International Journal of Engineering & Technology, 7 (2.6) (2018) 153-157

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Implementation of OpenId connect and OAuth 2.0 to create SSO for educational institutes

Tarun Sujanani[1]*, Smitha Vinod[2]

[1]Department of Computer Science, Christ University, Bengaluru, India
[2] Department of Computer Science, Christ University, Bengaluru, India
**Corresponding author E-mail: tarun.sujanani@mca.christuniversity.in*

Abstract

Increase in the number of users is directly proportional to the need of verifying them. This means that any user using any website or application has to be authenticated first; this leads to the creation of multiple credentials of one user. Now if these different websites or applications are connected or belong to one single organization like a college or school, a lot of redundancy of data is there. Along with this, each user has to remember a wide range of credentials for different applications/websites. So in this paper, we address the issue of redundancy and user related problems by introducing SSO using OpenId Connect in educational institutes. We aim to mark the difference between the traditional system and proposed login by testing it on a group of users.

Keywords: *SSO; OpenId Connect; OAuth 2.0; Education; Login*

1. Introduction

Education today has become far more than advanced than it was a decade ago. The technology boom in the education system is far more advanced than earlier. With easy access to the internet, the traditional education systems are replacing with digitalized ones. We can easily witness this shift by seeing the change in storage, access, maintenance of student/faculty details; we can also see that now online documents have taken space of pen paper-based assignments; not only this we can see mark sheet being distributed as well as attendance been taken online. All this digitization has led to different portals, websites, and applications etc. to come into existence. Each of these either store or provides access to a different set of data about students/faculties. For example, the student app or student login enables a student to view attendance, marks, timetable etc., whereas another web application enables the student to access all study materials, assignments etc. Now it is important to note that each of the portals has a different set of login credentials for each user. Each user is given a set of predefined credentials which generally do not have a provision to change. So now if a user forgets a password then he/she has to contact the higher authorities who provide him/her the password again. This also increases security risks in case the password is being hacked/ misused. If we consider of changing password then also we can see that it is difficult for a user to remember different user id and password combination. Here in this paper, we suggest establishing SSO for different logins using the Open ID Connect Provider and Relying Party. Also, we implement of Open ID Connect provider as well as deploy applications on it through the Relying Party. Here the user registry has the details of all the students/faculties. Now the basic idea whenever the user wants to access any application instead of separate login pages he/she will get one login page that sends data to the open id provider for verification.

2. Ease of Use

2.1. Existing Login Scenario

The existing login scenario provides the user with different login pages which validate the user using different credentials. These credentials are being maintained in a separate database which is linked with each website/application. These have a provision of sharing databases though. Now here the user logs in to the student login for checking his attendance, he has to use his id as university registration number and a pre-defined password. When checking this the user receives a mail and wants to check that also then he/she again logs onto the defined email using his/her email id(that the university has provided) and a password(may be same as one already provided or be changed). Now the user faces a problem of remembering passwords in case he/she changed it or remembering different login such as email id, registration id.

This becomes even a greater issue when at once the user needs to access more than 2 portals having a different login id and password provision. This problem intensifies when we consider the scenario where a new student is interacting with different logins, not only those that can be remotely accessed but also those applications that have different credentials for accessing library, repository or archives, LAN login details, intranet accessing etc.

The existing flow for authentication in education institutions is as follows:

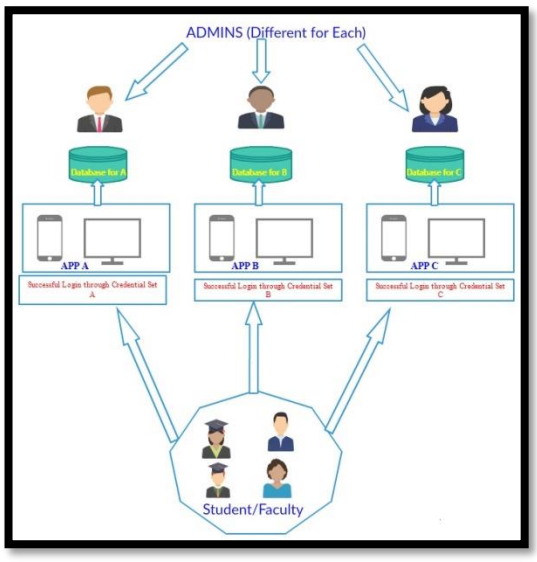

Fig. 1: Existing System Architecture

2.2. Proposed Open Id based Login

Here we propose a different approach where there is an Open Id Connect Provider. This particular server is responsible to hold all the user credentials like id, name, email [all the username used for different applications] and a password which is common to all usernames. Another part of this is the Open Id Connect Relying Party. One or more of these servers have the application deployed upon it. These applications when launched, the server sends a request to the Open Id Connect Provider. The Open Id Connect provider first verifies if the application requesting the single sign-on is a valid relying party or not. If the relying party is valid, then the provider provides a login to the application and authenticates the user. This is the same process that is repeated for all applications deployed on the relying party server.

The flow of the above-stated procedure looks like follows:

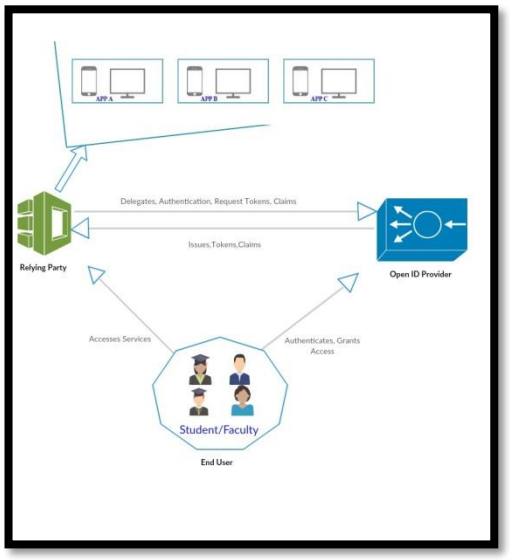

Fig. 2: Proposed System Architecture

2.3. Expected Advantages of Proposed System

The expected benefits of the proposed system are:-

- Better management of user details.
- End user has a better experience.
- No security concerns.
- Prevention of sharing of data to other SSO providers like Facebook, Google etc.

3. Literature Review

Oauth is the authentication protocol that came along with the rise of Facebook, Twitter, and their related applications [1]. Oauth enables SSO that enables users to be authenticated without sharing the credentials. The web developers and others if educated about this will result in proper implementation of SSO. But issues occur if the scope of the resource in consideration is limited, in this condition the user cannot know if any misuse is happening because there is no existing way by which the third party can access the resource technically.

A similar issue was found when Oauth was implemented in smartphones [2], here a new prototype was implemented and users tested that prototype for security and the results were compared with mobile browsing apps which do not use OAuth. It was found that the OAuth protocol was redirecting the browser to malicious apps that were not needed by the user, so to improve this, a built-in module was suggested along with the protocol.

GIAC Enterprises [3] also conducted extensive tests to ensure that application calls and traces that used OAuth 2.0 and caused security issues were handled by implementing a low cost, effective implementation fix. Thus the risks were mitigated by GIAC Enterprise.

OpenId Connect an OAuth based SSO [4] was tested with well-known SSO protocol attacks and its vulnerability was fixed using a better RFC draft. Thus it was proved that even though Open Id had proper countermeasures to most attacks it still needed a bridge between implementation and specification that is done by PrOfESSOS.

A similar analysis was also done to study the solution space of the problems that OpenId faced and its possible countermeasures were suggested by analyzing the advantages and drawbacks in detail.[5] So it was suggested as a long-term measure to redesign the URL parameters encoding and for short-term all OpenId transactions should mandatorily use HTTPS not HTTP.

ECC (Elliptic Curve Cryptosystem) along with OpenId [6] was proposed as an authentication scheme that would prevent eavesdropping, key control; attack, replay, and man in the middle attacks. This enabled a much more secure implementation in IoT than the normal security measures that were traditionally used in IoT earlier.

Similarly Trusted Platform Module (TPM), One Time Password (OTP) and Trust Multitenancy were applied along with OpenId [7] to stop phishing and identity theft. Thus using hardware-based activation and all of the above, a robust authentication module for cloud applications were created successfully.

Along with OpenId applying DI-r and PACS a new module for private and community cloud security was proposed [8]. This greatly helped in the area of medical imaging when the systems were equipped with this authentication module. So applying Security as a Service proper authentication for users was done on a medical imaging system.

In the field of e-learning, security is important so a UACM based SAML module is proposed that enables proper access of educational resources to users without involving a great amount of construction time and cost [9]. Thus elearning is improved by use of

RABC and UACM resulting in an overall increase in production and sharing of educational resources.
E-learning was also improved by use of iPLE networks [10]. These use Wen 2.0 tools to migrate the traditional VLE environment into more of updated secured and highly interactive one. This study was done on Students of Medicine and it showed a great improvement in resource sharing, security, and efficiency than traditional e-learning systems.
Finally, we see that not much is done to improve user authentication in education institutes apart from EduTone [11] that provides cloud-based single sign-on scenario along with data analysis provisions but again its application software of the third party that needs to have a detailed sharing of information from institutes. Similarly, OneLogin [12] also provides Identity Management and access by giving one-click options but faces the same issue of software deployment, sharing of data and possible lapse of security of data.

4. Implementation

We propose to implement this system using IBM's Liberty Websphere. We are going to create our Open Id Connect Provider and Relying Party on this server. The components used are as follows:-

4.1. Open Id Connect on Oauth2.0

Open Id Connect is a procedure of authentication. It is advancement over the well-known Open Id protocol. Open Id protocol is an open standard protocol for authentication. This basically authenticates users of sites which have been cooperated under Open Id Foundation. This protocol basically removed the middle man which used ad hoc systems to provide a login to users. The main aim is to authenticate a user through a single set of credentials [username and password] over multiple sites which are totally independent and unrelated to each other. It's important to note here that the Open Id protocol only authenticates the users, not authorizes it. This means that it allows a user to access a site but does not restrict what the user will view after successfully logging in.
The authorization feature was first done by OAuth standards. This is an open standard that helps to access delegation. In other words, it authorizes that the third party that has requested information has the access to the information that has been requested or not. Basically, it aids in validating parties when information is to be shared between them. This also specifies if the user after logging into site A will be able to view/use his information as available on site B.
An Oauth exchange looks like:

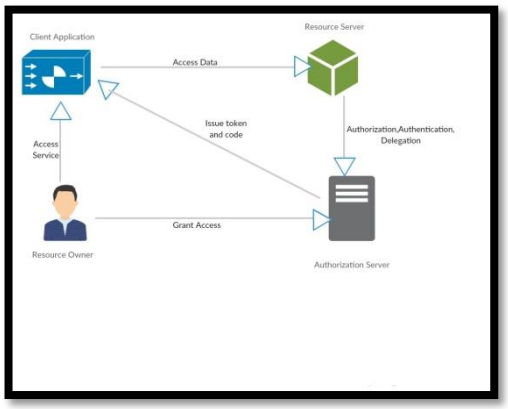

Fig. 3: Oauth Flow

This has been improved as in OAuth2.0 which details out the specific flow of authorization for a different application running on the web, desktop, mobile etc. OAuth2.0 is not a protocol per se; it rather falls into the category of a framework more. Thus this has a better implementation than the basic Oauth Protocol as it basically focuses on simplifying the client developer. It works on HTTP service using a REST API. It generally uses LDAP registry to authorize the clients.
Now Open Id Connect is an identity layer that is basically combined with the OAuth2.0 framework. This allows a User of Site A to get authenticated based on the credentials that he/she created while making an account/registering for Site B. A good example of this is YouTube that authenticates a user's age/ personal information through Gmail. Whenever a user wants to sign in on YouTube he/she uses Gmail to get authenticated and the information is shared. The Open Id Connect protocol basically uses RESTful API that is based on HTTP/HTTPS server. The basic data format that it uses for communication is JSON. As it a layer on top OAuth2.0 it works very well with a variety of client ranging from web to desktop to mobile. It also has data sharing feature, session management, encryption of data etc.

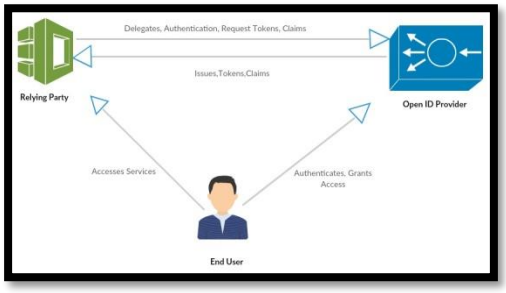

Fig. 4: OpenId Connect Flow

4.2. Liberty Websphere

Websphere by IBM is easy to use Java EE application server which is fast and dynamic. It is well suited for developers but can also be used for the production of cloud premise or library.
Liberty profile is an integral part of the Websphere version V8.5.5.5. Lightweight, dynamic and flexible liberty profile supports WAS server to deploy only the custom features which are required instead of deploying the whole java JEE package. The features included in this supports a wide variety of business requirements and helps to push the application made on the server.
Apart from suiting the developer community for creating enterprise level application, Liberty is also well suited for deployment of a new application.
The Liberty Websphere architecture includes the following components:

- Liberty Kernel: This is the main server profile of Liberty Websphere.
- Feature: Servlet, Web App Security, Java Servlet Framework, Java Server Pages and JMS.
- Java EE 6+: The standard Java EE6 API.
- OSGi Framework Runtime: The built-in bundles that help to run time.
- Application: Web Apps and Enterprise Applications.

5. Experimentation and Results

The proposed system is implemented using Liberty Websphere deployed on 2 different systems namely System A and System B. System A will be configured to have the settings of the Open Id provider and System B will have the Relying Party which will have the deployed application in the form of a .war file.
After initial setup, we have configured our servers and deployed the applications,

We launch the servers to establish the SSO, once the server and SSO are launched; we verify the user based on the credentials that we have specified.

For our experimentation, we have taken 5 users. These users are novice and have not been using the websites/apps that need authentication. So it is safe to assume not all the users remember the password and id combinations of the site.

So we asked the user to login into the university Gmail and the student login. In general scenario we noted that:

- Student often got confused about the Username especially for email as it was a custom made one as given by university on its domain.
- Some students who had changed their passwords faced issues when asked to switch email to login which verifies only the password that was handed out.
- It even took some seconds for the user to realize where they were actually trying to log in and what the last credentials they recalled were.

Then the same 5 users were asked to log in to the same sites and were given the login page from our proposed system, we saw that:

- Users did not take time to realize where they were logging in as all usernames, id, and email were working for one particular password.
- For any user who had updated their email password could log in using that for student login too.
- The users entered whatever username and password combination they recalled and got verified despite what was the site they were trying to log in.

So we summarize that:

Table 1: Analysis Report 1

System	Parameters		
	Same login for different sites	Password Change updates on all sites	Pre Processing Need
Existing System	NO	NO	Some Seconds Needed
Proposed System	YES	YES	NIL

So we can see that the existing system has many types of fallout, especially regarding actions when concerning a new user. Here we can see that when the user is not habituated to the environment it takes a bit of time to process and recall the needed information. If an experienced user is working on the existing system he or she would not face this problem. But we cannot assume that even the experienced user will recall all credentials without taking any time to process.

So we can see that for new users the proposed system works a lot better. So now we will analyze if compared to the existing system does the proposed system has same if not better performance metrics. For this analysis, we have taken one Existing System Site and mapped their data after logging in with the same for one login page of Proposed System. The metrics we are taking into account are:-

1. DNS Lookup
2. TCP/IP Connect
3. HTTPS Handshake

We executed both our existing and proposed system and monitored these above-stated parameters using Fiddler by Telerik.

We tabulated the results as follows:

Table 2: Analysis Report 2

System	Parameters		
	DNS Lookup	TCP/IP Connect	HTTPS Handshake
Existing System	56ms	12ms	99ms
Proposed System	54ms	13ms	103ms

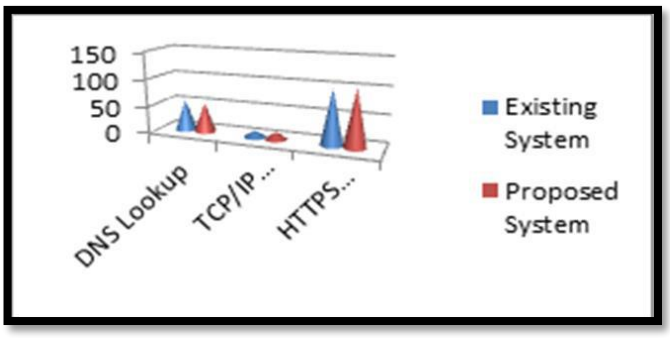

Fig. 5: Graphical Representation of result

6. Conclusion

In our Proposed System, we implemented the Open Id Connect Single Sign-On Scenario and successfully launched our login page on that. Here we saw that in comparison to the existing system our proposed system was better at handling new user and their credentials verification process. It enables the new user to interact with system freely without recalling multiple credentials and combinations.

In the proposed system, we can apply new user-oriented data such as password updates to what is previously specified and the updating will be uniform over all applications unlike the existing system

In the proposed system we are creating one single place to store all the credentials; here passwords can be changed for all sites. And verification is done from there instead of applying a database of credentials, login page, and a verification module for all our applications. This reduces the individual load of all the application and allows them to offer better functionalities. This also means that if due to some reason there is a fault in the database the users will not get properly verified, here in the proposed system if some username password combinations are not giving correct results we can verify using the same password for a different username like email or id etc.

The working time of both systems are approximately same but the proposed system has more security to offer as the password and username combination, as provided is prone to change, so even if the username may be known the password is still unknown. This does not happen in existing system as the same password is used again and again without any scope of change.

So we can sum up that considering the advancement we can only say that users are going to increase. To manage them we need more advanced techniques thus implementing Open Id will not only provide a shift from the database but also decrease dependency of verification on individual applications. Creating a uniform verification model that enables access from the shared applications and verifies the user without hassle is needed. Moreover this secures sharing of data as if the same SSO was to be implemented using some 3rd party like Google, Facebook etc. then the user information will have to be, either generated and managed by the

party as it happens in Facebook SSO, or the data created by us has to be completely shared and opened for manipulation by the third party as in Google SSO.

Thus finally we can say implementing Open Id Connect using Oauth does not only make it easier for the user but the organization benefits from it on the whole.

Acknowledgement

I would like to thank my guide Prof Smitha V to be a pillar of support through the writing of this paper. I would thank my institution to provide me this opportunity. Lastly, big thanks to god almighty, my friends, and family who gave continuous support and had faith on me.

References

[1] Peplin, Christopher. "OAuth for Privacy." 2012-03-11]. http://christopherpeplin. com/201 1/05/oauth-privacy (2011).
[2] Ho, Lee Kah, and Norliza Katuk. "Social login with OAuth for mobile applications: User's view." Computer Applications & Industrial Electronics (ISCAIE), 2016 IEEE Symposium on. IEEE, 2016.
[3] Quick, Brian, Russel Van Tuyl, and Sumesh Shivdas. "OAuth 2.0 Vulnerability Impact Study."
[4] Mainka, Christian, et al. "SoK: Single Sign-On Security–An Evaluation of OpenID Connect." (2017).
[5] Uruena, Manuel, and Christian Busquiel. "Analysis of a privacy vulnerability in the openid authentication protocol." IEEE Multimedia Communications, Services and Security (2010).
[6] Lee, Jong J., Youn-Sik Hong, and Ki Young Lee. "A Study of User Authentication Protocol Based on the ECC and OpenID Techniques in the Internet of Things." (2016).
[7] Ghazizadeh, Eghbal, et al. "Secure OpenID authentication model by using Trusted Computing." Abstract and Applied Analysis. Vol. 2014. Hindawi Publishing Corporation, 2014.
[8] Ma, Weina, et al. "OpenID Connect as a security service in cloud-based medical imaging systems." Journal of Medical Imaging 3.2 (2016): 026501-026501.
[9] Shang, Chaowang, et al. "SAML Based Unified Access Control Model for Inter-platform Educational Resources." Computer Science and Software Engineering, 2008 International Conference on. Vol. 5. IEEE, 2008.
[10] Casquero, Oskar, et al. "iPLE Network: an integrated eLearning 2.0 architecture from a university's perspective." Interactive Learning Environments 18.3 (2010): 293-308.
[11] EduTone - Cloud single sign-on solutions for schools and colleges. [online] Available at: https://www.edutone.com/ [Accessed 4 Aug. 2017].
[12] OneLogin. (2017). Single Sign On Solutions for Education - Identity and Access Management for K-12 & Higher Ed. [online] Available at: https://www.onelogin.com/solutions/education [Accessed 7 Aug. 2017].