Edinburgh Napier UNIVERSITY

Technical Review

Scott Postlethwaite, Giovanni Paolini
4028026@live.napier.ac.uk, 40276003@live.napier.ac.uk
Edinburgh Napier University - Android Malware Analysis

1 Introduction

The aim of this report is to explain and justify the design decisions for the provided application. This includes the optimization of the application for the ease use for test subjects as well as the client to be able to gather the data effectively. The application consists of a spot the difference game as well as a pdf document for the user to interact with all the while the touch input along with the accelerometer data is recorded. This is to allow the client to analyse this touch data in order to generate patterns and make assumptions of the user. We have optimised the application for this use case by exporting the data via email to the client with a format compatible with multiple types of visualisation software.

2 Design

The application is divided in 7 different classes:

AccelHandler: Handles the accelerometer sensors FileHandler: Handles accesses from and to files. MainActivity: First activity displayed in the application, contains a menu of options. GameScreen: Activity for spot the difference game. ImageAdapter: Custom adapter class for GameScreen. WikiAdapter: Activity for the wikipedia article. LogScreen: Activity which contains all the stored data.

2.1 Main Activity

Main Activity contains a range of buttons that allow the user to navigate through the app. In this class the Accelerometer handler is initialized, and will keep running until the application shut down.

2.2 AccelHandler

AccelHandler class uses the accelerometer sensor to log informations about how the users is holding his phone. It stores x,y and z position of the phone at any time and uses a File Handler to store it.

2.3 FileHandler

FileHandler class is initialized in every activity of the application. Contains the methods to read, write to and reset a file.

2.4 Game Screen

This activity contains the Spot the difference game. It uses a custom ImageAdapter to show a gallery of pictures that can be changed with a swiping motion. Over the ImageAdapter there is a transparent textview, which acts as an overlay, logging the X and Y coordinates of user touches. The class uses FileHandler to store the coordinates.

2.5 ImageAdapter

ImageAdapter class is used to create a picture gallery through the use of an array of drawable images, extending the class PageAdapter.

2.6 WikiScreen

The activity is used to display a wiki article in pdf format, through the use of a library. This part has proven problematic: The pdf viewer does not allow to override the ontouch function, and if it is changed, the user cannot interact with the article. If an overlay is placed over the pdfviewer, the touch function has to be consumed by the overlay to record sweeping motions, and this will also stop the interaction with the pdf, which needs the action that was consumed by the overlay.

2.7 LogScreen

The log screen provides the function to display all the current logs for each one of the other activity, including the accelerometer. It also provides a button to clear all the logs and a button to start an Email Intent.

2.8 Limitations

While the sensory data is collected in full while reading from a pdf, only the first touch is recorded. This is due to compatibility issues with the OnTouch function and the PDFViewer library used. To be more specific this library does not allow to override the OnTouch function, causing the application to crash. As well as this, using a textview as an overlay, the same way as with the spot the difference game, would also not work. The OnTouch function on the textview had to return true in order to log correctly all 3 the user actions : ACTION_DOWN (Finger touches the screen, first coordinate point), ACTION_MOVE(Finger still down, moving on the screen) and ACTION_UP(Finger lifted from screen, last coordinate point). Specifically, MOVE and UP need to consume the events to be logged. But if the event is consumed, the touch cannot pass through the overlay and interact with the pdf, so the OnTouch function must return false. So there are two cases, if the function returns false, user can interact with the pdf but only the first touch of a motion is logged, if the function returns true, every touch event is logged but the user cannot interact with the pdf. We currently have no solution to this. Possible solutions could include utilizing another library, using interceptors or dispatchers to route

the touch event from the child view(the overlay), to the parent view(the PDF viewer). So far none of these have been successful therefore further research or another approach is needed.

2.9 Fitness For Purpose

Our final product addresses all of the clients must have features while building upon this to incorporate the majority of the features that the client stated the application should have. We have developed an application which records the users touch data in app while the user plays a simple spot the difference game. This data is then output to a .txt file which allows the client to pass it into the visualization software of his choice. When compared to the initial specification we can see that while our application records all that the client requires there are several omissions that would have made the clients life easier.

First of all the specification outlined that the application should have a level of transparency to allow the touch data to be recorded in the background. Although this feature has been in development since the initial weeks of the project, incompatibility with the latest android versions as well as an overall lack of up to date documentation meant that we would be unable to complete this in a way that is suitable to the clients needs. While there are workarounds in the form of root permissions we decided that as the clients test subjects would be downloading this application to their own phones it would be unrealistic to assume that they all had rooted devices as well as unfair to require them to root their device for our application.

The final omission to the application was a way to visualize the data. We made the decision to omit this and focus our efforts on both the literature review and the sensory data due to our unfamiliarity with the clients area of study. As this was somewhat of an afterthought from the client and was said to be a finishing touch to the project provided we had the time we decided it would be more beneficial to have the must have elements of the application as well as the documentation rather than implementing complex visualizations.

Overall our project has delivered upon the clients requests while making the necessary omissions to ensure our product was polished and usable within the timescale provided.

2.10 Comparison to Other Applications

Perhaps the best comparisons that can be made to our application is to the 2016 cloak and dagger research. While our applications may look completely different, the key functions are the same: to silently capture users touch data. While our application handles this through the use of touch listeners over a transparent text view While in the application, the cloak and dagger attack makes use of android 7 and priors SYSTEM_ALERT_WINDOW ("draw on top") and BIND_ACCESSIBILITY_SERVICE ("a11y") services. This allows the application to appear transparent above other applications as well as enabling the app to record touch and therefore sensory data while the phone is in use. This security flaw has since been patched in android 8.0 and above. It is for this reason that we decided not to follow a similar procedure and to instead develop in app recording. While the same effect can be achieved through the use of root permissions this would require our users to either root their own handsets or for our client to provide a rooted handset to test on. It is for this reason that we decided to find a solution for gathering the touch input for android 8.0 rather than using a rooted solution or limiting the handset pool to version 7.1 and before. This way, to distribute we simply need to send the test subjects the APK file for the application and having the subjects accept permissions making it accessible to a larger degree of test subjects.

2.11 Potential Enhancements

As the clientâĂŹs intention for this program is to analyze touch data so that it can be used for continuous authentication systems a variety of tasks to interact with would benefit the research by allowing for a simulation of more real world scenarios to analyze.

The first of these that we would implement would be a media player for audio and video. This would allow us to study one of the main use cases for most young people, for audio visual media consumption. Ideally this service would have a layout similar to popular apps such as YouTube, Netflix or even a separate service to mimic spotify. This would ultimately benefit the research by allowing the analysis of another real world example.

Another potential use case to mimic in app would be a news feed. This popular form of media consumption appears to have taken over the social media market. This would greatly benefit the final research as social media is one of the primary uses of young peoples mobile phones and reflecting this in the application would allow the client to greater distinguish between different groups of users and could ultimately allow for greater accuracy in the proposed authentication systems.

The final potential use case that could be mimicked would be instant messaging. As this is the primary use for many handsets it would make an interesting point of research. Firstly you could distinguish whether the phone is being used one or two handed through the time discrepancy in key strokes as well as the inaccuracy in typing. Many users navigate their messaging apps differently as well as some prefer to compose a message and send to their intended contact where others will search for the contacts chat log and enter from there. This would be in no way ideal though as we would not be able to access their contacts or social media accounts to populate the service with real chats. It would instead be a simulation with some demo chats to interact with. This may alter the ways that users interact and therefore may hinder the research rather than help it.

By far the greatest enhancement that could be made to the application is for the touch logging to happen in the background. This is a limitation of the android operating system that could be exploited in android version 7 and earlier however as the test subjects will likely be using devices version 8 and above we decided to focus our efforts on compatible features and functions. Through research and experimentation we found that there were no known ways to exploit the operating system to allow us to record the touch data in the background. One way around this that we discovered was through the use of root permissions. We initially omitted this as this would limit testing to exclusively rooted handsets

however, if a research handset were to be used this solution would allow for a greater pool of data as the data gathered would be system wide as opposed to in app.

Following on from the prior point another potential enhancement would be to separate the data based upon the activity. This is extremely important for touch analysis as the context of the data has the potential to alter the outcome of the research. For example, users interact completely different with video games than they do with social media. This separation would allow each aspect of mobile interaction to be individually analyzed in order to better the clients understanding of each individuals differences.

3 Conclusion

When compared to other applications of the type, mainly the Cloak and Dagger research project, our application lacks in many of the key functions, mainly a degree of transparency, however when compared to the specification as well as the intended purpose we see that the application delivers on the key functionality while remaining as accessible as possible. While this should aid the clients research project it leaves a lot of potential use cases on the shelf. In order for the application to be completely optimised for the clients needs without adding transparency more use cases would be needed however these have not been implemented due to lack of time, knowledge and their absence in the specification. Despite this the application remains a useful tool to gather touch data while remaining easy to distribute as well as easy to use.