Combining Static and Dynamic Permission Analysis for Android

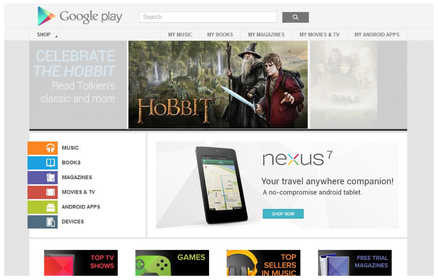
William May | Dr. Paolina Centonze
Iona College | Computer Science | April 20, 2013

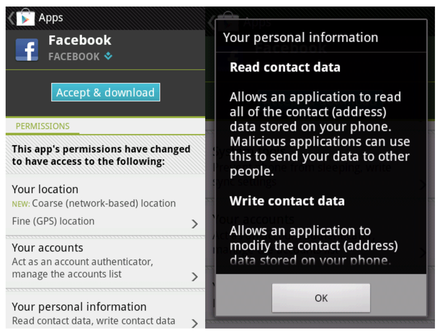
Abstract

As the world becomes increasingly reliant on mobile technologies, so has the importance of research in the area of mobile security in order to protect users and their data. My research is focused on the Android permissions-based security model and enhancing the effectiveness of Android application permission analysis.

Android provides a permission-based security model in which access to privileged system resources is protected through security mechanisms known as Permissions. By default, an Android application does not have any privileges, but applications can request additional privileges through the use of these permissions.

I present a novel method in which an Android application can be analyzed to determine the precise set of permissions an application needs to run properly and more securely on a mobile Android device. I present the tool, ACE4Android (Access-Control Explorer for Android), a Dynamic Analysis tool which, when combined with a Static Analysis tool, will allow application end users to know what they are installing when they download an application, and will allow application developers to improve the runnability and security of their applications.





Android Permissions

Prior to installation, an Android application must explicitly request permissions before it can be installed on an Android device. A list of the permissions the application is requesting is presented to the user, and the user must choose either to install, accepting those permissions, or cancel, declining those permissions. However, it is not necessarily transparent to the user whether an application actually needs all of the permissions (system resources) that it requests.

Underprivilege - If an application is not requesting enough permissions, then those features that require those missing permissions will not function properly and the application will generate an error which may cause the application to crash.

Overprivilege - If an application is requesting too many permissions, then the application violates the principle of least privilege and may compromise the underlying operating system which may cause the device to become vulnerable to a malicious attack.

Therefore, one of the challenges to this permission-based approach is ensuring that an application is requesting the right set of permissions. However, the user cannot modify the applications permissions; it is up to the application developer to select the appropriate set of permissions for their application.

Methodology

1. Static analysis
   Pros:
   - Theoretically sound (no false negatives)
     (modulo native methods, reflection, and callbacks)
   - No need for test cases
   - No need for configuration
   - Safe

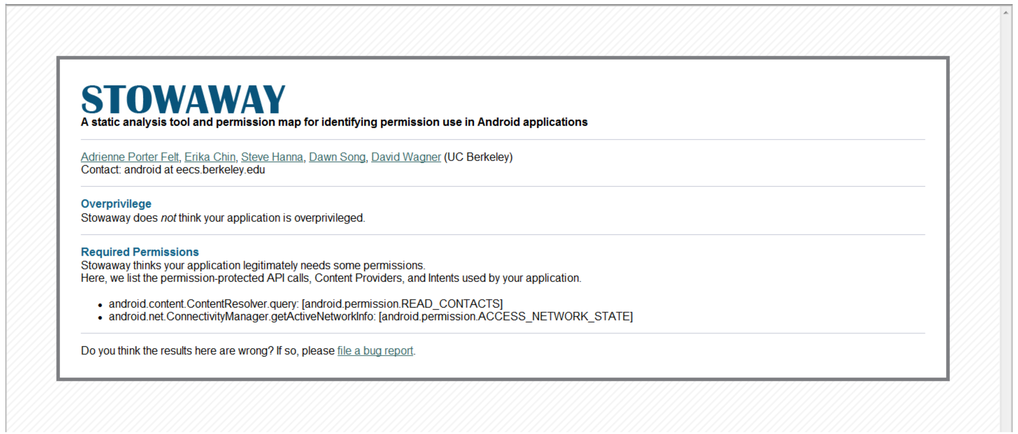
   Cons:
   - Unsound in practice (false negatives)
   - Conservative (false positives)

2. Dynamic analysis
   Pros:
   - Complete (no false positives)

   Cons:
   - Unsound (false negatives)
   - Need for test cases
   - Need for configuration
   - Unsafe (program under analysis can compromise the integrity or confidentiality of the system)

3. Combining Static and Dynamic Analysis
   Pros:
   - Complete (no false positives)
   - Theoretically sound (no false negatives)
     (modulo native methods, reflection, and callbacks)
   - No need for test cases
   - No need for configuration
   - Safe

   Cons:
   - (N/A)



ACE4Android