International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 2 | ISSN : 2456-3307

A Study on Automated Tools For Android Applications

[1]Dr. K B Priya Iyer, [2]Haritha J, [3]Divya G,[4]Sanchana G, [5]Subasri B

[1]Associate Professor, Department of Computer Science, M.O.P. Vaishnav College for Women(Autonomous), Chennai, Tamil Nadu, India

[2,3,4,5] Student-M.Sc. IT, M.O.P. Vaishnav College for Women(Autonomous), Chennai, Tamil Nadu, India

ABSTRACT

In recent years with the growth of mobile smart phones, the number of android application increases rapidly. Automated testing is necessary for these applications. Automated test are used almost during every testing process. The properly organized automated testing reduces mobile applications often simply called as "apps" are used often to perform number of activities. There is a wide choice of tools for automation some are free of cost. Some automation tools for android application are created and some appeared on the market and each has unique feature. The paper focuses on comparing the effectiveness of various automated tools for android applications based on the software quality factors. The paper also analyses the best tool for detecting faults by considering various features.

Keywords : Greendroid, prototype, UI, C-cores, Smartphone application, Conservation

I. INTRODUCTION

The testing process for andriod applications should be through automated tools. Properly organized automated testing process reduces the chances of error during execution process. There are N-number of testing tools for android application. This paper focuses on six android application testing tools Asdroid, Greendroid, Flowdroid, Amandroid, Reran, Dynodroid.

Asdroid is mainly used to detect the stealthy behaviour which is secretive or designed specifically to avoid detection. Now-a-days android application is becoming popular. It is an open source so that it will allow the user to install miscellaneous application like malicious ones, The major portion of existing operations like sending short messages, making phone calls or http connections and installing additional nasty components. Asdroid tool is mainly used to detect stealthy behaviour. Among 182 apps, asdroid reports stealthy behaviours in 113 apps with 28 false positive and 11 false negative.It has been said that 52-64% of existing malwares send stealthy premium rate message and make phone calls, which causes unexpected bills. Some apps can provide adult content. The existing technique is insufficient to detect this behaviour. Though this is insufficient to detect this behaviour Asdroid will protect the application.

Greendroid, tool concentrates mainly on energy consumption and energy insufficient for smart phone or android application. Present studies shows that many android applications are not well-organized in energy because of 2 major reason. One is that the framework for android exposes the hardware operations to developer. Another reason is, android application are mostly established by small teams without devoted quality assurance. The efforts in locating energy problems for android application are very difficult. This problem sorts out the tools for diagnosis process to compare the tools with one another.

Flowdroid, is a highly precise static fault study for android application. Taint analysis addresses the problems by analysing application and presenting malicious data flows to human analysts or to automated malware-detection tools. These approach track sensitive "tainted" information through the application by starting at a pre-defined source and then following the data flow until it reaches a given sink. Flowdroid successfully finds escape in a subset of 500 apps from google play and about 1000 malware apps from the virus allocated project. Flowdroid uses a very exact call graph which helps us to confirm flow and context concern. It needs a complete exhibiting of androids lifecycles and call-backs. Flowdroid produces a special technique which reflect all possible combination to make sure no fault is lost.

Amandroid is used to determine a flow and context sensitive way across android application component. It also tracks the inter-component communication activities. It can combine the component level information into application level information to perform inter application and intra application analysis. In this paper, the application analysis is feasible in terms of calculating the modern hardware. Main intention of this tool is to go beyond previous work intended for the specific problem. It can be used to report security difficulties that result from associations among multiple mechanisms from either the same or different application. It can provide assurance of the absence for specific security problems in an application with well-specified assumption.

The above tools have inter-component communication, enhance security problems, context sensitive and dataflow analysis and contains flow sensitive. These tools also perform GUI events which is user-friendly.

II. LITERATURE REVIEW

Amandroid can combine the component-level information into the app-level information to achieve intra-app and inter-app analysis. In this paper, the comprehensive app analysis is completely possible in terms of computing resources with modern hardware, demonstrate that one can easily leverage the results from this general analysis to build various types of specialized security analyses.[1].

In this, investigate the detection capability of main stream vs android specific tools to guide decision making during tools selection.[2]. Proposes an android malware detecting system that provides highly accurate classification and efficient sensitive data transmission analysis. Efficient approach to detect android malware.[3]. Still taint-analysis methods have the possible of perceiving such data leaks ahead of time, all approaches for Android use a number of coarse-grain approximations that can yield high numbers of missed leaks and false alarms.[4].

In this, it also propose DROIDBENCH, an open test suite for evaluating the effectiveness and accuracy of taint-analysis tools specifically for Android apps. These tools have been evaluated for their features, platforms, code coverage, and efficiency. [6]. Main part of this transformation is the utilization wall, the percentage of the transistors device that chip can switch at maximum frequency is decreasing exponentially due to power constraints.[9].

It is intended to make UI development easier and consistent through your application .A prototype that demonstrates the use of such cores to save energy broadly across the hotspots in the android mobile software.[11].To detect high precision low amplitude variability of its targets requires a robust model for the expected performance of its instruments.[13].

AUTOMATED TESTINGS TOOL FOR ANDROID APPLICATION:

Amandroid perform data flow and data dependency analysis. The tool uses flow-sensitive, context-sensitive data flow analysis to calculate object points-to information by building inter-procedural control flow graphs (ICFG). It used to track inter-component communication. It is feasible to compute the resource with modern hardware. Amandroid's analysis directly handles inter-component control and data flows, it can be used to address security problems that result from interactions among multiple components from either the same or different apps. Amandroid computes points-to information for all objects and their fields at each program point and calling context. It is extremely useful for analyzing the number of security problem. Amandroid can form a extremely inter-procedural control flow graph (ICFG).

Amandroid introduces component-level models instead of FlowDroid's whole app-level model. It can be easily expanded to achieve a number of specialized security analyses. It can be effectively address multiple specialized security problems.Malicious applications are constant threats to user data on Smartphone's as they could manipulate them by exploiting software weaknesses in legitimate mobile applications. It can be used to reduce risks during development. Malicious application could manipulate the sensitive user data.This proposes an Android malware detecting system that provides highly accurate classification and efficient sensitive data transmission analysis. It study's the use of dataflow application program interfaces (APIs) used to detect the android malware . The proposed scheme provides an efficient approach to detecting Android malware and invest privacy violations in malicious apps.

Reran is a record and replay tool for Smartphone's that have Android operating system. It captures input event sent from the phone to the OS of a user session and after ,that allows the sequence of events to be sent into the phone programmatically at high level. Reran captures the low level events and replays them that are triggered on the phone, which allows it to capture and playback GUI events such as touch screen gestures, and input sensors on device .

In this paper, EvoDroid is used to check system of Android apps. It combines two techniques to identify parts of the code open to be searched independently an android-specific program analysis; and an algorithm performs search step by step under the given info. Its main goal is to look for test cases that amplify code coverage . On the basis of state machine, it makes new test adequacy criteria. This test age group technique uses the copies and principles to produce test cases automatically. It delivers fully automatic testing that works on security policies of Smartphone platforms .

Dynodroid automatically generates inputs to Android apps. It generating both UI inputs ie.,touchscreen and system inputs ie., incoming sms. It allows interleaving inputs from machine and human. Through a sequence of events it interacts with its environment. Dynodroid is an observe-select-execute cycle, it observes which events are important to current state, selects those events, and execute those events to make a new state in which it repeats this process .Compared with the traditional model-based testing approaches, it enhances the diversity of test sequences by 85%, but reduces the number of them by 54%.

Flowdroid is a context, plan,f ield, object-responsive and lifecycle aware static spoil study tool intended for Android applications. The aim is to do investigation with very high recall and precision in flowdroid. To achieve this goal it has to accomplish two challenges: First challenge is to increase precision it needs to build an analysis that is context, plan, field and object-sensitive; to increase recall it has to create a complete model of Android's app lifecycle. The analysis is based on Soot and Heros. Flowdroid uses a very precise callgraph which helps to ensure flow and context sensitivity.
Flowdroid wants a complete modelling of Android's

lifecycles and callbacks. Flowdroid generates special main method which consider every possible combination to make sure no taint is lost. Flowdroid achieve 93% recall and 86% precision , where the Android benchmark suite.

Android applications is packaged in apk files (Android Packages),which is essentially zip-compressed archive. After unzipping an archive, Flowdroid searches for the application of lifecycle and callback methods and calls to sources and sinks.Next,the Flowdroid generates the dummy main method from the list of lifecycle and callback methods. This main method is used to generate a call graph and the inter-procedural control-flow graph (ICFG). Starting from the detected sources, the taint analysis then tracks taints by traversing the ICFG as explained . Flowdroid is configured with sources and sinks inferred by the SuSi project , by far the most comprehensive one available. The concrete lists of sources and sinks is available in the Flowdroid website. At the end, Flowdroid reports all the discovered flows from the sources to drops. This reports include full path information. Though Flowdroid is generally aiming for the sound analysis, it share some inherent limitations with most other static-analysis tools.

GreenDroid is a tool to examine the power consumption . The source code of a Android application is Greendriod .It will tell the power consumed when using the application.The Greendroid is a 45-nm multicore processor mobile application that is capable of executing general purpose mobile application with an energy 11 times less than the most energy efficient designs which is used today, due to this capability it performs at a better level. It does this through hundreds of highly specialized energy reducing cores called conservation cores. The c-cores spans roughly 95 % of the execution time of Android based test. The system has an array of tiles. Each tile uses a unique and standard template such as an energy efficient in-order processor which is of 32-Kbyte Level 1 (L1) data cache. Each tile is tightly coupled to the host CPU through the L1 data cache and an interface. This interface will let the host CPU to pass arguments to the c-core and helps them to perform context switches to reflect the changes in the application code.

This automated tool will automatically convert those hotspots into specialized circuits. The circuits are attached to a nearby host CPU through a L1 cache. The cold code returns the less frequently executed code that runs on the host CPU, where as the c-cores handle the hot code. This contain a logic that allows modification of the c-cores behaviour, which allows the CPU to inspect the codes variables during execution . A special compiler is responsible for identifying regions of the code generating CPU code and c-core patches. For the code that is translated to specialized circuits the c-cores releases less energy per instruction. The overall system energy savings have a negative impact three things: the energy for running cold code on a less efficient CPU , energy spent in the L1 cache, and the energy spent in seepage and for clock power. It decreases the effect by high execution exposure on the c-cores, regions that cover even less than 1% of total execution exposure.

AsDroid is a automated tool that is used to detect the stealthy behaviours in android applications.Now-a-days android smartphones are becoming popular.The open nature of Android is that it will allows the users to install miscellaneous applications, including the malicious things, from third-party association without accurate sanity checks.It perform the stealthy operations such as sending small messages, calling random phone calls and installing additional unwanted malicious components.It has been found that in many cases, there is no direct connection between an API intent.So that,it formally present their design using datalog rules.This also implement a prototype which is called as AsDroid (Anti-StealthDroid). It propose a technique to detect the behavior.So ,it created a model which detect the stealthy behavior as the program behavior that mismatches with user interface, which denotes the user's expectation of program behavior. To evaluate AsDroid, it download a software named pool of 182 apps that are normally problematic by looking at their permissions. Among the 182 application, AsDroid has been reported that stealthy behaviors are presented in 113 application, with 39 false as 28 false positives and 11 false negatives. In China, it has been reported in March 2012 that more than 2,10,000 Chinese smartphones were affected by a kind of malwares that could make stealthy HTTP connections inducing charges. It caused around 8 million dollars in loss .

Next is the ASTROID Simulator,Asdroid Simulator is a software package for realistic modeling of high precision space-based imaging observations of a selected stellar field.The preparation of a space-mission that carries out any kind of image to detect high- precision,low-amplitude variability of its targets that requires a robust model for the expected performance of those instruments.It presented a high-precision simulation software package and designed to pretend the CCD images that will include all accurate models of the CCD and its electronics.It also include telescopeoptics, the stellar field, the jitter movements of the spacecraft, and all important natural noise resources.Next is,The Stoner-Wohlfarth astroid is a fundamental object in magnetism. It separates regions of magnetic field space with two stable magnetization equilibria from those with only one stable equilibrium and it characterizes the magnetization reversal of nano-magnets that induced by applied magneticfields.The study of magnetization reversal has provoked continuous interest during the last 60years.

SOFTWARE QUALITY FACTORS :

	SOFTWARE QUALITY FACTORS									
TOOLS	Feasibility	Performance	Efficiency	Highly Precise	Robustness	Usability	Correctness	Scalability	Maintainability	Capability
Asdroid	yes			yes						Yes
Amandroid	yes			yes				Yes		
Dynodroid					yes	yes	Yes			
Greendroid		yes	Yes							
Flowdroid			yes	Yes					yes	
Reran		yes			yes	yes	yes	yes		

Table 1

In table 1, there are various software quality factors for each tools whereas , AsDroid contains feasibility, highly precise and capability. Amandroid has feasibility, high precise and scalability . Robustness, usability and correctness are present in Dynodroid whereas Greendroid is popular in performance, and efficiency .The factors present in Flowdroid are efficiency, high precise and maintainability. Finally Reran has performance, robustness, usability, correctness and scalability.

FEATURES OF ANDROID TOOLS:

FEATURES OF ANDROID TOOLS						
Features	Amandroid	Greendroid	Asdroid	Reran	Flowdroid	Dynodroid
Inter component communication	yes					
Context sensitive data flow	yes				yes	
Enhance security features	yes		yes			
Playback UI and system input				yes		Yes
Web	yes					yes
Code coverage				yes		
Generate test case automatically				yes		
Playback GUI event	yes			yes		yes
Android app	yes	yes	yes	yes	Yes	yes
IOS apps						yes
Recall and precision					yes	
multithreading					yes	
Power consumption		yes				
Stealthy behaviour			yes			

Table 2

Table 2 describes the features of automated testing tools. Amandroid contains intercomponent communication, context sensitive data flow and GUI events. The main feature found in Greendroid is to analyse the power consumption. The Reran's main goal is to look for test cases that amplify code coverage. The most important feature in Asdroid is to detect the stealthy behaviour. Dynodroid supports testing of iOS apps and generate both UI inputs and system inputs. The main aim of Flowdroid is to analyse high recall and precise.

III. CONCLUSION

Test automation can bring many benefits to the android application testing cycles that allows to build better application with less effort as well as it contains less time consuming. Many companies still runs only manual testing because they don't know how to properly integrate automated testing in their application development process.In this study, certain factors and features has been implemented. The best tools for android application is Greendroid and Asdroid because it performs power consumption and stealthy behaviour.

IV. REFERENCES

[1]. FENGGUO WEI, University of South Florida SANKARDAS ROY, Bowling Green State University XINMING OU, University of South Florida ROBBY, Kansas State University , Amandroid: A Precise and General Inter-component Data Flow Analysis Framework for Security Vetting of Android Apps. http://www.arguslab.org/documents/tech_reports/2017/amandroid_fgwei_2017.pdf

[2]. Tosin Daniel Oyetoyan and Marcos LordelloChaim , Comparing Capability of Static Analysis Tools to Detect Security Weaknesses in Mobile Applications https://brage.bibsys.no/xmlui/bitstream/handle/11250/2456625/paper2.pdf?sequence=7

[3]. Effective detection of android malware based on the usage of data flow APIs and machine learning https://www.sciencedirect.com/science/article/pii/S0950584916300386

[4]. D. Amalfitano, et al., "MobiGUITAR: Automated model-based testing of mobile apps," IEEE Software, vol. 32, pp. 53-59, 2015.

[5]. L. Gomez, et al., "Reran: Timing-and touch-sensitive record and replay for android," in 2013 35th International Conference on Software Engineering (ICSE), 2013, pp. 72-81.

[6]. R. Mahmood, et al., "Evodroid: Segmented evolutionary testing of android apps", in Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2014, pp. 599-609.

[7]. A. Machiry, et al., "Dynodroid: An input generation system for android apps," in Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, 2013, pp. 224-234.

[8]. T. Su, "FSMdroid: guided GUI testing of android apps," in Proceedings of the 38th International Conference on Software Engineering Companion, 2016, pp. 689-691.

[9]. http://www.internationaljournalssrg.org/IJECE/2016/Volume3-Issue8/IJECE-V3I8P106.pdf

[10]. http://home.deib.polimi.it/silvano/FilePDF/ARCMULTIMEDIA/GreenDroid_Spiriti_Morici.pdf

[11]. http://www.theijes.com/papers/vol6-issue3/J0603018692.pdf

[12]. Jianjun Huang, Xiangyu Zhang, Lin Tan, Peng Wang, Bin Liang Published 2014 in ICSE,"AsDroid: detecting stealthy behaviors in Android applications by user interface and program behavior contradiction"

[13]. P. Marcos-Arenal1, W. Zima1, J. De Ridder1, R. Huygen1, C. "The ASTROID Simulator Software Package: Realistic Modelling of High-Precision High-Cadence Space-Based Imaging"

[14]. https://scholar.google.co.in/scholar?hl=en&as_sdt=0%2C5&q=Distortion+of+the+Stoner-Wohlfarth+astroid+by+a+spin-polarized+current&btnG