Scott Bebington u21546216

Name of the scheme/implementation: OAuth

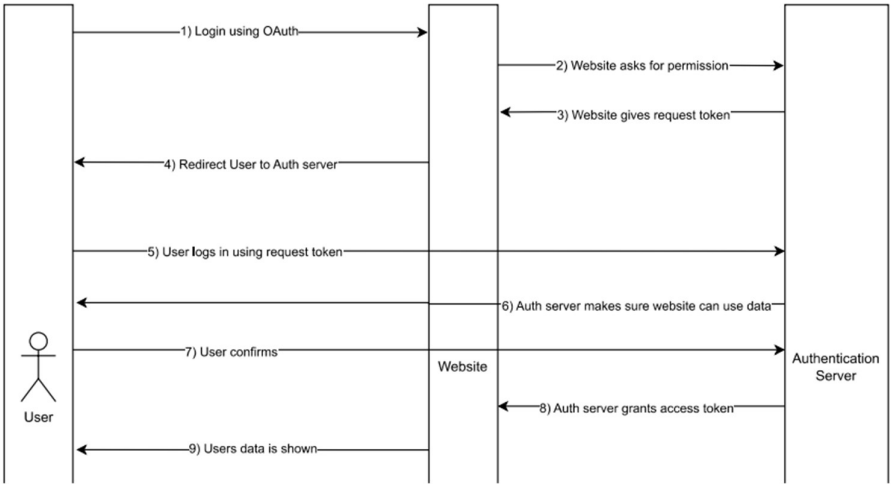
What is used for identification: The authentication server will issue a token to the user to be used when making requests.

What is used for authorization: The authentication server will ask the user for permission for the website to use its data. This will come in the form of an access token.

Brief description of the scheme, how it works:

The user navigates to the website but decides to login using a 3rd party such as google instead of his/her login details. The website goes to google and asks if this user can access google data for this person. Google then sends back a request token to the user and the user is then redirected to google to login using this token, google asks if the website may access the data, in which case the user says yes and is redirected to the website. The website then uses that request token to ask Google for an access token for the data which Google says yes as the user has been verified. The website then uses that access token to retrieve the user's data from Google.

Supporting image:



How can it be attacked: If this website is a phishing website, it may direct a user to a website that looks like google to trick the user into entering their login details. Another vulnerability is a man in the middle attack, a hacker could be listening in on the traffic and may intercept the access token.

What countermeasure(s) could be put in place: Make sure the URL you are being redirected to is valid. Using tokens that have an expiry date and limit the tokens to what they can do could mitigate damage done

References:

https://www.varonis.com/blog/what-is-oauth

https://developers.google.com/identity/protocols/oauth2/javascript-implicit-flow#:~:text=OAuth%202.0%20allows%20users%20to,called%20the%20implicit%20grant%20flow.