Securing Embedded User Interfaces: Android and Beyond

Paper Summary

Ruturaj Kiran Vaidya
University of Kansas

1 Summary

Embedded third party widgets are used commonly in websites, as well as in smart phone (such as android) mobile applications, for e.g. twitter tweet, Linkedin share, Google maps, advertisements etc. Paper talks about securing the embedded interfaces or widgets in an android viewpoint. In android apps, These inclusion of third party widgets or interfaces is done by adding their library codes into an app. Android doesn't provide secure isolation between an app and these libraries, as these libraries run in the app's context and android doesn't provide, cross application UI embedding, leading to security vulnerabilities. These third party libraries (malicious child) have a power to replace application's UI elements with the malicious ones. Also, a malicious app (malicious parent) can eavesdrop on child elements and steal the user information. To address these vulnerabilities paper proposes 'layercake', which is their modified version of android, that securely supports application embedding. It does this in three main steps - "first it separates embedded UI into its own process, into its own application"[1]. To achieve this, it introduces a new view called EmbeddedActivityView and also provides parent-child communication. "Second, it separates layout trees of the embedded content and parent content"[1] (separate windows). Third, it handles the additional security concerns, e.g. size conflicts, click-jacking prevention, etc. Authors claim that, 'layercake' handles most of the issues and provides a secure environment.

2 Contribution

The paper contributes by presents 'layercake' (similar to the 'iframes', which are used to secure web browsers), to securely support application embedding. Paper gives "concrete set of criteria and techniques", to support secure interface embedding.

3 Strengths

'Layercake' supports cross-principle APIs, handles sized conflicts, supports click-jacking prevention and preventing ancestor redirection. Also, the load time of the parent activity is unaffected by the existence of the embedded child content.

4 Limitations

It takes 2500 modifications to 50 files and changes in android activity manager and android window manager to implement 'layercake', which I think is a lot (they mentioned in their USENIX video). An important limitation is the performance degradation. With embedding, it takes about 5 times longer to start the app.

5 Future Work

The future work can be done to increase the performance, and decrease the load time. As authors suggested, future work can also be done to handle the case "when an application embeds an activity from another application that is not installed"[1], and the issue of principle identification, i.e. "ease in determining the source of embedded interface"[1].

References

[1] Franziska Roesner and Tadayoshi Kohno *Securing Embedded User Interfaces: Android and Beyond*, USENIX Security, (2013)