

[Join us on the demo](#), while our product experts provide a detailed walkthrough of our enterprise platform.



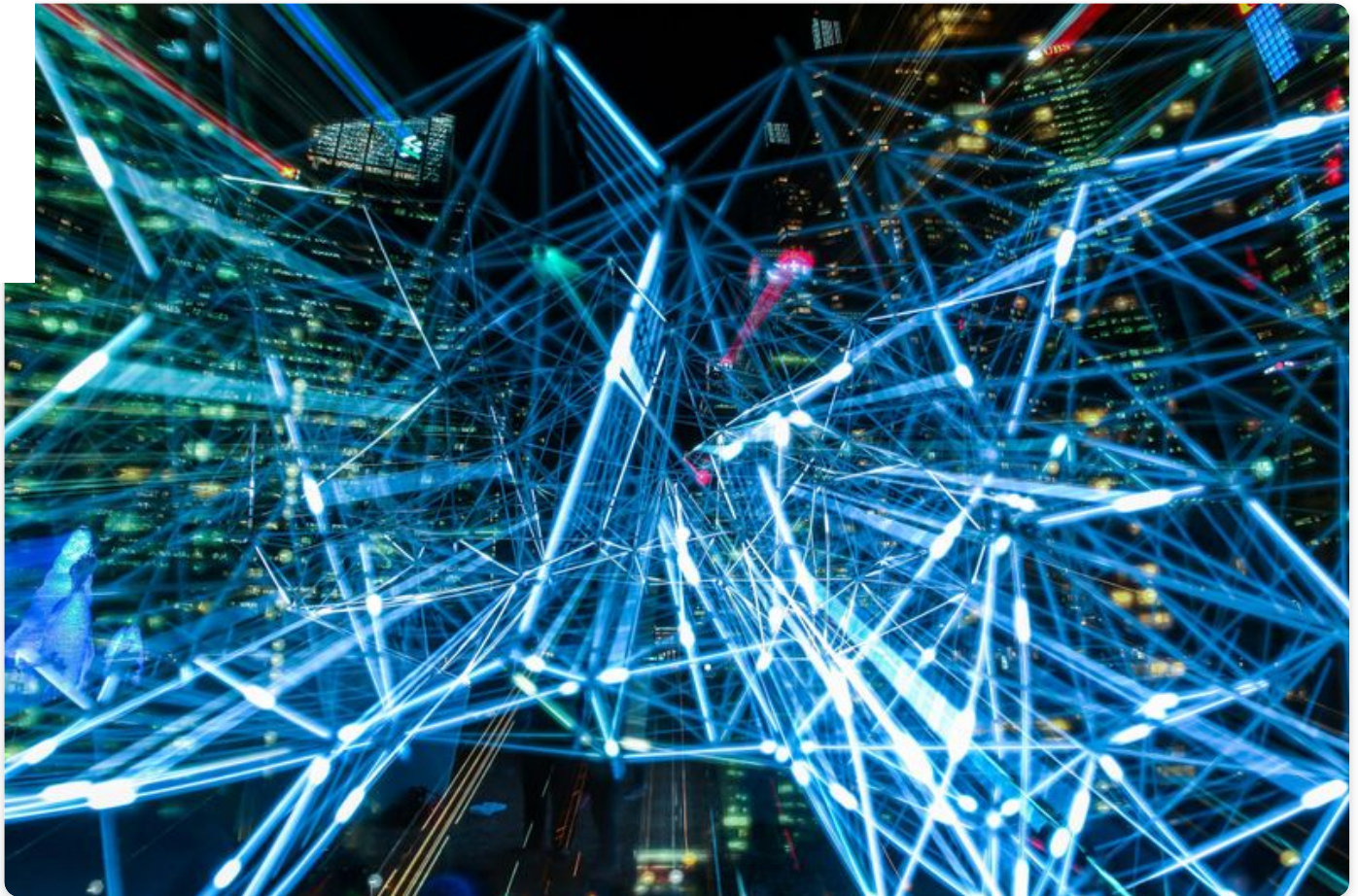
Developers

Docs

Open Source

Write for Us

Free Sign Up



OAuth

Engineering

OAuth 1.0 VS OAuth 2.0

Learn about the differences between OAuth 1.0 and OAuth 2.0 and how OAuth 2.0 is superior to OAuth 1.0



Ti Zhang

May 31, 2019

3 min read

OAuth2 is an authorization delegation protocol that allows one party's accessing an end user's resources stored with another party without sharing any credentials. OAuth2 is often compared with SAML and OpenID Connect as their purposes and uses overlap, however these comparisons often refer to OAuth2 as Auth. This has resulted in some confusion regarding OAuth2 and OAuth1.

[Auth1](#) was published in 2010, and [OAuth2](#) is a complete rewrite of OAuth1 released in 2012. The following section will go over the most significant needs that led to this rewrite, along with the change associated to address them.

Support for non-browser based applications

One of the commonly agreed-upon disadvantages of OAuth1 was the lack of support it offers to non-browser based application clients. OAuth2 has different authorization work flows to address authorization initiated by native application clients. This was one of the main advantages OAuth2 has over OAuth1. However, abuse of the flows in favour of convenience and ease can lead to insecure implementations of OAuth2. When using OAuth2 for mobile, desktop, or single page applications, it is recommended to refer to the IETF paper going over best OAuth2 practices for mobile apps: <https://tools.ietf.org/html/rfc8252>.

Ease of implementation

OAuth1 was often criticized for the barrier it poses to writing a client as each exchange between client, server, and resource server requires a validation of a shared secret. This secret is used to sign the arguments for the authorization request by the client, subsequently the server signs the arguments with the client's key to verify the legitimacy of the client. The arguments need to be passed in the exact order and is often finicky to write. Moreover, dealing with cryptographic signing of the requests in addition to this can be a pain.

Auth2 has delegated this part of the security to transfer over HTTPS. This means while OAuth1 is protocol-independent, OAuth2 requests must be sent over SSL. Since TLS already provides transport-level message privacy and integrity, some question the merit of arguably redundant client-side signing and argument sorting. Others have brought up concerns with completely delegating security to HTTPS, and mention reasons such as yet-undiscovered zero-day TLS vulnerabilities potentially compromising entire systems.

Clear separation of roles

The conceptualization of OAuth2 defines a resource server in addition to an authorization server. This means there is a clear separation of roles between the server that handles the authorization request, and the server that makes access-control decisions based on the response to the authorization request. This separation of concerns allows support for more flexible use cases.

So, OAuth1 or OAuth2?

All of the above points seem to suggest OAuth2 as a superior alternative to OAuth1, and that OAuth1 is obsolete. This is not the case. It is very rare to see a greenfield authorization system using OAuth1, and the only major player still using OAuth1 is Twitter — they call their version OAuth1.0a. However, as far as security and usability is concerned, OAuth1 is still viable and perhaps even more

secure than OAuth2 since it offers additional security on top of TLS-based precautions, and creates barriers in potentially compromising flows. An existing system that uses OAuth1 probably does not need to upgrade to OAuth2. New systems that rely on server-to-server authorization could probably leverage OAuth1 for the additional security as well. On the other hand, use cases that could benefit from a separation of concerns, non-browser support, and ease of client development should go for OAuth2.

OAuth2 has received its own share of criticisms. For example, in 2012 Eran Hammer, one of the original authors of OAuth2, withdrew his name from the specification and wrote [an article](#) calling out its many flaws. However, even in this article he agreed with the usefulness of OAuth2, and that “at the hand of a developer with deep understanding of web security will likely result in a secure implementation”.

OAuth2 is not necessarily more secure than OAuth1, and using OAuth2 does not inherently lead to better security. Many considerations must go into each specific implementation. For starters, the appropriate grant flow must be chosen with care pertaining to the use case; the `redirect_uri` must be validated sufficiently; and measures must be taken to prevent access tokens from ending up in the browser history. For additional security considerations, see this [IETF work in progress draft on OAuth Security Best Current Practice](#).

Related Posts

Javascript tips and tricks to Optimize Performance

Engineering

JavaScript

Hacks

Array

3 Simple Ways to Secure Your Websites/Applications

Engineering

A Bot Protection Overview

Engineering


Captcha

Spam

Secure

IP

Follow LoginRadius

 via feedly

 on twitter

LoginRadius Docs

Implement Authentication in Minutes

[CLICK HERE](#)

Most Popular Tags

Engineering

JavaScript

NodeJs

React

CSS

Security

Go

OAuth

SocialLogin

Authentication

Are your customers safe on your application?

According to Forbes, data breaches exposed 4.1 billion records in the first six months of 2019. If this gets you worried, we've got your back!

LoginRadius protects your customers' identities. We provide world-class security for your customers during login, registration, password setup, and any other data touchpoints, and make sure that their data is safe. We do so through / offering open source SDKs, integrations with over 150 third party applications, pre-designed and customizable login interfaces, and best-in-class data security products such as MFA, RBA, and Advanced Password Policies. The platform is already loved by over 3,000 businesses with a monthly reach of 1.17 billion users worldwide.

SECURE YOUR APPLICATION NOW



Ti Zhang

Software engineer at LoginRadius with a love for good security practices and bad puns.

[View Profile](#)

Login Powered by **Social9**

Add a comment

Add a comment

M ↓ Markdown

Add Comment

Login

Add a comment

Markdown

Add Comment

Powered by **Social9**

Try a Modern Authentication Solution

\$0 / month

FREE SIGN UP

5,000 MAU

1 Web or mobile app

Standard login

3 Social Login Providers

Transactional Email Template

Customizable Login Interfaces



radius®

LoginRadius empowers businesses to deliver a delightful customer experience and win customer trust. Using the LoginRadius Identity Platform, companies can offer a streamlined login process while protecting customer accounts and complying with data privacy regulations.

© Copyright 2021, [LoginRadius Inc.](#)

[Privacy Policy](#)

[Terms](#)

[Security Policy](#)

[Site Map](#)