An OAuth-Based Authentication System for IoT Networks Using LabVIEW

P. Kalpana Devi, M. Manasa, S. Naga Chandra Prakash, and B. Vittal Teja

Abstract OAuth is a significant authentication method used for ensuring the security of Internet of Things (IoT). In Internet of Things (IoT) networks, OAuth is a framework used for incorporating authentication, which has developed the workflow model for web server applications. This framework offers authentication to the web service through which the user's account is hosted, as well as secure API access for external users. To protect server resources and enhance scaling and distribution, a dedicated authorization framework is required. Henceforth, this research work has suggested OAuth2.0 for accessing data from IoT devices with sensors, which allows the users to maintain precise records of temperature and pressure in industries. It is informed that, creating APIs in other operating systems is difficult. For developing the Web services API, the API tokens are used to include the user account into the access token. OAuth2.0 offers authorization without requiring a user account. To address the insecurity issue, we integrate the OAuth2.0 process with the API in order to securely access data. OAuth2.0 grants the client "secure delegated access" to the server.

Keywords OAuth2.0 · LabVIEW · API · Server · Web servers

P. Kalpana Devi (✉) · M. Manasa · S. N. C. Prakash · B. V. Teja
VelTech Rangarajan Dr. Sagunthala, R & D Institute of Science and Technology, Chennai, India
e-mail: drkalpanadevip@veltech.edu.in

M. Manasa
e-mail: vtu10533@veltech.edu.in

S. N. C. Prakash
e-mail: vtu9814@veltech.edu.in

B. V. Teja
e-mail: vtu9945@veltech.edu.in

P. Karrupusamy et al. (eds.), *Sustainable Communication Networks and Application*,
Lecture Notes on Data Engineering and Communications Technologies 93,
https://doi.org/10.1007/978-981-16-6605-6_46

1 Introduction

IoT security has scope for widespread IoT visions such as trusted sensing, communication, computation and security. Now a days most of the users suffers for securing the important data while login/signup with their email or password. In view of overcome, the problem of insecurity of the personal data of the users OAuth2.0 plays an important role in providing such delegated services where in which their username or password are not stored, with the help OAuth API'S it establish the secure connections to users [1]. OAuth 2 is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service, such as Facebook, GitHub, and DigitalOcean or any application of the users. It works by delegating user authentication to the service that hosts the user account, and authorizing third party applications to access the user account. The proposed research work intends to design a delegated authorization framework for the API called as OAuth2.0, which primarily create LabVIEW web service because as some users will attempt to use the toolkit on other operating system. Also, certain functions are not supported and it is more important to make a design decision for target operating systems and hence it is difficult for the user to design the API in other operating system and so we know that "ALL WEBSERVICES ARE API," so in order to overcome the difficulty we have decided to create a Web services, secondly we are going to enable the OAuth2.0 workflow in the API /web services as the API tokens incorporate the user account in the access token, while OAuth2.0 performs authorization without a user account so in order to overcome the difficult of insecurity [2], we add the workflow of OAuth2.0 with the API in order to access the data more securely OAuth2.0 provides to client a "secure delegated access" to server resources on behalf of a resource owner. In this project, OAuth2.0 plays important role in accessing the data from the IOT device such as Aurdino UNO Device and helps the users to make the precise record of the temperature, pressure in the Industries. OAuth 2.0 provides authorization flows for web and desktop applications, and mobile devices. In this project, we aim to create a web application in LabVIEW where we have enabled the OAuth2.0 work flow in the API'S for enabling the service for the third party applications for the signing into the particular web application created for controlling the industry related problems such as pressure temperature by using Arudino UNO interfacing with the LabVIEW. OAuth 2.0 is an authorization framework for enabling resource sharing in a secured manner through a sequence of steps where resource owner permits a client application to a certain protected resource for a limited time. By using delegation pattern, client applications will be enabled by OAuth framework [3]. From existing resource server, some functionalities can be delegated without repeation. For access the files on a Google Drive is made through OAuth-based authorization. A website provides single sign-on, utilize your existing Facebook, Google or Twitter account through OAuth-based authorization and then continue using that site without signing up with a separate account on that site. We have used the Arduino UNO, Breadboard, Temperature sensors MakerHub linx and NIVISA package and LabVIEW interface with the arduino.

2 Methodology

2.1 *Creating the Web Server*

HTTP protocols are used in web server and for responding the client requests made with WWW [4]. The website content is displayed in web pages with storing, processing and delivering to users. Client server model is working in a similar process with web server process. Website content is delivered to the requesting user through the web server software [5]. The domain names can be used for accessing the web server software. HTTP server is the key component in software. HTTP and URLs can be handled with HTTP server.

Web server software and website contents are stored in web server. Website contents are HTML documents, images and Javascript files. Webserver should consists of web browser information like google chrome and Firefox files [6]. The request is made through HTTP. When request raised through HTTP to web server, the content requested is identified and sent back to the browser requested. Even more a specific page can also be requested through HTTP by web browser with series of processing steps.

Web browser URL address link to be specified. The domain name IP address will be provided to the web browser.

The domain name is provided with Domain name system or search in Cache to enable the browser. The browser will then make another HTTP request for a specified file. The webserver accepts the request through HTTP and returns the desired page. If that particular page is not available, the web server will issue an error message.

LabVIEW web service request proceeds with refnum by identifying the current HTTP request as mentioned in Fig. 1 Interfacing Makerhub with LabVIEW for LED lighting by using arduino board as an example program to carry out the operation. Include this file in your web resource to allow access to the web server.

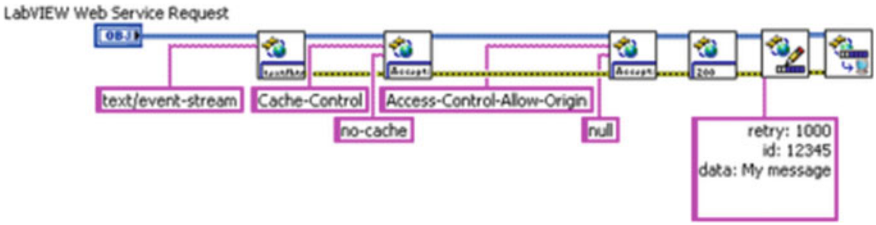


Fig. 1 Creating the web server

2.2 OAuth in LabVIEW

The limited access will be provided to HTTP service by any third party application and further it will be done by OAuth 2.0 application framework.

This access is provided through the approval interaction on behalf of resource owner with HTTP service. The third party can also access the service on its own behalf.

The OAuth2.0 provides the limited access for the application in order to access the protected resources. User does not require their login credentials to access the application. This application sends a request to the server in order to access the protected assets from the owner's resources on mobile devices and standard web applications. The request to access was granted by the owner.

3 Simulation of OAuth2.0 in LabVIEW

Specification

Facebook and Google HR services in intranet are the services offered by the resource server. When OAuth 2.0 token is raised, validation is checked further to process API requests from Apgee edge. The authorization is required for the resource server, especially for the application's protected resources (Fig. 2).

The OAuth 2.0 specification is validating the authorization grants from server and it also issues the access tokens. The token endpoints is required to be configured on Apgee edge. The authorization server is operated by Edge. The application permission is provided by authorization grant for retrieving the access tokens [7, 8]. Four specific grant types are defined by OAuth 2.0. To access the protected resources,

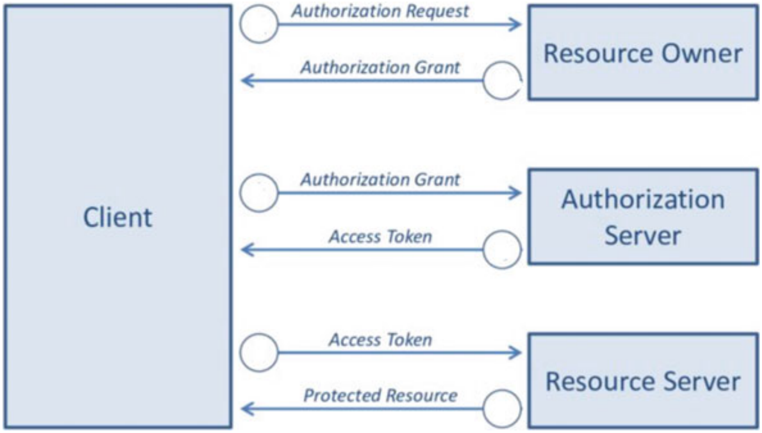


Fig. 2 Flow chart for OAuth 2.0 framework

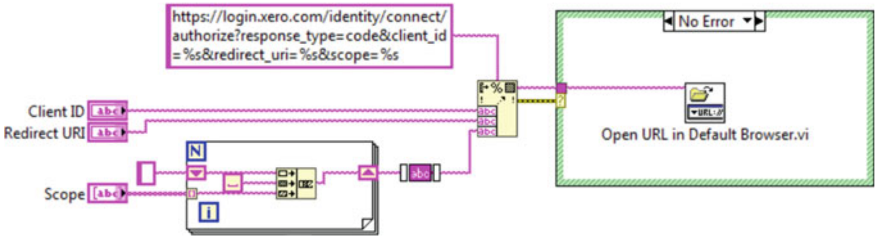

https://login.xero.com/identity/connect/
authorize?response_type=code&client_id
=%s&redirect_uri=%s&scope=%s
No Error
Client ID
Redirect URI
Scope
N
i
Open URL in Default Browser.vi

Fig. 3 Overall block diagram of string to Base64url

a long character string called "access token" is used. Resource owner handles the protected data resource such as the user's contact list, account information and other sensitive data. The steps of OAuth 2.0 flow for creating the webserver, getting the url of the web server is as follows:

3 Steps to design OAuth2.0

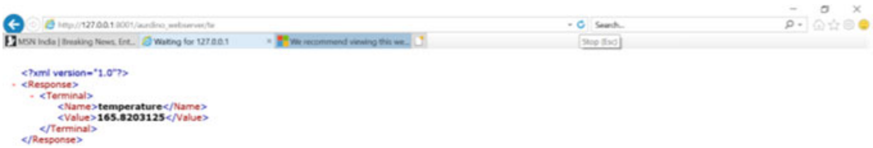
1. String to Base64url (Fig. 3)

Publishing Web services

A stand alone web services called Application web server is published in LabVIEW for accessing Web server. The stand alone web service applications is hosted by web server in a network. To protect the network data exchange, the multiple security related features is provided along with secure Socket Layer (SSL) encryption. This method is executable for even complex application.

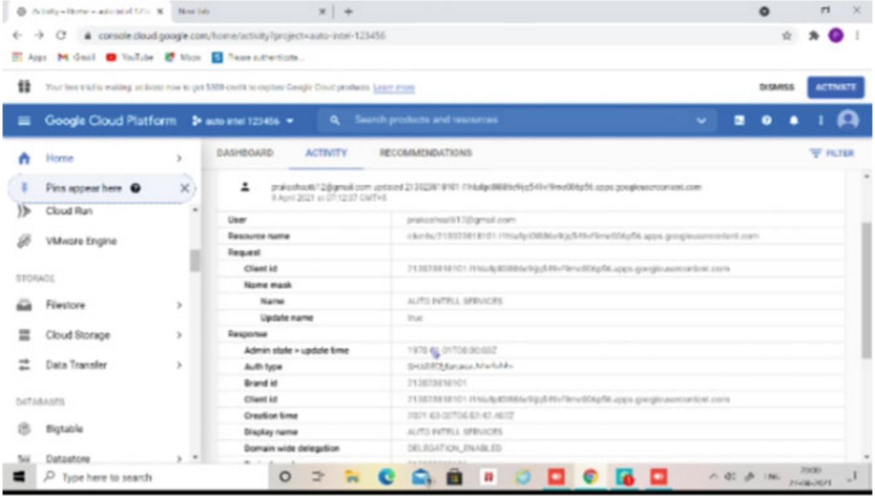
The web service handles the different web servers in stand alone applications. As a result, the application web server publish should be selected and the deployment progress dialog box appears. If the web service is successfully published, the close button will be selected. Now, data exchange from web clients will be established via HTTP method. As mentioned in debugging, select the web service project item and choose the application web server in order to manage the web server. Web browser opens the NI web-based configuration for monitoring. Then, the web services management and tutorial service will be selected from the list. If the list is shown as empty, the process will be refreshed. Also, ensure that the web service status is in running mode. To make any further adjustments, such as resume, restart, or unpublished web service, use the buttons at the bottom of the page. To register the firm name with the Google Console Developers, first connect to the Google Cloud Platform with the appropriate gmail id and accept the required access credentials requirements (Fig. 4).

For credentials, enter our firm name or any other app name under project name, choose the place where it is located, and click on as indicated. Next, go to the OAuth consent screen and enter the company under the external and click on next to update the procedure. Now fill the required things such as the app name and domain name, respectively, in the OAuth consent form and click on save and continue to move forward for the further process and similarly fill the scope and next process by save



```
<?xml version="1.0"?>
- <Response>
  - <Terminal>
      <Name>temperature</Name>
      <Value>165.8203125</Value>
    </Terminal>
</Response>
```

Fig. 4 Output readings from sensor ınto the webpage in xml format



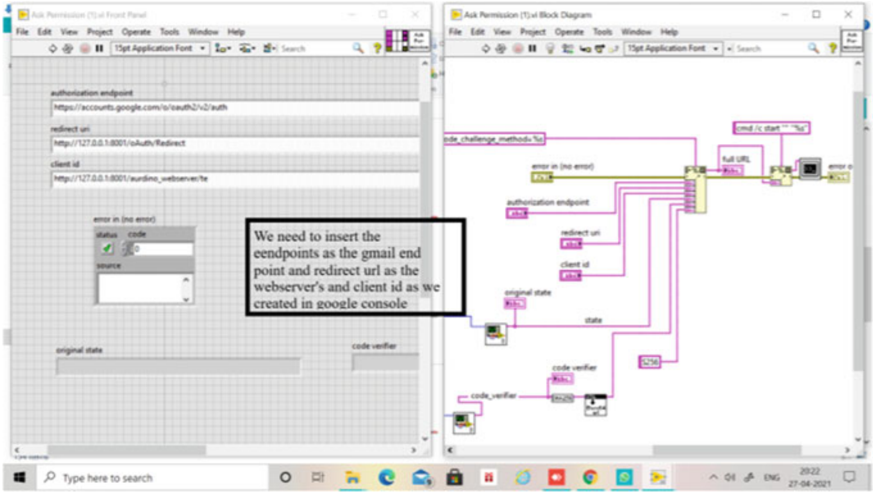
User	[illegible]
Resource name	[illegible]
Request	
Client id	[illegible]
Name mask	
Name	AUTO INTELL SERVICES
Update name	true
Response	
Admin state > update time	1970-01-01T00:00:00Z
Auth type	[illegible]
Brand id	[illegible]
Client id	[illegible]
Creation time	[illegible]
Display name	AUTO INTELL SERVICES
Domain wide delegation	DELEGATION_ENABLED

Fig. 5 Client Id and secrete code of the company

and continue. again open the credentials and open on the company name and register the redirect URL, which is created by the users using LabVIEW webserver a new client id and secret code is released for different redirect url you will get different client id and secret copy paste all the required things in the LabVIEW code such as redirect link client id and client password as shown in Fig. 5 and inserting the sample inputs as shown in Fig. 6 .

4 Conclusion

From this research work, it is evident that, nowadays, people transfer most of their day-to-day activities to be computerized. Traditionally, different industrial activities are monitored by using a camera, which can also lead to some errors. To overcome such challenges, this research work has successfully created a webserver with OAuth2.0 authentication. In this, we need to enter the industry and also the user name and password should also be entered; this may also create a fear of stealing important information or accessing important things, or it may cause insecurities.



authorization endpoint
https://accounts.google.com/o/oauth2/v2/auth

redirect uri
http://127.0.0.1:8001/oAuth/Redirect

client id
http://127.0.0.1:8001/aurdino_webserver/te

error in (no error)
status code
0
source

We need to insert the eendpoints as the gmail end point and redirect url as the webserver's and client id as we created in google console

original state

code verifier

Fig. 6 Insertıng the sample ınputs

Henceforth, this research work has integrated OAuth2.0 with LabVIEW for security purposes and also efficiently developed a web server, where one may retrieve the data of the sensor or any data from their location.

References

1. The OAuth 2.0 Authorization Framework. https://tools.ietf.org/html/rfc6749. [Online]; Accessed October 2019. User Authentication with OAuth 2.0.https://oauth.net/articles/authentication/. [Online]; Accessed October 2019
2. R. Buyya, A.V. Dastjerdi, Internet of things: principles and paradigms. (Elsevier, 2016)
3. M. Picone, P. Gonizzi, L. Veltri, G. Ferrari, IoT-OAS: an OAuth-based authorization service architecture for secure services in IoT scenarios simone cirani. IEEE Sensor J.
4. K. Yokogi, N. Kitagawa, N. Yamai, Access control model for IOT environment including automated configuration. in *2018 IEEE 42nd Annual Computer Software and Applications Conference* (COMPSAC), Tokyo, Japan, July 2018, vol. 02 (IEEE, 2016), pp. 616–621
5. National Center for Biotechnology Information. http://www.ncbi.nlm.nih.gov
6. C.-J. Chae, K.-N. Choi, K. Choi, Y.-H. Yae, Y. Shin, The extended authentication protocol us-ing e-mail authentication in OAuth 2.0 protocol forsecure granting of user access. J Internet Comput Services (JICS) 16(1), 21–28 (2015)
7. D. Dolev, A. Yao, On the security of public key protocols. IEEE Trans. Inform. Theory **29**(2), 198–208 (1983)
8. X. Hang, J. Li, Z.Z. Xu, D. Feng, H. Hu, Multiple handshakes security of TLS 1.3 candidates. in SP. (IEEE, 2016)
9. A. Kumar, Using automated model analysis for reasoning about security of web protocols. in *ACSAC* (ACM, 2012)
10. N. Fotiou, G.C. Polyzos, Authentication and authorization for interoperable iot architectures. in *International Workshop on Emerging Technologies for Authorization and Authentication* (Springer, Barcelona, 2018) pp. 3–16

11. A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, A. Panya, Authorization mechanism for mqtt-based internet of things. in *2016 IEEE International Conference on Communications Workshops* (ICC). Kuala Lumpur, Malaysia, May 2016 (IEEE), pp. 290–295
12. H. Ning, H. Liu, L.T. Yang, Cyberentity security in the internet of things. Computer **46**(4), 46–53 (2013)