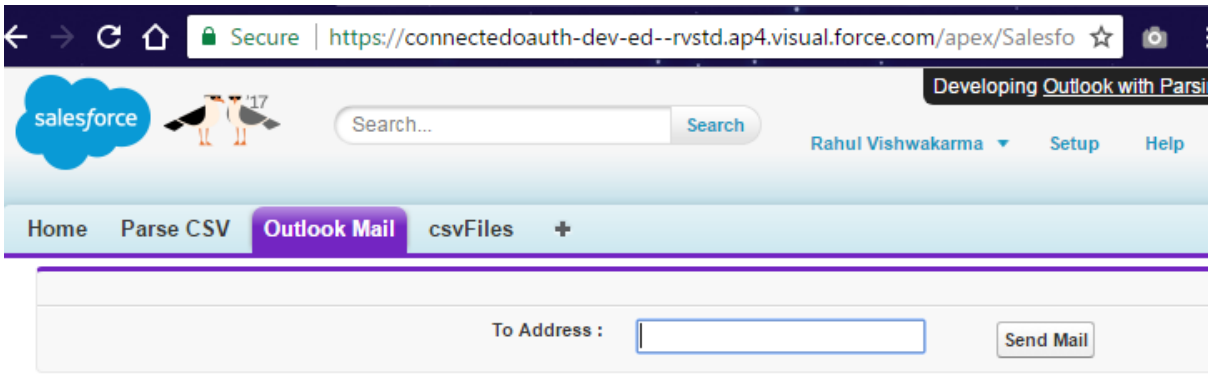

Fig 3. Making an Authenticated Requests

III. WORKFLOW OF OAUTH

In a mentioned scenario, if a company ask his developer to build an application which access the resources of clients or users from their own application. Developer might cerebrate of something which will provide security in aspect to verbalize with resource server and sanction some inhibited access. Here OAuth is most perfect to be implemented solution for above mentioned quandary.[2]

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

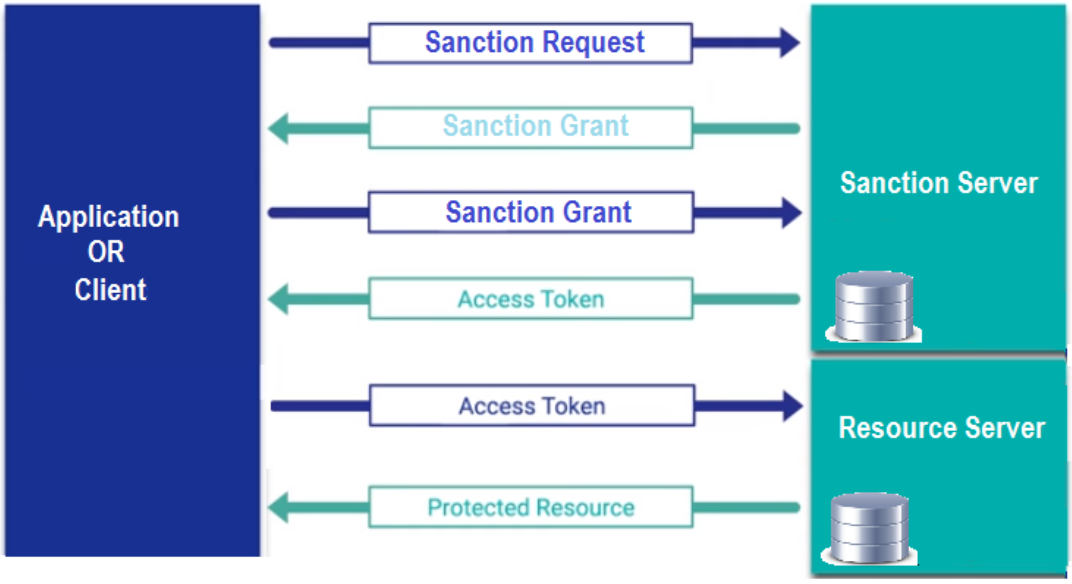

Fig. 4 Workflow of OAuth

IV. BENIFITS OF OAUTH

1. *API security:*

Every time a request is made to the API, instead of username and password, an access token is sent. This token is obtained by the client application before making the requests, and represents the utilizer on whose behalf the client application is utilizing the API [4].

2. *Internal enterprise applications:*

In all the other applications it simply redirected to the provider where utilizer authenticated in and attests that he wants to be sanctioned. This way, in lieu of storing passwords these applications are storing the tokens for the users. The benefit is that when a password is purloined, the utilizer has to reset his password, compared to when a token is purloined and it is revoked (invalidated) [4].

3. *More facile accommodation monitoring:*

Enterprises can track and monitor more easily which access token is making which request, predicated on this they can make calculations and gain better insight about which services are utilized more often by its clients, and make optimizations [4].

4. *Federated identity*

Another key strength of OAuth is federated identity. With federated identity, a person's digital identity and details (such as e-mail, name and cognomen, and gender) can be linked between several distinct accommodations [4].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

V. VERSIONS

Initially OAuth was implemented with versions as OAuth 1.0.Then OAuth 2.0 came over the drawback of the OAuth 1.0. OAuth 1.0 required signed tokens to process the OAuth, where as OAuth 2.0 does not uses signed tokens it works with access tokens[8].
The major drawback of OAuth 1.0 is that it does not have a refresh token for current processing connection. OAuth 1.0 would start the processing from beginning if connection is disturbed. But OAuth 2.0 has mechanism of refresh token with the access token, so that the refresh token would continue the connection[8].

VI. CONCLUSION

Authentication and sanction is necessary aspect to be looked after. Now a times many third party applications are hosted on an application, there comes a threat for credentials of resource owner. Even delegation is also most common aspect which ever resource owner likes to bring in action. For securing credentials, delegating the resource owner access levels is more practical and secure with the use of OAuth. The implementation of OAuth allows resource owner to delegate the access of resource with application without risking the credentials but with help of access and refresh tokens.

REFERENCES

1. Authentication,http://www.webopedia.com/TERM/A/authentication.html
2. Sanction introduction, https://en.wikipedia.org/wiki/Sanction
3. History of OAuth, https://oauth.net/about/introduction/
4. Introduction of OAuth 2.0, https://www.youtube.com/watch?v=CPbvxxslDTU
5. Roles in OAuth, https://aaronparecki.com/oauth-2-simplified/#roles
6. Benefits of OAuth, http://meuslivros.github.io/OAuth-2-0-Identity-and-Access-Management-Patterns/OEBPS/ch01s02.html
7. Introduction to OAuth, https://en.wikipedia.org/wiki/OAuth
8. Versions of OAuth, https://stackoverflow.com/questions/4113934/how-is-oauth-2-different-from-oauth-1

BIOGRAPHY

Mr. Rahul Ravindra Vishwakarma is a Post Graduate student of Master of Computer Application (MCA), College of NCRD's Sterling Institute of Management Studies, Mumbai University.

Dr. Jayalekshmi K.R is a Head of dept. of in Master of computer Application Department, College of NCRD's Sterling Institute of Management Studies, Mumbai University.