CS858 Progress Report : Comparing the security impact of Online generated apps and apps developed using IDEs

Joy Idialu
j2idialu@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

Amirreza Shamsolhodaei
amirreza.shamsolhodaei@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

1 Background

Our project aims at detecting security vulnerabilities, mainly privilege escalation in apps built by Online App Generators. We intend on analyzing the apps and comparing it with apps created using an Android Studio where best practices such as least-privilege principles are followed to be able to infer what features and access controls are expected. This will give an idea on online app generators in terms of security and permission usage and if they are a safe way to create apps by "citizen developers". if this is not the case, then it will negatively affect the already concerning state of security in android apps. We are planning on recreating parts of research done by Oltrogge et al. [1], in smaller scale and developing and generating two categories of apps with different motivations.

2 Building Apps

In order to grasp a better understanding on Android app development and decompilation and analyzing the Android apps, we tried to make a cross development scenario between each other, so each of us will develop a simple hello world app with no permissions, another app with a kind of permission given to it, as well as this generate a simple hello world app (which is not supposed to have any permissions) and another app with different permission request than the developed app. main functionality of our apps will be an SMS sending app which uses SMS permissions, and a location locator app which has access to location permission. based on this one of us will develop an SMS app and generate a location app, and the other one does the exact opposite. As of now, both of us have developed and generated the app without any permissions needed and we have started developing our respective apps.

3 App Decompilation

As app generators provide a single APK file, we need to decompile the APK file in order to get access to source code, different components, and Android manifest for finding the permissions which are used in our generated apps. we have successfully decompiled the simple app which we expected to have no permission. The decompilation process of the apps is as follows:

1. Used apktool to decompile the APK file. The apktool extracted the compressed files such as assets, resources, compiled code. One of the extracted files, the AndroidManifest.XML file was what we used in identifying the permission requirements of the OAG apps.
2. Used dex2jar to convert the APK file into standard class file in .jar format.
3. Used jd-GUI to get access to Java source codes of ".class" files so we can analyze the source code.

4 Manual Analysis

After decompiling the apps, we manually checked different files such as the AndroidManifest.XML file to see what permissions were used in both and discovered that although the apps built using Android studio required no permissions as no protected resource was needed, the generated apps required permissions. For example, in one of the apps, these were the permissions we saw in the manifest file:

```
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
```

We cannot tell what Andromo tends to do with these permissions and we hope to make some more discoveries in line with this as we progess further in the project.

Andromo also included certain third party apps which were not necessary and could be used to gather user information. We observed that:

- Andromo included third party libraries for Ads for example facebook Ads even when it wasn't included while generating one of the apps.
- Andromo also included other third party libraries from Google such as Firebase even when there was no feature in the app that required back-end communication.

5 Current Challenges

The Online App Generator, Andromo used wasn't user friendly and had different interfaces for different users which made the generation of one of the apps so complex and probably corrupted that it couldn't be decompiled to the java source file. we are still investigating this issue. one other issue is the purposeful limitations that andromo has in its free version, only three apps could be built using the free version. we will

try to find a workaround for this. However, if the problems persist, We plan on checking other online app generators to generate the main apps which would use certain sensitive resources and require permissions.
We have not faced any challenges regarding app development as of yet and it has been an interesting learning curve.

6 Pending Work

As mentioned earlier, each of us will build the two apps requiring permissions using Android studio and generating similar apps using an OAG. Firstly, we are going to manually analyze the differences in permission usage and after that, we plan on statically analyzing the apps using WALA. We plan on identifying vulnerabilities through our manual and static analysis . Finally, we plan on discussing our results, highlighting security vulnerabilities in form of privilege escalation which will form the contribution of our project.

References

[1] Marten Oltrogge, Erik Derr, Christian Stransky, Yasemin Acar, Sascha Fahl, Christian Rossow, Giancarlo Pellegrino, Sven Bugiel, and Michael Backes. 2018. The Rise of the Citizen Developer: Assessing the Security Impact of Online App Generators. In *2018 IEEE Symposium on Security and Privacy (SP)*. 634–647. https://doi.org/10.1109/SP.2018.00005