International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) ISSN: 2395-5325
Volume 3, Special Issue 2 (March '2017)
Proceeding of - National Conference on Recent Trends in Information Technology and Communication,
Conference held at Velammal Engineering College,Chennai,India.

ANDROSF: ANDROID SECURITY FRAMEWORK

S.Revathi,
Assistant Professor,
Department of Computer Science and Engineering
Velammal Engineering College,
Chennai,India.

M.Vignesh,
UG Scholar,
Department of Computer Science and Engineering
Velammal Engineering College,
Chennai,India.

S.J.Vivek Senthil,
UG Scholar,
Department of Computer Science and Engineering
Velammal Engineering College,
.Chennai,India.

R.Vijaykarthick,
UG Scholar,
Department of Computer Science and Engineering
Velammal Engineering College,
Chennai,India.

Abstract: Thousands of new apps hit the market each week. At the same time, thousands of hackers work hard to tap into these apps to try to steal user information like credentials, personal data, and cardholder data. That means app developers have to be vigilant about security to protect users. An app developer, need to know how to avoid the OWASP top ten vulnerabilities most widely affecting users today. It takes a lot of time to develop from the ground up, but there's no need to do so when so much free code exists to build on. Some hackers create code in the hopes that app developers pick it up to use in their apps. If the application developer doesn't ensure that their app is secure, they put all of the app's users at risk. So the application developer should never rush to release an app before properly testing it. In this paper an android security framework is proposed and designed to help the developer to test how secure their applications. AndroSF test for the vulnerabilities they find after reverse engineering the source code to understand the binary. Android Security framework is an intelligent, mobile application automated pen-testing framework capable of performing static, dynamic analysis.

Keywords: Security Framework, Android threats

I.INTRODUCTION

Mobile users demand instant anytime/anywhere access, uncompromised convenience, and intuitive functionality on all devices. At the same time, enterprises must prevent confidential customer information from getting into the hands of malicious adversaries who view the mobile environment as an irresistible target of opportunity. Hackers regularly exploit mobile app vulnerabilities and steal sensitive data. Enhancing **mobile app security** is a critical element of an effective fraud prevention strategy. This papers deals the security issues in android application .Main goal is to classify mobile security risks and provide solutions to control and reduce their impact. AndroSF report bugs based on the OWASP(Open Web Application Security Project) top 10 mobile risk. Fig 1 Lists the top 10 mobile risk in android application.

M1: Improper Platform Usage

M2: Insecure Data Storage

M3: Insecure Communication

M4: Insecure Authentication

M5: Insufficient Cryptography

M6: Insecure Authorization

M7: Client Code Quality

M8: Code Tampering

M9: Reverse Engineering

M10: Extraneous Functionality

Figure 1: Top 10 Mobile Application Risks in OWASP

More attacks started happening in mobile application, is because development is focused on features not on **security.** A weakness in one can lead to exploitation of another. This creates a space to develop a framework which deals with threats.

II. LITERATURE SURVEY

Android is a Linux kernel mobile platform that has been popular throughout its existence on a huge variety of devices, especially mobile smart phones. Most organizations, ranging from banking to telecom companies, have also come up with their apps for Android. Just like generic web applications, these mobile applications need a pen-test exercise as a part of their SDLC life cycle.

This market is projected to reach a huge size by the end of 2017 with the growing demand for high end smart phone applications. Also, people generally rely too much on their Android devices, so compromising them might lead to the loss of a good amount of critical data, including passwords, mails, etc. Therefore, security testing of the applications carrying sensitive user data is very important. This series is a solution for those who want to take a deep dive into mobile application security testing, as this project focuses on the approach for an automated pen-testing Android-based mobile applications. It also provides an introduction to the tool set available for the Android platform. During the whole series, we will try to understand the complete process of mobile application testing in a very comprehensive manner.

Android platform basically needs to be secure at two levels, i.e., the application level and the device level. For application level security, we need to uncover the bugs in applications that are going to be installed on the device. For this, we look out for server-side as well as client-side security issues in the application. The existing system, address the problem of automatic testing of mobile applications developed for the Google Android platform, and presents a technique for rapid crash testing and regression testing of Android applications. The technique is based on a crawler that automatically builds a model of the application GUI and obtains test cases ththe at can be automatically executed. The technique is supported by a tool for both crawling the application and generating the test cases.

In the proposed system, an android security framework designed to helps the developer to test how secure their applications. AndroSF tests for the vulnerabilities they find after reverse engineering the source code to understand the binary. Android Security framework is an intelligent, mobile application automated pen-testing framework capable of performing static, dynamic analysis.

III. SYSTEM ARCHITECTURE

AndroSF is the framework for intimating the security issues in the mobile application. The original source code is first converted into binary code then it is compared with the standard mobile security issues like authentication problem, insecure cryptography problem. Finally the bugs are intimated to the developer along with the severity of the bugs. Fig 2 shows the architecture of AndroSF. Extracting the information of mobile application means fetching the application name, version ,package name, platform,Keys like MD5,SHA1 etc.

These information gets stored in the MYSQL Database, Analyzing phase compares the actual application values with the standard vulnerability database values. If any matches found then it report the bugs along with the severity. Severity of the bugs classified as Dangerous, Normal, Info, Warning, and Notice based on the affect factor. Affect factor nothing but the ratio between percentage of original value modified and the total values. If the affect factor exceed the threshold value 0.5 then it is intimated as dangerous.

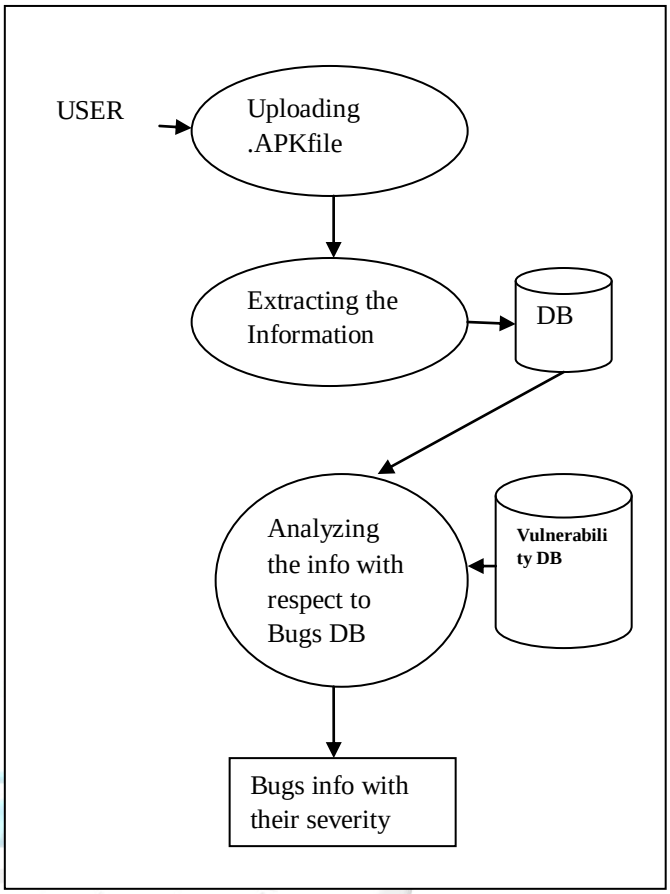

Figure 2:A security checking framework for android APP

IV. EXPERIMENTS AND RESULTS

The purpose of testing is to discover errors.

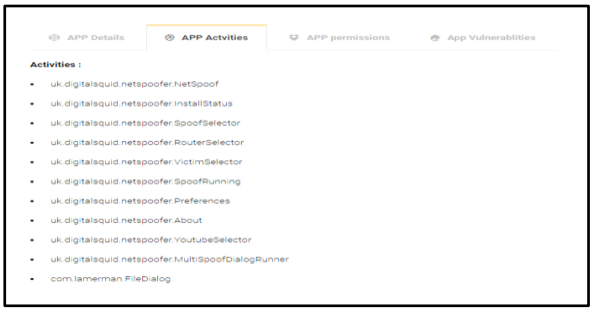

Figure 3: Extracting the information of mobile application from .apk file

Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

There are various types of test. Each test type addresses a specific testing requirement. AndroSF is created using HTML as front end and it uses PHP coding to extract the information in .apk files. MySQL is used for the database purpose. Various android apps are taken for the testing purporse. Fig 3 shows the fetched information from the .apk file.The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

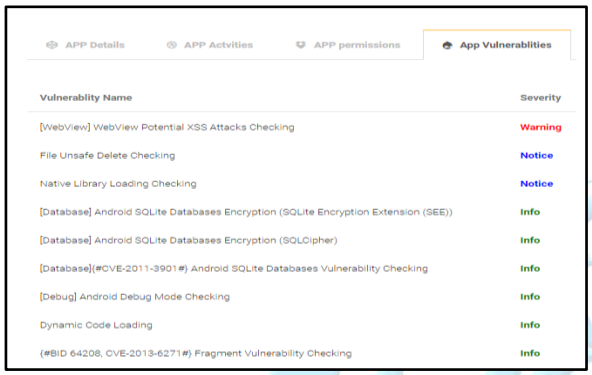

Figure 4 shows the activities list of the given mobile application

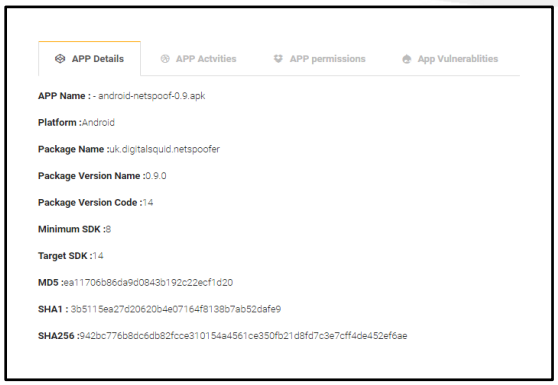

Figure 5: Application Permission Extraction

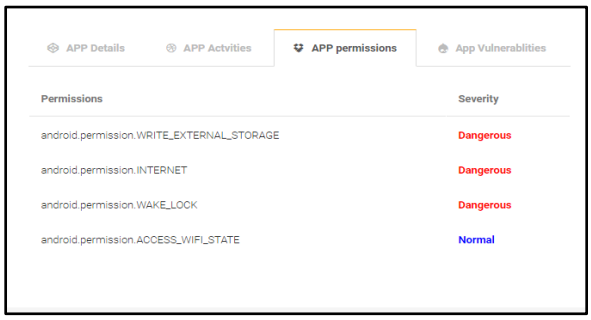

Figure 6: Vulnerability of the android net spoof mobile app

V. CONCLUSION

In this paper, AndroSF is proposed and experimented with static analysis of the android application security. AndroSF test for the vulnerabilities they find after reverse engineering the source code to understand the binary. IT achieves 98% success in finding. the vulnerability of the mobile application statistically. In future Android Security Framework (AndroSF) is enhanced as an intelligent, all-in-one mobile application (Android) automated pen-testing framework capable of performing both static and dynamic analysis. So that it can be used for effective and fast security analysis of Android mobile Applications and even zipped source code.

REFERENCES

[1]. Selvam ,R, ―Mobile Software Testing – Automated Test Case Design Strategies,‖ Rep., vol. 3, No. 4, 2013
[2]. Mamoon Rashid* and Lovepreet Kaur, ―Finding Bugs in Android Application using Genetic Algorithm and Apriori Algorithm, vol. 9, 2016.
[3]. Diana Gabriela NoemíBenítez-Mejía, Gabriel Sánchez-Pérez, Linda Karina Toscano-Medina, ―Android Applications and Security Breach,ISBN 978-1-4673-9379-9. 1181–1188, 2016