SPE: Security and Privacy Enhancement Framework for Mobile Devices

Abstract

A security and privacy enhancement (SPE) framework for unmodified mobile operating systems. SPE introduces a new layer between the application and the operating system and does not require a device be jail broken or utilize a custom operating system. We utilize an existing ontology designed for enforcing security and privacy policies on mobile devices to build a policy that is customizable. Based on this policy, SPE provides enhancements to native controls that currently exist on the platform for privacy and security sensitive components. SPE allows access to these components in a way that allows the framework to ensure the application is truthful in its declared intent and ensure that the user’s policy is enforced. In our evaluation we verify the correctness of the framework and the computing impact on the device. Additionally, we discovered security and privacy issues in several open source applications by utilizing the SPE Framework. From our findings, if SPE is adopted by mobile operating systems producers, it would provide consumers and businesses the additional privacy and security controls they demand and allow users to be more aware of security and privacy issues with applications on their devices.

INTRODUCTION

Mobile computing devices are quickly becoming the platform of choice for consumers and businesses. Given that mobile devices started to outsell PCs in 2011 and mobile applications are freely available in marketplaces, consumers are more likely to focus on mobile devices as their primary personal computing platforms. Additionally, users perform many of the same tasks that were performed
previously with traditional computers on their mobile devices. Most devices are equipped with numerous sensors such as cameras, microphones, GPS, accelerometers, and gyroscopes where users can share data about their environment or habits quickly, but also unknowingly. Here are a few examples of such unintentional sharing: (a) Facebook leaked the phone number from a mobile device before the user logged into the application (b) Angry Birds collected user data, which was found to be used by the NSA to profile users (c) out of 25,976 Android applications, 969 applications leaked location data and 347 recorded audio without the user’s permission (d) Path was found to geotag photos even after a user disabled location services, and sent user’s privacy data unknowingly by uploading a user’s entire address book. Even when a user permits an application to access data on the device, the user is not aware of what else the data is being used for, how often it is being accessed, and with whom it is being shared. There is no way to confirm that the application is truthful in how it states the information will be used.

EXISTING SYSTEM

The rapid growth in the mobile device ecosystem demands viable solutions to security and privacy concerns. Even though mobile devices are becoming more powerful, there still exist constraints on computing power, memory capacity, and a virtual endless supply of energy that traditional computing platforms offer today. These constraints limit mobile devices from performing computationally expensive operations such as pattern-based intrusion detection or fuzzy checking of privacy leakage. Even if computing power on mobile devices were to increase, the effect on the device’s battery would be unacceptable for a user. Additionally, the user experience may be affected if more computationally expensive operations are being executed while the user is interacting with the device. Recent research in this area has introduced novel methods for providing additional security and privacy.
controls. However most of these methods require a modification to the operating system or the device to be jailbroken

**DRAWBACK OF EXISTING SYSTEM**

- Privacy data is leaked immediately.
- Accessing information without user permission.
- Path are easily found, when the location services are disabled.

**PROPOSED SYSTEM**

In proposed system, we have presented the Security and Privacy Enhanced (SPE) framework. We described the policy model it utilizes, the core design of the framework, and details on an implementation that allows a consumer or business to effectively ensure that security and privacy policies are enforced. Additionally, we proposed a novel approach that uses intents to describe to the user how the application will use their data and enforce these intents. Compared to recent research that has focused on modifying open mobile operating systems or jailbreaking closed-source operating systems like iOS, the SPE Framework takes a different approach. While the SPE Framework does require modification to the application, it does not require modification to the OS or for a device to be jailbroken or rooted. We believe this is a more sustainable approach as OS updates do not impact the SPE Framework unless there are significant API changes. Frequent updates to mobile operating systems have led to fragmentation, with modifications to Android by both carriers and device manufacturers. Additionally, a consumer does not need to compromise the built-in security of their device by jail breaking or rooting the device; with SPE they add another layer of protection. Lastly with SPE a consumer can use a stock device with a stock operating system. Based on the results of our evaluation, SPE is highly effective and prevents several
Do Your Projects With Domain Experts...

privacy and security concerns from several iOS applications. In the near future we plan on releasing the SPE Framework, SPE Conversion Assistant, and SPE Policy application as open source projects. From this, an external entity can be created for developers to retrieve the SPE Framework to incorporate within their application or the framework could be tied into the workflow for application submission.

ADVANTAGE OF PROPOSED SYSTEM

- Highly effectiveness of ensure the security and privacy.
- In OS updation, customer need not wait a long time.

Hardware Requirement

System: Pentium IV 2.4 GHz.
Hard Disk: 40 GB.
Floppy Drive: 44 Mb.
Monitor: 15 VGA Colour.
Mouse: Logitech
Ram: 512 Mb.
MOBILE: ANDROID

Software Requirement

Operating system: Windows 7.
Coding Language: Java 1.7
Tool Kit: Android 2.3 ABOVE
IDE: Eclipse