MOSES: Supporting and Enforcing Security Profiles on Smartphones

ABSTRACT:

Smartphones are very effective tools for increasing the productivity of business users. With their increasing computational power and storage capacity, smartphones allow end users to perform several tasks and be always updated while on the move. Companies are willing to support employee-owned smartphones because of the increase in productivity of their employees. However, security concerns about data sharing, leakage and loss have hindered the adoption of smartphones for corporate use. In this paper we present MOSES, a policy-based framework for enforcing software isolation of applications and data on the Android platform. In MOSES, it is possible to define distinct Security Profiles within a single smartphone. Each security profile is associated with a set of policies that control the access to applications and data. Profiles are not predefined or hardcoded, they can be specified and applied at any time. One of the main characteristics of MOSES is the dynamic switching from one security profile to another. We run a thorough set of experiments using our full implementation of MOSES. The results of the experiments confirm the feasibility of our proposal.
EXISTING SYSTEM:

Smartphones allow end users to perform several tasks while being on the move. As a consequence, end users require their personal smartphones to be connected to their work IT infrastructure. More and more companies nowadays provide mobile versions of their desktop applications. The previous version of MOSES completely relies on Taintdroid to split data between different profiles. Data separation occurred using user-defined policies, which restricted the flow of information between different profiles.

PROPOSED SYSTEM:

There are a lot of solutions proposed to improve the security of Android. We consider the ones that are more related to our system. In Android, at installation time users grant applications the permissions requested in the manifest file. Android supports an all-or-nothing approach, meaning that the user has to either grant all the permissions specified in the manifest or abort the installation of the application. Moreover, a permission cannot be revoked at runtime. To circumvent this coarse-grained approach, several solutions have been proposed.

HARDWARE CONFIGURATION:

- Processor - Pentium -IV
- Speed - 1.1 Ghz
- RAM - 256 MB(min)
IEEE Projects 100% WORKING CODE + DOCUMENTATION + EXPLANATION – BEST PRICE
LOW PRICE GUARANTEED

- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

SOFTWARE CONFIGURATION:

- Operating System : Windows XP
- Programming Language : JAVA
- Java Version : JDK 1.6 & above