Lane Gramling
EECS 700
Reading 10

Summary - LayerCake

This paper concerns itself with studying potential security vulnerabilities emergent from how Android applications handle embedding of third-party applications and UIs, and how cross-application interface embedding may be implemented in Android in a secure manner. To do this, the authors analyze the approach that modern browsers and other existing software take to support secure third-party interface embedding. In the paper, the authors use these archetypes as a guideline for laying out their experimental implementation of secure third-party interface embedding, dubbed LayerCake.

Contributions

Firstly, the article provides an elaboration of the major security vulnerabilities to be tackled in a third-party interface embedding system. Among these, examples of attack strategies made possible by insecure interface embedding are given, including clickjacking, denial-of-service, eavesdropping, and other attacks. This leads to the first major contribution of the paper, which is an outline of their threat model used to account for the main security pitfalls inherent with using an unrestricted child view to embed a third-party interface into a parent activity. The established threat model mainly focuses on isolating aspects of the embedded child from the parent activity. Thus, emphasis is applied on restricting the freedoms of the child to manipulate the display, access components and data of the parent, and origin verification for API calls. The next major contribution of the paper is an elaboration of implementation details on how each of these vulnerabilities are tackled in LayerCake. On the top level, LayerCake exists as a modified version of the Android Framework to include these security measures that satisfy their established threat model. Additionally, LayerCake introduces a new Android view component EmbeddedActivityView, which includes these safety measures to ensure that android applications can safely embed third-party interfaces within the boundaries of their threat model.

Limitations

As expressed in the paper, there are a number of limitations that exist with LayerCake that could complicate apps making use of its capabilities. The first limitation pointed out is that embedded applications that require unavailable third-party software may not properly handle the absence of such software. In their experimental implementation, this was observed to occasionally cause unhandled errors in this case. The authors also point out another issue with their implementation, which is that an embedded application cannot necessarily detect that it is in fact embedded, potentially leading to improper or otherwise undesired performance & usability effects in the application. Another limitation not mentioned by the authors is that their design requires direct modification of the Android framework, meaning that it would be infeasible to deploy on a system using an older version of Android, and may not be possible to distribute at all without releasing an update to the operating system.

Future Work

In this paper, the authors do an adequate exploration of the potential for allowing the possibility for cross-application embedding within an Android activity. There are a number of limitations existent in LayerCake, which leave room for further fleshing out of a more practical implementation that is more friendly to edge cases of the child application's operation. In conjunction, there are also many doors for development opened up by practical support of cross-application interface embedding in Android software which were not previously available in the arsenal of mobile developers.