An OAuth-based Single Sign-On Solution for Mobile Applications

Alessandro Armando*†, Roberto Carbone*, Silvio Ranise*, and Giada Sciarretta*‡
* Security & Trust, FBK-Irst, Trento, Italia
†DIBRIS, University of Genova, Italia
‡ DISI, University of Trento, Italia
Email: {armando, carbone, ranise, giada.sciarretta}@fbk.eu

In this paper, we focus on the security analysis of native Single Sign-On (SSO) solutions. For native SSO, we refer to the process that enables Client (C) mobile native applications to securely access user profile made available through a remote Identity Provider (IdP) and permitting users to grant or deny access to their user profiles. OAuth 2.0 (OAuth, for short) is one of the most widespread protocols used in the case of mobile native apps. Even if OAuth is an authorization protocol, it is typically used for authentication by assuming that the resource that C wants to access is the user profile. If C can access a user profile, this means that the user has granted the permission. This authorization act is used by C as a proof of the user identity. The problem is that, in the case of native apps, OAuth is not precise enough about how to implement the protocol. There are three possible flows (Authorization Code, Implicit and Resource Owner Password Credentials), but they were originally designed for a browser-based scenario that involves web applications, and are not completely applicable in the mobile case. To illustrate, consider the client authentication process. It is of utmost importance to include the identity in a client request to obtain an access token from the authorization server. This allows for avoiding app impersonation attacks, which make large scale exploit and privacy-leak possible, especially when used in combination with APIs. While in web scenarios, client authentication can be achieved by using a client secret (i.e. a value, known only to the client after completing a registration phase with the authorization server), this is not possible for native mobile apps. Indeed, these shall be considered as public clients and thus unable to keep any value secret; the OAuth standard does not specify how to address this issue. The NAPPS specification tries to solve this gap by adding in the SSO process a new entity that manages authentication and token exchange. However, we could not evaluate the NAPPS solution because it is only a draft. Thus, we have performed a security analysis of social network solutions, focusing on Facebook.

To obtain a precise description of the Facebook native SSO flow, we adopted the following two-step methodology. First, we have analyzed the source code of the Facebook SDK to understand the interaction between a C app and Facebook mobile app inside the smartphone. Second, we have used the Fiddler proxy tool to carefully inspect the HTTP(S) traffic between the Facebook mobile app and the Facebook server. As a consequence, we have noticed that the Facebook native SSO flow is the combination of two OAuth flows (Resource Owner password credential and Implicit), and, similarly to NAPPS, it releases a mobile application that has two purposes: it permits Facebook to authenticate C apps, and it manages user authentication and token release. Unfortunately, Facebook develops both the SDK and the best practices referring to its security requirements and business logic, making this of little or no use in other contexts. For instance, Facebook SDK assumes user authentication to be done via its IdP whereas in the scenario considered in this paper, users can authenticate with any IdP, willing to provide its own native SSO solution. In addition, we have discovered that, even if some implementation choices protect the Facebook native SSO solution from some known attacks, it anyway suffers from attacks such as phishing, user impersonation and client (Facebook app) impersonation. Starting from the Facebook native SSO solution and taking advantage of our security analysis, in this paper we have describe our native SSO solution capable of mitigating the identified vulnerabilities and accommodating any IdP.

This contribution was presented at SECRYPT 2016 (http://www.secrypt.icete.org/). The full paper is available on our website (https://st.fbk.eu/publications/SECRYPT-2016) under the title “Security of Mobile Single Sign-On: a Rational Reconstruction of Facebook Login Solution”.