In this thread I would like to discuss OAuth as well as OpenID Connect. It will be my aim to compare and contrast the two by using tables to show their unique pros and cons. To understand what both are I do believe it is important to have a definition of each as well. Let’s start by defining them.

“OAuth is an open-standard authorization protocol or framework that provides applications the ability for “secure designated access.” For example, you can tell Facebook that it’s OK for ESPN.com to access your profile or post updates to your timeline without having to give ESPN your Facebook password. This minimizes risk in a major way: In the event ESPN suffers a breach, your Facebook password remains safe.” (Sobers, 2022). Some of the pros and cons of OAuth are presented in the table below:

|  |  |
| --- | --- |
| **OAuth Pros** | **OAuth Cons** |
| OAuth security tokes are great at authorizing levels of access for specific users. | OAuth security is less popular (and less commonly understood than API kyes. |
| OAuth security tokes can be set to expire. | OAuth security is more complicated than API key security. |
| OAuth security tokens offer exceptional access to user data. | Oath security tokes are best used in authorization headers. |

(Harris, 2022).

“OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It allows Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner. OpenID Connect allows clients of all types, including Web-based, mobile, and JavaScript clients, to request and receive information about authenticated sessions and end-users. The specification suite is extensible, allowing participants to use optional features such as encryption of identity data, discovery of OpenID Providers, and logout, when it makes sense for them.” (OpenID, 2022). Some of the pros and cons of OpenID Connect are presented in the table below:

|  |  |
| --- | --- |
| **OpenID Connect Pros** | **OpenID Connect Cons** |
| You can offload the authentication of a user to an OpenID provider such as Yahoo! or Google. Using this method, you can take advantage of the provider’s large membership and security systems to log your users in to your site. | OpenID is simply an authentication service for verifying a user account state, not an authorization system like OAuth, which allows an application or service to perform actions on the user’s behalf once authorized. What this means is that a simple OpenID integration will not be able to make signed requests to the provider site to get, set, or delete a user’s social information. |
| You will not need to store user login credentials in your own database systems; rather, you simply map the OpenID user on the provider site with whatever information your application or site stores about that user. | The support for OpenID extensions—such as Simple Registration, Attribute Exchange, and PAPE—is inconsistent from provider to provider. Some providers support all the most popular extensions, while others support none. In addition, the personal information that you can obtain through such extensions varies among providers. |
| The straight OpenID approach is more lightweight than the hybrid auth implementation. |  |

(Programming Social Applications, n.d.).

References:

Sobers, R. (2022, June 29). *What is OAuth? Definition and How it Works*. Retrieved November 20, 2022, from <https://www.varonis.com/blog/what-is-oauth>

Harris, T. (2022, July 13). *How to Secure REST APIs: API Keys Vs OAuth*. DreamFactory Software- Blog. Retrieved November 20, 2022, from <https://blog.dreamfactory.com/how-to-secure-rest-apis-api-keys-vs-oauth/>

OpenID. (2022, November 15). *OpenID Connect | OpenID*. OpenID - the Internet Identity Layer. Retrieved November 20, 2022, from <https://openid.net/connect/>

*Programming Social Applications*. (n.d.). O’Reilly Online Learning. Retrieved November 20, 2022, from <https://www.oreilly.com/library/view/programming-social-applications/9781449302481/ch12s02s02.html>