International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Identity Management as a Service in Cloud

Rahul Relan[1], Savaridassan P.[2]

[1]MTech (IT Department), SRM University Chennai, India

[2]Assistant Professor (I.T Department) SRM University Chennai, India

Abstract: ***Cloud computing Technologies have decreased the expense of advancement as expense of responsibility has been diminished and it empowers adaptable and proficient access to data. With these innovations there is inevitable risk of unapproved access of basic data. One of the regions that need to be checked is identity management. In this paper we are going to propose a conceivable answer to oversee identity management in cloud.***

Keywords: Identity Management, Federated Identity Management, OAuth, Single Sign On, Identity As A Service

1. Introduction

As these new cloud advancements rise, digital security difficulties connected with these innovations have expanded at a fast pace. One of the discriminating ranges that need consideration for secure cloud computing is identity administration where the different identity of cloud clients working potentially in a unified domain must be overseen and kept up. In this paper, we investigate identity management.

There is a basic need to safely store, manage, impart and examine massive amounts of complex (e.g., semi-organized and unstructured) data. The developing cloud computing model endeavours to handle massive amount of information.

Since the identities for a large number of users must be overseen in a cloud environment, we have to rethink the whole idea of identity management for the cloud. Users and also web services must be authenticated before accessing the assets. Single Sign-on is the mainstream arrangement where one time sign on gives a user of service access to different resources. Moreover, SAML and OAuth at present give authentication for web service. Nonetheless, with administrative necessities for e-business, and with the rise of the cloud computing standard, one needs a stronger way for authentication and this mechanism now be known as identity management.

2. Identity Management

Two concepts that are at the establishments of Digital Identity Management are (i) Single sign-on and (ii) Federated Identity Management. As expressed in, Single sign-on (SSO) is a property where a person logs in once and gets entrance to all services perhaps in a federation. Along these lines the client needs to log in once and has admittance to the resources in the organization or coalition without being provoked to log in again at each of them. Two sorts of SSO mechanism are Kerberos-based and savvy card based. With Kerberos mechanism, Kerberos ticket allowing ticket TGT is utilized to grant access. In the card based sign-on, the client utilizes the card for sign-on. Enterprise Single Sign-on (ESSO), gives the backing to minimizing the number of passwords and IDs when getting to different application. The use cases incorporate common utilization cases including cross-domain, online single sign-on.

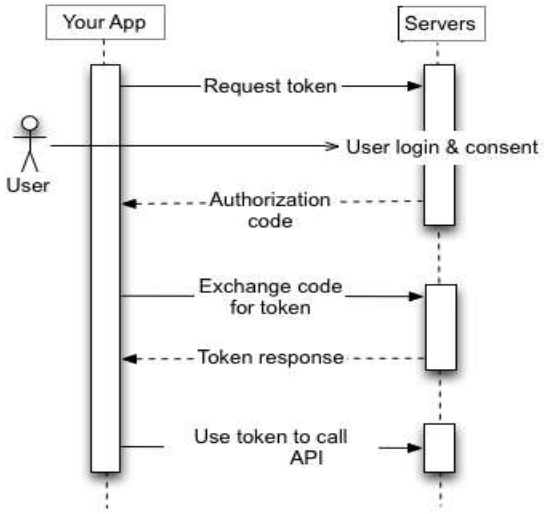

Figure1: OAuth

Various websites are presently using OAuth. OAuth is an open standard for approval. OAuth gives customer applications a 'safe designated access' to server resources for the benefit of an resource owner. It determines a procedure for resource managers to approve outsider access to their server without imparting their credentials.

3. Implementation

Conventional on-premises IAM arrangements aren't a solid match with SaaS applications. In today's period of cloud computing, it takes too long and costs are a great deal to actualize an old-school IAM framework. Such frameworks aren't sufficiently adaptable to handle new business courses of action or applications - particularly those outside the endeavor firewall, for example, SaaS applications - when they are added to the registering blend.

Presently there is a little however developing business for IAM offered as an service, or IDaaS. Enthusiasm for IDaaS originates from fair size to vast endeavours that need to oversee access to applications in the cloud, in addition to legacy on-premises applications. These associations need a solitary IAM arrangement that can give secure account

provisioning crosswise over both situations. They likewise need an answer that doesn't oblige an enormous investment in outside ability to create or modify all the application connectors.

An IDaaS for the undertaking is ordinarily bought as a membership based managed service. A cloud sevice provider might likewise have applications for an expense and give subscribers role-based access to particular applications or even whole virtualized desktops through a protected portal.

In our model we utilize OAuth to give Single Sign On and managed access to services in cloud. There are basically four main components of our model.
1. USER: The person who gets secure and managed services through secure portal.
2. Client-Agent: This is the secure portal or application in which user signs in.
3. Authorization server: This is the server that authorizes user to different web services or resource servers.
4. Resource Servers: The server that hosts the services that are being managed

Prior to our application can get to private information utilizing a API, it must get an entrance token that allows access to that API. A solitary access token can give changing degrees of access to numerous APIs. A variable parameter called scope controls the arrangement of operations that an entrance token licenses. Amid the entrance token demand, our application sends one or more values in the scope parameter.

A few applications require a confirmation step where the client logs in with their account. After logging in, the client is asked whether they are willing to concede the consents that your application is asking. This procedure is called client assent.

In the event that the client gives the consent, the Authorization Server sends your application an entrance token (or an approval code that your application can use to acquire an entrance token). On the off chance that the client does not concede the authorization, the server gives back an error.

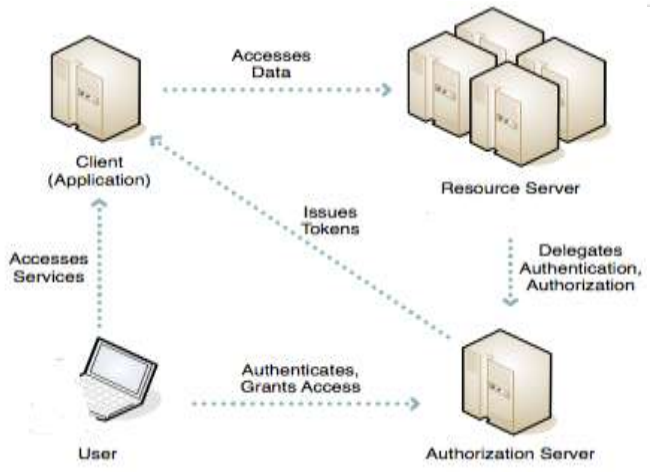

Figure 2: Architecture

After an application acquires an access token, it sends the token to REST API in a HTTP authorization header. It is conceivable to send tokens as URI query-string parameters, yet we don't suggest it, on the grounds that URI parameters can wind up in log records that are not totally secure. Additionally, it is great REST practice to abstain from making superfluous URI parameter names.

Access tokens are legitimate just for the arrangement of operations and assets portrayed in the extent of the token request. Case in point, if an access token is issued for the view profile API, it doesn't allow access to the Contacts API.

Access tokens have constrained lifetimes. On the off chance that our application needs access to an API past the lifetime of a solitary access token, it can acquire revive token. A revive token permits your application to get new tokens. Adding different services support to the secure portal gives the benefit of single sign on to diverse applications.

Profits of utilizing single sign-on include:
- Diminishing password weakness from diverse user name and password mixes
- Diminishing time spent re-entering passwords for the same identity
- Diminishing IT costs because of lower number of IT help desk calls about passwords

SSO brought together authentication servers that all different applications and frameworks use for validation purposes and joins this with procedures to guarantee that clients don't need to effectively enter their certifications more than once.

4. Conclusion

The above actualized setup has been tried with a few services supporting OAuth for approval and authentication. Such framework is profoundly effective in giving access to different services with same certifications with utilization of tokens. This facilitates the client work by not making him recall the qualifications. Also, it additionally gives better security by not writing the secret key and sends it in the request.

5. Future Work

Improving reliability for Identity and access management in cloud is critical to facilitating broader enterprise adoption. To meet this challenge, proper security methods need to be employed. Provide accessibility to services that do not bolster OAuth, SAML or other broadly utilized conventions.

References

[1] *https://developers.**google**.com/**apis**-explorer/*
[2] *en.**wikipedia**.org/**wiki/OAuth***
[3] https://www.**salesforce**.com/developer/docs/**api**/
[4] Lee, Hyangjin ; Inkyoung Jeun ; Hyuncheol Jung Emerging Security Information, Systems and Technologies, 2009. SECURWARE '09 Third

International Conference on DOI: 10.1109/SECURWARE.2009.31
[5] Balasubramaniam, S. ; Lewis, G.A. ; Morris, E. ; Simanta, S. ; Smith, D.B.DOI: 10.1109/SYSTEMS.2009.4815794
[6] *https://www.**okta**.com/*
[7] Samlinson,E.; Usha, M. Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on DOI: 10.1109/ICCCNT.2013.6726636

Author Profile

Rahul Relan is an M.Tech student in SRM University Chennai. His specialization is in Information Security And Cyber Forensics.

Savaridassan P. is Assistant Professor (Department of Information Technology) in SRM University Chennai.