Mark V. Lese

[lesemv01@students.ipfw.edu](mailto:lesemv01@students.ipfw.edu)

CS59000-09

Dr. Elish

# Progress Report #1

February 2, 2016

So far, I have read some of the research papers on colluding Android. These papers have given guidance to what is needed to create colluding application capable of malicious activity. Since it has been some times since I have developed Android applications, I have been familiarizing myself with Android Studio. Since these applications need to communicate via messaging, shared files, or some other means, I have been prototyping to learn the ins and outs of these parts.

## General Requirements

This semester’s project is to create at least two colluding Android applications that exhibit malicious activity when working together unbeknownst to the operator. Each application shall have at least two components. Each application shall have allocated permissions suitable for the application that masks the malicious intent. The allocated permission for the malicious intent shall be different in the applications. The malicious activity shall be a result of a user action, as malicious activity due to user activity is more difficult to detect than a background process activity. I have been able to send a message from one application to another via messaging.

## Proposed Applications Synopses

### Application #1

One application utilizes a NASA web service from their API catalog. The application requires internet permission, a potential port for malicious activity. The application displays data from the web service as a result of user request. This request triggers a service action from a secondary component to retrieve data from another application. This data is then sent to some web service hosted by **leseonline.net** without the user’s knowledge.

### Application #2

The second application is basketball statistics tracker. This application allows a user to enter statistics for a single game. The user may share the results to an email address in the user’s contact list. This requires the application to have to the Contact List permission. This application exposes an external interface to allow the first application to retrieve the contact list. I am not sure at this time how a user action in this application triggers the contact list exposure.

See the block diagrams on the following pages.

**Simplified Block Diagram**

