Login via mobile app

If you have an app where users need to authenticate, you can improve security by adding federated authentication to your app. You can use OpenID Connect for that. SURFnet offers a code base you can embed in your code. Read on to learn more about adding federated authentication in your app.

Best practices of apps and user authentication

The IETF has published a list of recommended best practices for security and user experience around use of these specifications in native apps. Please read this Ping Identity blog about it:https://www.pingidentity.com/en/company/blog/2017/08/08/single_sign-on_and_ios_11.html .

The Carnegy Mellon CERT also published a blog, https://insights.sei.cmu.edu/cert/2016/08/the-risks-of-google-sign-in-on-ios-devices.html , about what makes a good app authentication.

How adding federated authentication improves security

Offering your customers federated authentication the right way means end-users visually only hand off their password to their home organisations (like an institution), and see a familiar home-organisation login page. Opposed to this are app-developers offering their own in app login page: by doing that, users get more vulnerable to phishing attacks, since they get used to inputting their passwords in all kinds of apps. App-developers offering ‘the right’ way of federated authentication can use this in their sales pitch to prospective customers!

Ways of adding federated authentication in your app

You have a couple of options to do great authentication in your app:

- SURFnet has build a code-base which you can find at https://github.com/SURFnet/nonweb-sso
- Ping Identity blogged about Google’s AppAuth they donated to the OpenID Foundation:https://www.pingidentity.com/en/company/blog/2016/03/10/using_appauth_to_enable_your_apps_with_mobile_sso.html
- GLUU has blogged about Google’s AppAuth initiative: https://www.gluu.org/blog/webviews-are-bad-use-appauth/

But my own in app login page looks far better!

One of the most heard objectives to ‘doing login right’ is: the user-flow/user-experience is worse than when I just offer 2 input fields, one for a userid and another for a password. This is true. But why do you think companies like Google and Facebook, and IETF, use and recommend the ‘right’ way? Because helping the end user stay secure is more important!

More information

We blogged about the SURFnet-SDK: https://blog.surf.nl/en/federated-login-to-native-applications-sdk/

Questions?

If you want more information, please email Raoul.teeuwen@surfnet.nl