Cloud app security

Daniel Hedin
Mälaradalen University, Västerås, Sweden

The cloud is becoming increasingly popular as more applications include cloud services, like online storage or connections to social media. From an application provider's point of view the notion of the cloud and its benefits are relatively well understood. Flexible pricing enables easier initial deployment of new services and easier adaptation to varying demand which reduces the risk for over- or under provisioning. From the application user's perspective, though, the notion of cloud application is less well defined. What constitutes a cloud application?

Without trying to define what a cloud application is we can try to identify some properties we would typically associate with cloud applications. The first keyword that can be brought up is availability. There are many aspects of availability, e.g., the key aspect of availability of data. E.g., in the setting of the cloud, we expect the data made on one device to be available on other devices. Examples of this include the data and setting synchronization of Chrome and Firefox, or the account synchronization provided by Apple iCloud or Microsoft OneDrive. The second keyword that can be brought up is collaboration, e.g., collaborative production of content. Examples of this include collaborative editing in Evernote or one of the alternatives from Microsoft, Google or Apple, and user generated content such as the wine reviews of Vivino. The third keyword is simplicity. With simplicity we mean both ease of use but also freedom or ease of installation and maintenance. The latter invites the concept of Software as a Service (SaaS) where the application is delivered to the user in the form of a service rather than an installed product. With the idea of collaboration and applications as services follows the idea of connecting (services of) different cloud applications.

With this architecture, multiuser collaborative applications built from several different cloud services, comes a number of security considerations. In these lectures we will be interested in the confidentiality of user data in the presence of active attackers. A user of a cloud application may allow the application access to sensitive data of one form or another - the most direct example of which is the credentials the user enters to authenticate with the application. The question we ask ourselves is how can we guarantee that sensitive information is handled in a secure way. More precisely, our focus will be on ensuring confidentiality on the client side in the scenario where the cloud application is implemented as a cloud web application. There are a number of reasons for this focus. First, the general area of cloud applications is to vast to serve as a useful basis. Second, web applications are the dominating form of SaaS and embodies many of the security challenges of cloud applications in general, also on the client side. Third, it allows us to easily experiment with practical attacks.

The lectures are built around JSFlow, a security-enhanced JavaScript interpreter employing fine grained tracking-of information flow, and a simplified example cloud application, Hrafn. The key argument we put forward is that the traditional access control is not enough to guarantee information security. In the presence of untrusted users or services even more important is what the application does with the sensitive information after access has been granted. To this end we suggest the use information-flow control to guarantee that the application does not violate the set security policies.

From a practical stand point there are two challenges associated with these lectures. The first is an information-flow challenge where you are introduced to different static variants of enforcement of secure information flow and asked to bypass the enforcement. This exercise highlights the limitations of common forms of enforcement as well as some common mistakes. The second challenge is a code injection challenge. Hrafn is built around one application server, one ad service and one analytics service that all contain implementation flaws. Hence, the challenge is to identify flaws and leverage them to steal the credentials of users that log in to Hrafn.

From a more theoretical standpoint we will look at static an dynamic enforcement of secure information flow with emphasis on the latter. Based on this we will present suggested solutions to the injection attacks and show that JSFlow stops them and why. In addition, we will also contrast the attacks and JSFlow to dynamic taint tracking and show why the latter is not enough in the presence of attackers with code injection capabilities.

At the very end, we leave the focus on the client side to briefly discuss how information-flow control can be leveraged on an application-wide level to provide confidentiality of user data. Online information about the course can be found at `http://www.jsflow.net/coins-2015.html`