CSC714 - Project Proposal

Fall 2011

An Application Profiler for Android

http://www4.ncsu.edu/~arezaei2/CSC714/

Arash Rezaei
arezaei2@ncsu.edu

Gopikannan Venugopalsamy
gvenugo@ncsu.edu

Objective

Analysis of monitoring data gathered from application profiling in Android, for the purpose of finding behavior patterns to distinguish malicious applications.

Problem Statement

With the advent of smart phones, mobile computing has been revolutionized and many new opportunities have been introduced into the world of research. A challenge with today's smart phones is their security issue with regards to third party applications [1]. The way a given malicious application uses the phone resources like CPU, battery, network and private data is different from the others. The Android market is getting to a considerably large share and screening that for malicious applications becoming so hard, if not impossible. In addition, the platform providers, hardware manufacturers and application markets are not under the same hood. These pose a unique security challenge on the Android based smart phone area.

In this project, we propose to aid in identification of malicious application and in attack forensics after identifying that a malicious application is running in the system. This can be achieved by implementing a profiling application which gathers the formation related to the usage of system resources like computing, battery, network and other sensing resources/information related to each application. We believe that this data may help to identify behavior patterns which in turn provide assist to differentiate the applications that consume suspicious level/patterns of resources. Finally, in case of finding an application with suspicious usage by services and activities, appropriate action like warning the user or flagging the application can be taken.

Challenges/ Design issues

- Potential monitoring information includes System calls, performance counters, accessed files (R or W), new files, and deleted files, resources like CPU, battery, network, and sensor usage.
- Some of the resources might be hard to monitor, not to be visible by monitoring application/

- The method for analyzing the collected information
- How to send the fairly large amount of data in a secure way while it does not use the whole network bandwidth of the device

Milestone:

1. **Install Eclipse + SDK (both - due OCT 27)**
   During this phase we need to get familiar with development environment and install the required software on our computers.
2. **Background reading - profiler applications (both - due OCT 30)**
   In this phase we need to get information about the other works that has been done in this area, and decide which information we are going to collect about a given application
3. **Find the classes to get the monitoring data(due NOV 5)**
   In this phase, we specifically look into Android to find the places where the monitoring data can be collected.
   - Daemon, monitoring: Network, files (creation/deletion/change) **(Arash)**
   - Monitoring: CPU, Battery, Sensors **(Gopi)**
4. **Design the profiler(both - due NOV 10)**
   During this phase we design the profiler application.
5. **Profiler Implementation (due NOV 23)**
   During this phase we design the profiler application.
   - Daemon, monitoring : Network, files (creation/deletion/change) **(Arash)**
   - Monitoring: CPU, Battery, Sensors **(Gopi)**
6. **Analysis of gathered data (both - due NOV 25)**
   In this phase we analyze the gathered data.
7. **Final project report (both - due NOV 29)**

References:

[1] P. Gilbert, B. G. Chun, L. Cox, and J. Jung. Automating privacy testing of smartphone applications. Technical Report CS-2011-02, Duke University, 2011.

[2] Android developer reference (http://developer.android.com/guide/topics/fundamentals.html)

[3] Android 2.1 Platform | Android Developers http://developer.android.com/sdk/android-2.1.html