Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 87 (2016) 210 – 214

Procedia
Computer Science

4th International Conference on Recent Trends in Computer Science & Engineering

PLANT:Permission Leakage AvoidaNce Through Filteration

M.Kireet[a], Dr.Meda Sreenivasa Rao[b]

[a]Lecturer,Dept of CSE, JNTUCEH, Hyderabad,India

[b]Professor, Dept of CSE, JNTUSIT, Hyderabad, India

Abstract

The instantaneous growth of mobile apps in mobile markets elevating the concerns on the protection of user's sensitive data in Smartphone's. The Majority of apps in all mobile markets are gathering the data which is insignificant to the app functionality. The mobile app collects the data of the user either users' sensitive data or device data by using orthodox permission control policy.This Permission policy in apps takes acceptance of user for installing an app and user has only one option to accept all permissions to continue the installation of app. In this work we proposed a framework PLANT which is Permission Leakage AvoidaNce Through Filteration for identifying data leakage done by mobile apps by using permissions .Finally framework gives the knowledge to the user to deny if there are any irrelevant permissions required by the app based on its functionality

Keywords: PLANT; Data Leakage; Filteration

1. Introduction

The expansion of Smartphone's throughout the world has extended the use of mobile apps.Almost 13% of today's e-commerce transactions are done using mobiles[3] and 50% of the online orders are placed by mobile apps. The number of apps abruptly increasing in the app stores. At present 2.6 billion people are using Smartphone's and by 2020 the Smartphone's usage may rise to 6.1 billion[4].Till 2015 more than 100 billion apps were downloaded from different app stores[4].Due to this sudden rise in usage of apps in Smartphone's several concerns raised on the privacy of user data in Smartphone's. By analysis [3] 98% of the mobile apps are vulnerable to security attacks which are the apps of top 50 e-commerce companies in India. Most of the apps follow orthodox security policy which controls access based on the permissions. As per this Permission based control policy it's mandatory for the user to accept all the permissions for installing an app. This mandatory acceptance of each and every permission by the user for an app made a good platform for the attackers to extract sensitive user mobile information. More than 50% of the apps require the permissions which are irrelevant to their actual functionality.

Peer-review under responsibility of the Organizing Committee of ICRTCSE 2016
doi:10.1016/j.procs.2016.05.150

2. Related Work

Most of the apps from different app stores have their own security policies as a part of security provision to the mobile users. For resource management Apple ios[5], Blackberry[6], Windows Phone[7], Android[8], use Mandatory Access Control(MAC) mechanism. The static permission based model is used by Android by which user acceptance of all requested permission is required. There are several limitations on this permission based model as a result most of the mobile user data is released as data is released or leaked by using permissions we call this as permission leakage. There are various types of Permission leakage attacks [1][9] like collusion attacks[9],Confused deputy attack and Intent spoofing.There are methods and tools available for permission leakage which perform detection and avoidance by static and dynamic analysis.Some of the detection methods are as follows.

TaintDroid[8] is one of the tool which provides analysis of the app by tracking the flow of sensitive information and alerts the user about the behaviour of an app if any of the sensitive user data is leaked by the app. AppFence[9] an application service provider tracks, detects and gives responses without any influence of legitimate users. MockDroid[10] known as modified version of Android Operating systems mocks application utilization to the resource. Mockdroid manily fakes the program data in order not to reveal the user contents which forms a protection. Fire-Droid[11] a policy –based framework utilizes rendering system calls to obligating security policies.

To intensify the basic conventional security policy of Permission control model used in mobile apps, we proposed a framework which gives the user an option to ignore the unnecessary permissions required by an app that would control the sensitive mobile user data leakage. At the beginning we took the datasets from statista portal [4] and Pew Research centre [6]. Atmost there are 1,041,336 apps in pew data set[4] and 235 permissions are observed , 70 allow access to various user information, 165 permissions allow access to device .Then We categorized the apps into 20 different categories taking the apps data, statistics from statista portal[2].Then we calculated the pertinent permissions which require user info for all the category of the apps i.e., for example if app is a games app by categorization ,which are the pertinent permissions from 70 identified user info permissions required to that app are mapped. Based on the datasets for each category of apps we have obliterated the permissions which have less utilization or usage.For all of the 20 categories of apps the required no of permissions for each app and their pertinent permissions based on their utilization or usage are calculated and maintained. To get more sophistication we used RecDroid[11] for the recommendation from experts on permission granting system. By using RecDroid with the final recommended permissions we run the app in probation mode to reduce the risks if there are any false decisions.Inorder to reduce the main permission leakage attack of collusion attack,we have used android apk decompiler[14] and extracted the Android manifest XML file.By using the Android XML manifest file we compared the intent filter components,SharedUid,risk level.Based on some rules we developed by rule based classification to check the