

Ad Capsule: Practical Confinement of Advertisements in Android Applications

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

PADAVALA SANDHYA RANI(37110530)

PAJJURI GAYATRI(37110533)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING **BONAFIDE CERTIFICATE**

This is to certify that this project report is the bonafide work of **PADAVALA SANDHYA RANI (37110530)** and **PAJJURI GAYATRI (37110533)** who carried out the project entitled “ **Ad Capsule: Practical Confinement of Advertisements in Android Applications** ” under my supervision from **August 2020** to **March 2021**.

Internal Guide

Dr. L. Lakshmanan, M.E., Ph.D.,

Head of the Department

Dr. S. Vigneshwari, M.E., Ph.D.,

Dr. L. Lakshmanan, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **PADAVALA SANDHYA RANI** hereby declare that the Project Report entitled “**Ad Capsule: Practical Confinement of Advertisements in Android Applications**” is done by me under the guidance of **Dr. L. Lakshmanan, M.E., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who are helpful in many ways for the completion of the project.

ABSTRACT

Nowadays, app developers tend to integrate advertisement libraries (or ad libraries) into their appstoget revenue from adnetworks. However, researches haveshown thatboth ad libraries and ad contents could raise serious security and privacy concerns. In this paper, we propose Ad Capsule, a user-level solution to practically confine advertisements, including ad libraries and ad contents. Our solution does not need to change the Android framework, nor requires the root privilege, thus can be readily deployed. Specifically, we propose the permission sandbox, which isolates the permissions used by ad libraries from the host app, and the file sandbox, which separates the file operations ofadvertisements. Thead library andad content cannot read or write any file outside this sandbox. Wehave implemented a prototype of Ad Capsule. Our evaluation results indicate that Ad Capsule can successfully enforce security policies to block attempts of accessing private information or manipulating files of the host app, and the performance overhead introduced by Ad Capsule is low.

TABLE OF CONTENTS

ABSTRACT	v
LIST OF FIGURES	viii

CHAPTER	TITLE	PAGE
No		
1	INTRODUCTION	1
1.1	INTRODUCTION	1
2	LITERATURE SURVEY	2
2.1	Practical Policy Enforcement for Android Applications	2
2.2	Hybrid User-level Sandboxing of Third-party Android Apps	2
2.3	Understanding Malicious Advertisements	2
3	SYSTEM ANALYSIS	3
3.1	Existing system	3
3.2	Proposed system	3
4	REQUIREMENTS SPECIFICATION	4
4.1	Introduction	4
4.2	Hardware and Software specification	4
4.3	Technologies Used	4
4.3.1	Java	5
4.3.2	Introduction to Java	5
4.3.3	Working of Java	5
4.3.4	The java programming language	5
4.3.5	The java platform	7
4.4	Apache tomcat server	7
4.5	Andriod introduction	8
5	Project Purpose and Scope	12

5.1	Purpose	12
5.2	Project Scope	12
5.3	Product Perspective	12
5.4	System Features	13

5.5	Design and Implementation Constraints	13
5.6	Other Nonfunctional Requirements	13
6	SYSTEM DESIGN	15
	AN	
	D METHODOLOGY	
6.1	Architecture Diagram	15
6.2	Sequecnce Diagram	16
6.36.3	Use case Diagram	16
6.46.4	Activity Diagram	17
6.56.5	Collaboration Diagram	18
6.66.6	Data flow diagram	19
6.76.7	Class Diagram	20
6.8	Modules	21
6.9	Module Explanation	21
7	CODING AND TESTING	23
7.1	Coding	23
7.27.2	Coding Standards	23
7.37.3	Test Procedure	24
7.47.4	Test Data and Output	24
7.57.5	Testing techniques	24
8.	RESULTS AND DISCUSSION	27
9.	CONCLUSION AND FUTURE WORK	31
	REFERENCES	32
	APPENDIX	34
	A.PLAGIRISM REPORT	34
	B.JOURNAL PAPAER	35
	C.SOURCE CODE	40

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.1	BLOCK DIAGRAM	6
4.2	JAVA VIRTUAL	6
4.3	MACHINE JAVA PLATFORM	7
4.4	ACTIVITY DIAGRAM	9
6.1	ARCHITECTURE DIAGRAM	15
6.2	SEQUENCE DIAGRAM	16
6.3	USE CASE DIAGRAM	17
6.4	ACTIVITY DIAGRAM	18
6.5	COLLABORATION DIAGRAM	19
6.5.6	CLASS DAIGRAM	20
8.1	ADMIN LOGIN PAGE	27
8.2	REGISTERING APPLICATION	28
8.3	HEALTH CARE APP	29
8.4	APP INSTALLED MOBILE	29
8.5	AD ASKING ACCESS PERMISSIONS	30

1.1 Introduction:

Aim:

C H A P T E R 1 I N T R O D U C T I O N

The main aim of this paper is to have control of the in app advertisements in order to restrict the unauthorized access of personal information.

Synopsis:

Due to the proliferation of mobile phones in recent years there are many advantages and some disadvantages in smart-phones usage. Many applications like Ecommerce, Social media and games get access of user's permission to use some privileges. Normally while installing any application we used to grant permission to the particular application to access

all the privileges. So when the applications which we install in our smart-phones have some tie-up with some advertisement company. So that for the benefits of the app developer those ads will run in our app. Here in this case the advertisements can behave maliciously some times. This malicious behavior extracts personal information like user's contact and files by using the permission access policy of particular app. This data theft may happen without the knowledge of user; the malicious ad will upload the collected information to the desired server. We propose a third party Server to validate the advertisement before the ad get loaded into the app.

CHAPTER 2

LITERATURE SURVEY

2.1 : Practical Policy Enforcement for Android Applications

We can able to develop a solution that bypasses the need to modify the Android OS while providing much of the security and privacy that user's desire. We can automatically repackage arbitrary applications to attach user-level sandboxing and policy enforcement code. The results has been demonstrated that our framework works well with a set of real-world applications, and incurs only modest overhead on benchmark programs.

2.2 Hybrid User-level Sandboxing of Third-party Android Apps

This study explores two complimentary user-level sandboxes to interpose and regulate an app's access to sensitive APIs. It causes some platform's security and privacy controls which requires extensive modification to the operating system, which has significant usability issues. It lacks in retrieve users sensitive information and its privacy issues. . The Native Guard isolates native libraries of an Android application into a second application where unnecessary privileges are eliminated. There is no evaluation of the performance which requires applications to issue requests for additional permissions to their users at install time.

2.3. Understanding Malicious Advertisements

we perform the first large-scale study of the safety of the advertisements that are encountered by the users on the Web. This can help both advertisement networks and website owners in securing their web pages and in keeping their visitors safe. The data is harder to justify, such as the user's call logs, phone number, browser bookmarks, or even the list of apps installed on the phone. This process examine the ad libraries for risks, ranging from uploading sensitive information to remote servers to executing code from Internet sources. This problem clearly show the need for better regulating the way ad libraries are integrated in Android app.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Though we have shown security concerns in previous sections reported by other researchers, we want to further understand the state of the art of ad libraries and the privacy related behaviors. Due to the special nature of the Android ecosystem in China, e.g., highly customized Android systems without Google services, we also collected 29 ad libraries which are popular in China.

Moreover, to check whether ad libraries perform more privacy-concerned operations than other libraries, we collected 119 popular development tool libraries on App-Brain for comparison. This malicious behavior extracts personal information like user's contact and files by using the permission access policy of particular app.

Normally while installing any application we used to grant permission to the particular application to access all the privileges. So when the applications which we install in our smart-phones have some tie-up with any advertisement company. So for the benefits of the app developer those ads will run in our app. Here in this case the advertisements can behave maliciously some times. This malicious behavior extracts personal information like user's contact and files by using the permission access policy of particular app.

3.2 PROPOSED SYSTEM

We propose a third party Server to monitor the advertisement library, Here the User whenever install any new application the third party server will start checking the incoming adds. So before loading a new advertisement into the application our system will analyze the triggering components inside the ads and its behavior. If there is any chance of accessing the user details or files or contacts, thus an ad can get unauthorized access permissions from the hosted app.

Our system is capable of detecting that kind of ads and restricts their access, then pop up the required access permission to user. Only when the user accepts that permission it will allow the ad to run inside app. Otherwise our system will block the malicious ads.

CHAPTER 4

REQUIREMENT SPECIFICATIONS

4.1 INTRODUCTION

There is no doubt that smart phones have become an essential part of our life. One potential reason of such popularity is that there are millions of applications (or apps) available to users. For instance, 2; 600; 000 apps are available in Google Play Store. Among these apps, more than 90% are free ones. Users download and use these apps without paying app developers.

4.2 HARDWARE AND SOFTWARE SPECIFICATION

4.2.1 HARDWARE REQUIREMENTS

- RAM : 4GB & Above
- Processor : Pentium IV & Above
- Hard disk : 250GB and Above

4.2.2 SOFTWARE REQUIREMENTS

- Windows 7 operating system and above
- JDK 1.7
- Tomcat7➤
- My-SQL 5
- Android Studio 3.1

4.2.3 Android Phone Requirements:

- OS : Jelly-Bean & Above
- No of Devices : 1

4.3 TECHNOLOGIES USED

- Java
- JSP Servlet.
- Android

4.3.1 JAVA

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

4.3.2 INTRODUCTION TO JAVA

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

4.3.3 WORKING OF JAVA

When the interpreter executes a class, it looks for a particular method by the name of **main**, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv [] of C), and is declared as a static method.

Java consists of two things :

- Programming language
- Platform

4.3.4 THE JAVA PROGRAMMING LANGUAGE

Java is a high-level programming language that is all of the following:

- Simple
- Object-oriented
- Distributed
- Secure
- Architecture-neutral

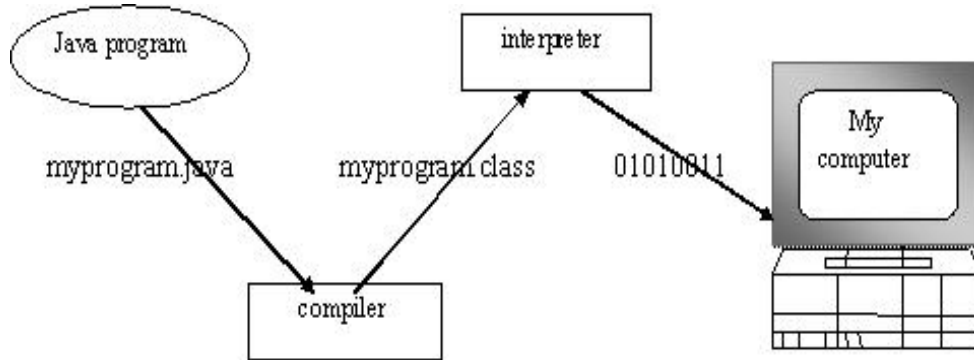


Fig 4.1

You can think of Java byte codes as the machine code instructions for the **Java Virtual Machine (JVM)**. Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be

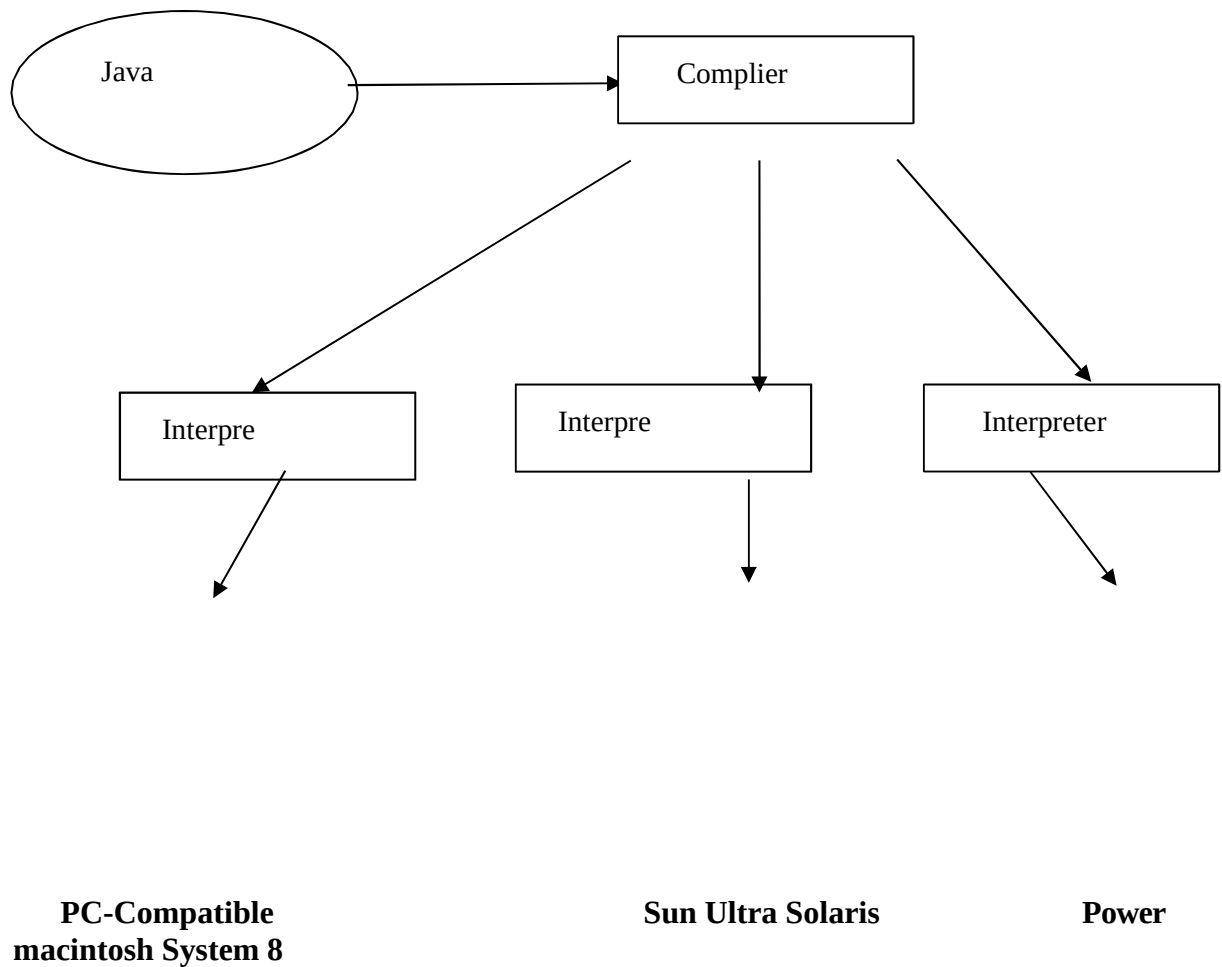


Fig 4.2

4.3.5 THE JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms.

The Java platform has two components :

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.

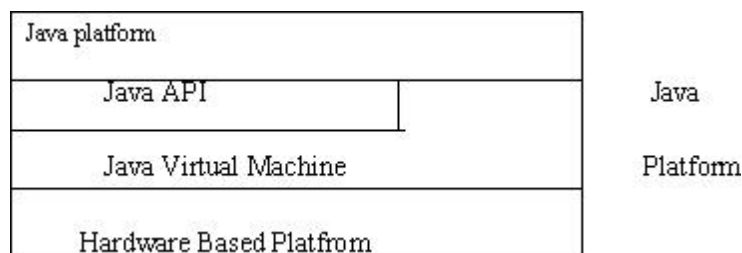


Fig.4.3

4.4 APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

Product features

Tomcat 3.x (initial release)

- implements the Servlet 2.2 and JSP 1.1 specifications
- servlet reloading
- basic HTTP functionality Tomcat 4.x
- implements the Servlet 2.3 and JSP 1.2 specifications
- JSP engine redesigned as Jasper
- Coyote connector

HISTORY

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

4.5 Android Introduction:

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. The operating system has developed a lot in last 15 years starting from black and white phones to recent smart phones or mini computers. One of the most widely used mobile OS these days is android. The android is software that was founded in Palo Alto of California in 2003.

The android is a powerful operating system and it supports large number of applications in Smart phones. These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system means that it's free and any one can use it. The android has got millions of apps available that can help you managing your life one or other way and it is available low cost in market at that reasons android is very popular.

The android development supports with the full java programming language. Even

other packages that are API and JSE are not supported. The first version 1.0 of android development kit (SDK) was released in 2008 and latest updated version is jelly bean. The android is a operating system and is a stack of software components which is divided into five sections and four main layers that is

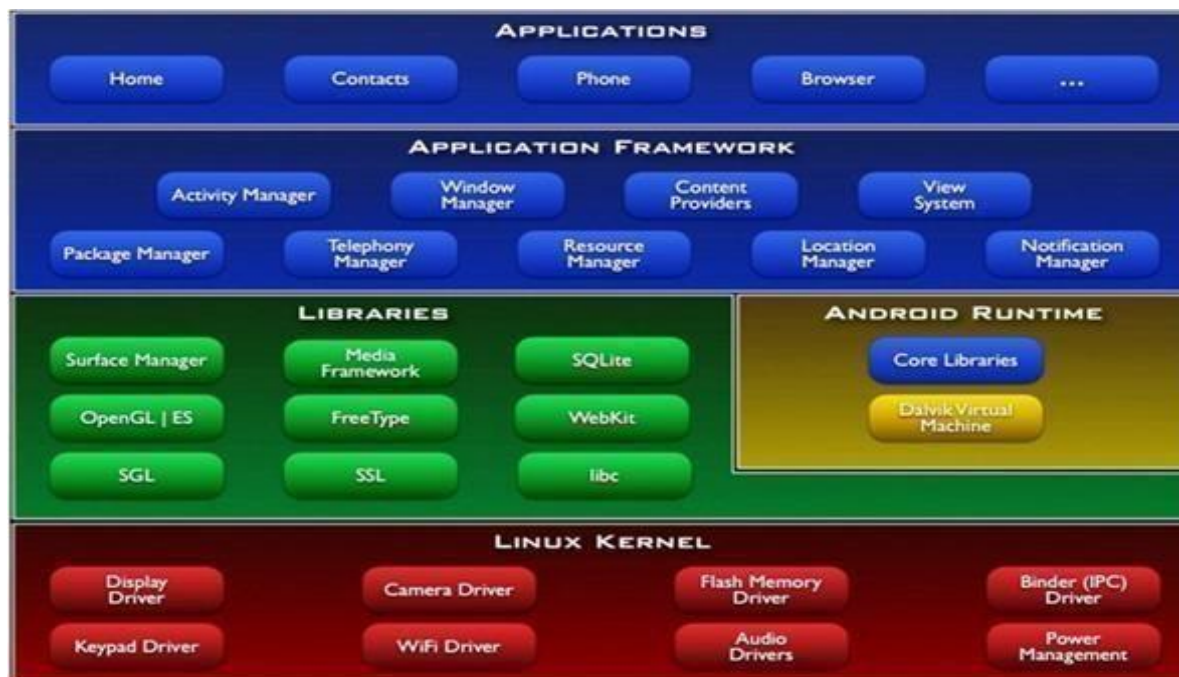


Fig 4.4

4.5.1 Linux kernel:

The android uses the powerful Linux kernel and it supports wide range of hardware drivers. The kernel is the heart of the operating system that manages input and output requests from software. This provides basic system functionalities like process management, memory management, device management like camera, keypad, display etc the kernel handles all the things. . The Linux is really good at networking and it is not necessary to interface it to the peripheral hardware. The kernel itself does not interact directly with the user but rather interacts with the shell and other programs as well as with the hard ware devices on the system.

4.5.2 Libraries:

The on top of a Linux kenneel there is a set of libraries including open source web browser such as webkit, library libc. These libraries are used to play and record audio and

video. The SQLite is a data base which is useful for storage and sharing of application data. The SSL libraries are responsible for internet security etc.

4.5.3 Android Runtime:

The android runtime provides a key component called Dalvik Virtual Machine which is a kind of java virtual machine. It is specially designed and optimized for android. The Dalvik VM is the process virtual machine in the android operating system. It is software that runs apps on android devices. The Dalvik VM makes use of Linux core features like memory management and multithreading which is in a java language. The Dalvik VM enables every android application to run its own process. The Dalvik VM executes the files in the .dex format.

4.5.4 Application frame work:

The application frame work layer provides many higher level services to applications such as windows manager, view system, package manager, resource manager etc. The application developers are allowed to make use of these services in their application.

4.5.5 Applications:

You will find all the android applications at the top layer and you will write your application and install on this layer. Examples of such applications are contacts, books, browsers, services etc. Each application performs a different role in the overall applications.

Advantages:

- Android is Linux based open source operating system , it can be developed by any one
- Easy access to the android apps
- You can replace the battery and mass storage, disk drive and UDB option
- Its supports all Google services
- The operating system is able to inform you of a new SMS and Emails or latest updates.
- Its free to customize
- Can install a modified ROM
- Its supports 2D and 3D graphics

- It supports Multitasking
- Android phone can also function as a router to share internet
- Android is Linux based open source operating system , it can be developed by any one
- Easy access to the android apps
- You can replace the battery and mass storage, disk drive and UDB option
- Its supports all Google services
- The operating system is able to inform you of a new SMS and Emails or latest updates.

CHAPTER 5

PROJECT PURPOSE AND SCOPE

5.1 Purpose

The purpose of this project is to compensate for their work, app developers have incentives to integrate one or more advertisement (or ad) libraries in their apps (or host apps), and get paid. To facilitate the integration, ad service providers offer SDKs or libraries to app developers. App developers integrate these libraries with minor efforts, sometimes adding only small amount lines of code. After that, ad libraries communicate with the ad network, retrieve and render ad contents.

Ad libraries and the host app are running inside a same virtual machine and sharing the same privileges. Thus, ad libraries could probe and abuse permissions of the host app, which causes privacy concerns to users, and further ruins the reputation of the host app. However, in order to get revenue from the ad network, app developers have no choice but trusting and integrating ad libraries. We need a practical solution to confine these libraries.

5.2 Project Scope

Our system is mainly developed for app developers. In particular, app developers integrate our library together with ad libraries into their projects, and add a few lines of code to activate Ad Capsule when the app is started. After that, developers can define the security policies according to ad library's package name list and put the policy file into the raw apk file.

Since our system has the same privilege with the ad library, and they are running inside the same process, malicious ad libraries could bypass our system leveraging the similar hooking mechanisms we used. Further, ad libraries can invoke the services or remove our system by native code. Our system took this into consideration and proposed several mechanisms for self-protection.

5.3 Product Perspective

In this work, in-app ad libraries and ad contents are un trusted. Ad libraries are un

trusted either because they are aggressive or malicious, or they are not malicious but vulnerable. For better ad targeting, ad libraries have the motivation to aggressively track users, using the retrieved identifier, location data or the list of installed apps

5.4 System Features

In summary, this paper makes the following contributions:

We propose a practical solution called AdCapsule to confine advertisements, including both ad libraries and ad contents, without the need to change the Android framework or existing ad libraries.

5.5 Design and Implementation Constraints

5.5.1 Constraints in Analysis

- ◆ Constraints as Informal Text
- ◆ Constraints as Operational Restrictions
- ◆ Constraints Integrated in Existing Model Concepts

5.5.2 Constraints in Design

- ◆ Determination of the Involved Classes
- ◆ Determination of the Involved Objects
- ◆ Determination of the Involved Actions
- ◆ Determination of the Require Clauses

5.5.3 Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

5.6 Other Nonfunctional Requirements

5.6.1 Performance Requirements

The application at this side controls and communicates with the following three main

general components.

- embedded browser in charge of the navigation and accessing to the web service;
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following. Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

5.6.2 Safety Requirements

1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.

CHAPTER 6

SYSTEM DESIGN AND METHODOLOGY

6.1 Architecture Diagram:

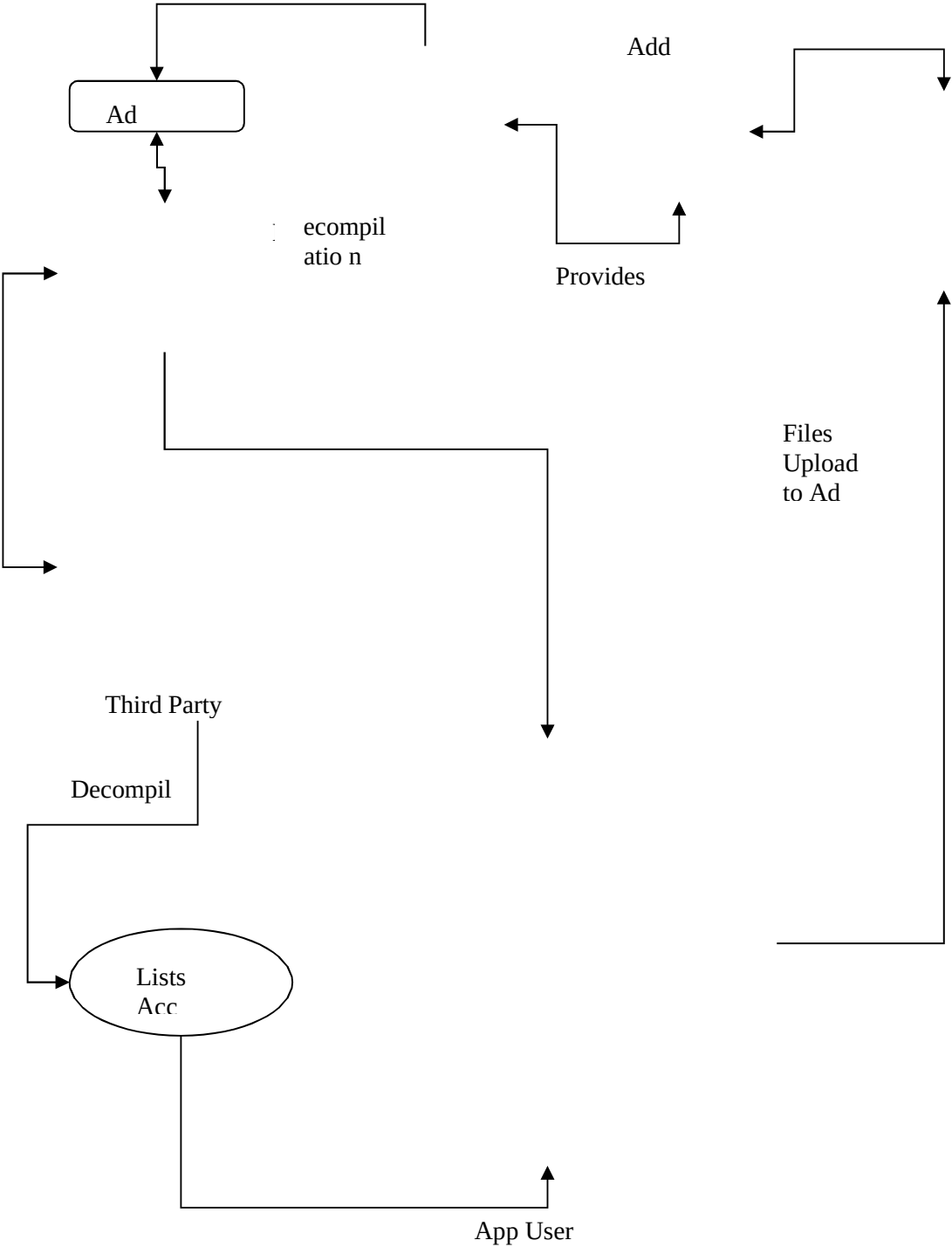


Fig: 6.1

6.2 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

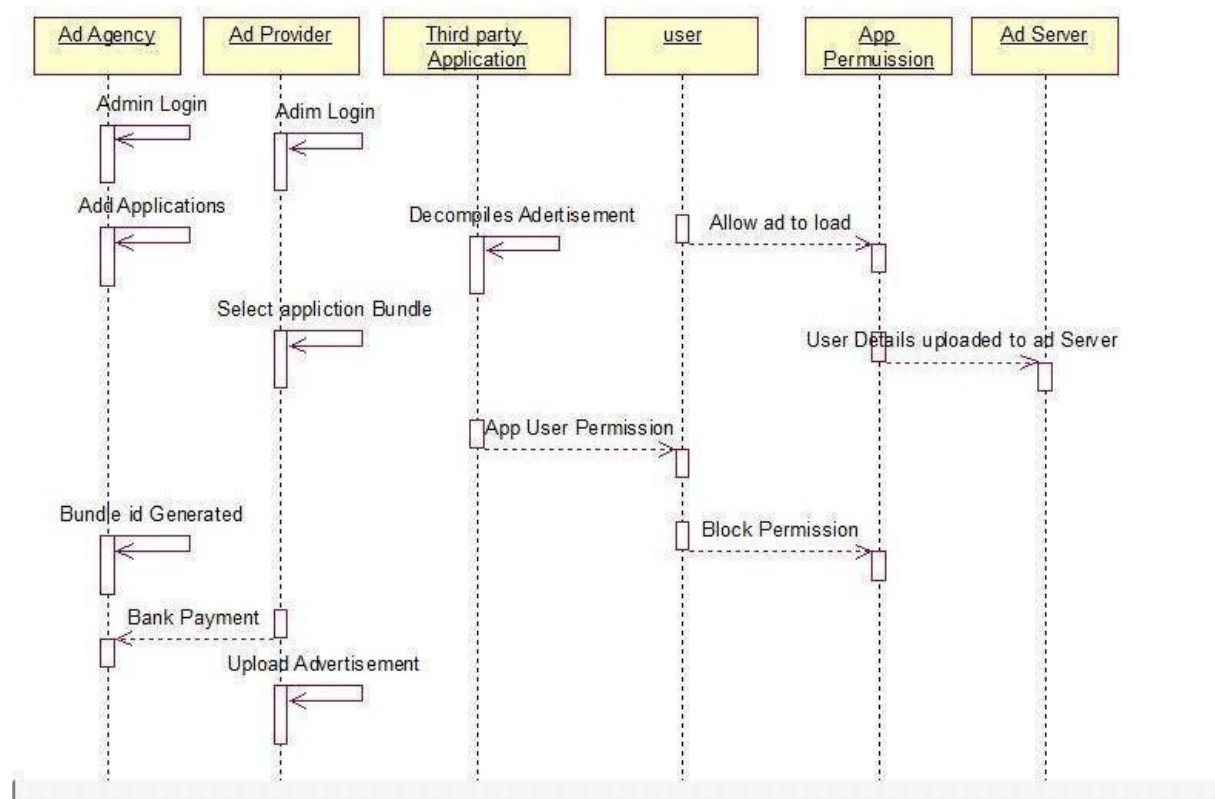


Fig 6.2

6.3 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

6.3.1. USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality

provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

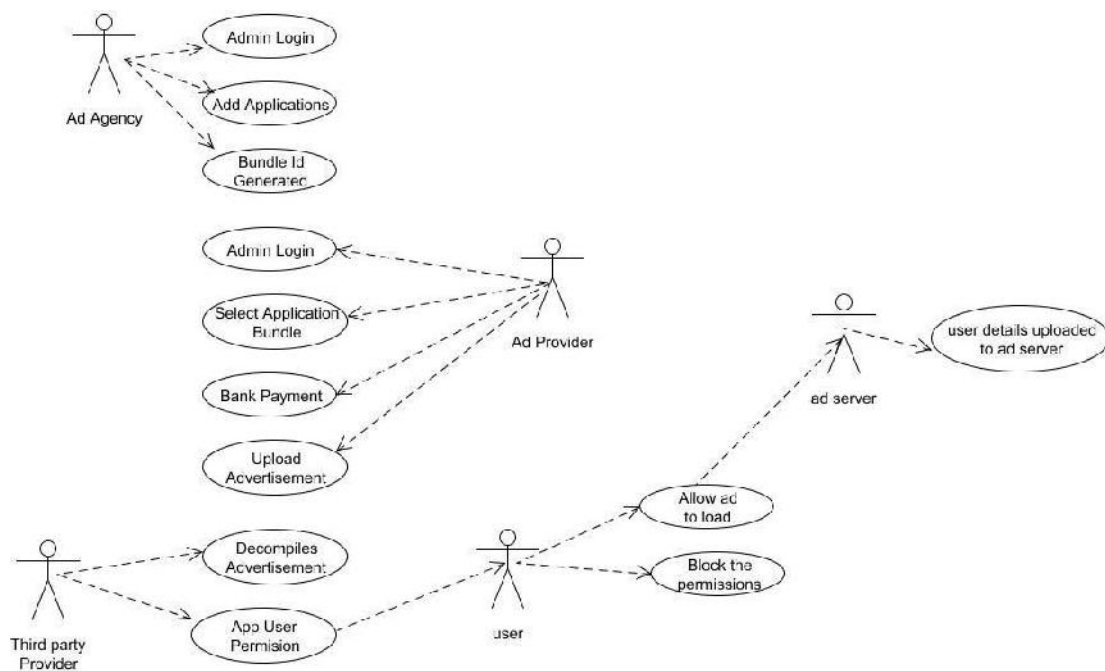


Fig 6.3

6.4 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.

- An encircled circle represents the end of the workflow.

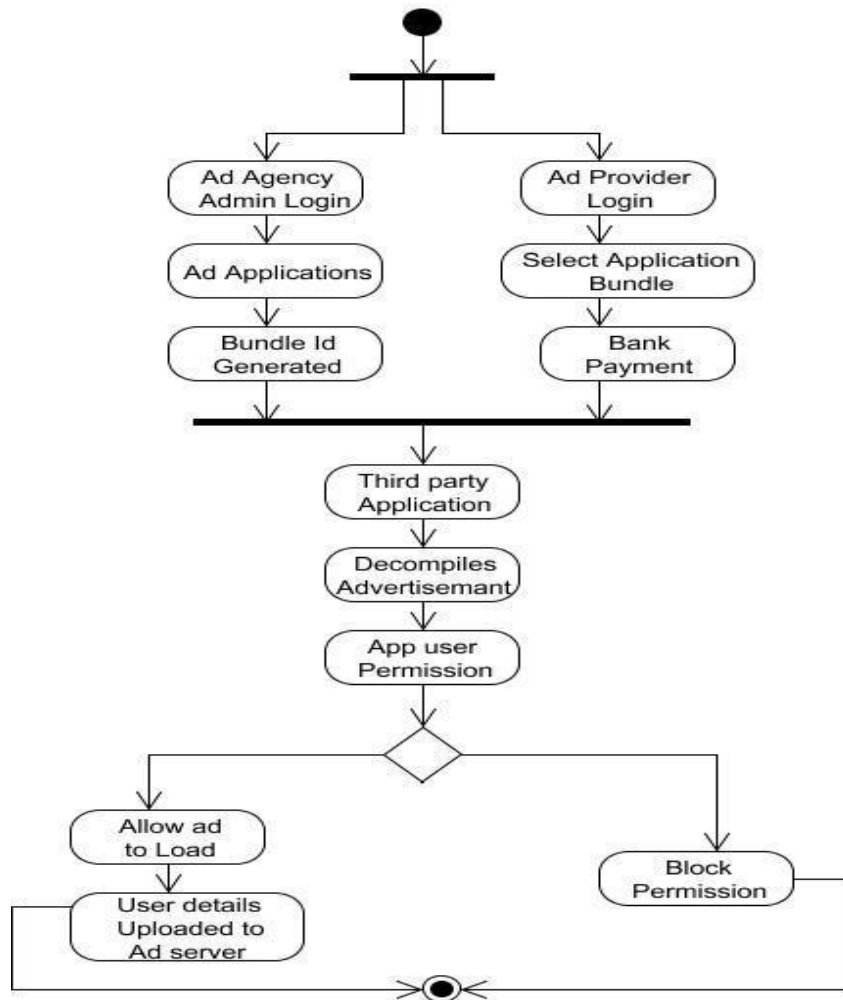


Fig 6.4

6.5 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

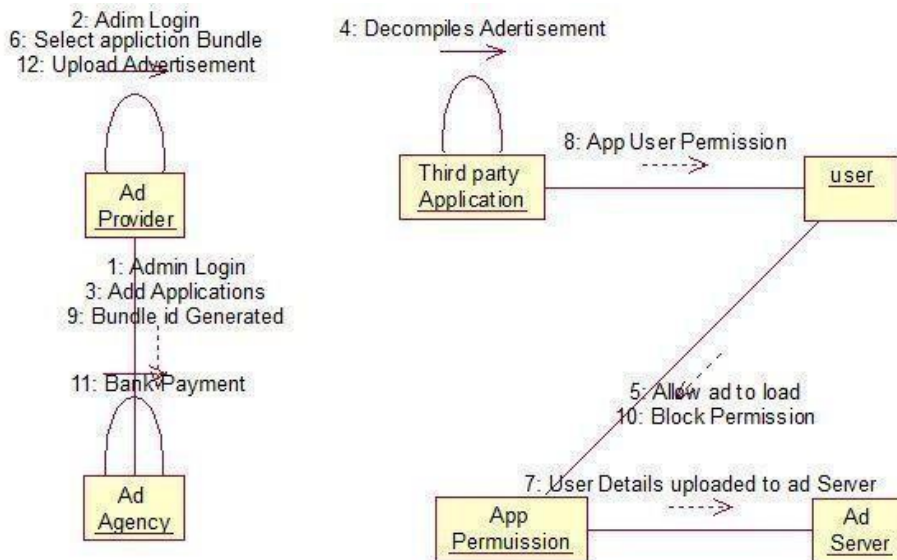
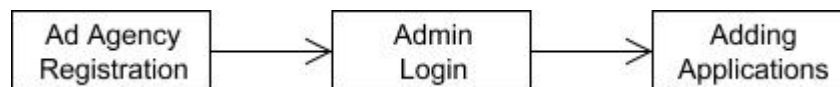


Fig 6.5

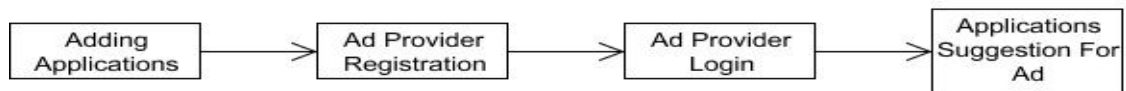
6.6 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

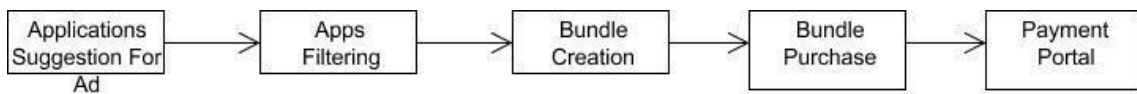
Level 0:



Level 1:



Level 2:



6.7 Class Diagram

A Class diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

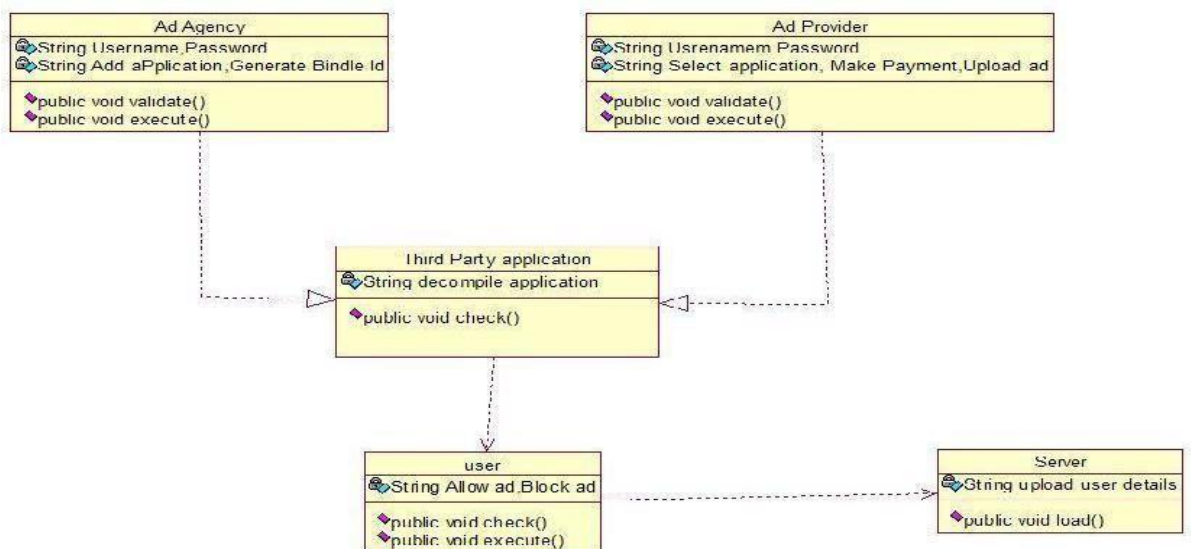


Fig 6.5

In this work, we have designed a user-level solution to confine advertisements, including ad libraries and ad contents. Our system could be readily deployed since it does not need to change Android framework, nor needs the root privilege. It leverages two sandboxes to confine privacy-concerned operations and file operations of advertisements. We have implemented a prototype of our system and the evaluation results demonstrate the effectiveness and compatibility of our system, and the performance overhead is low.

6.8 MODULES

- ▢ **Adding Applications in Ad Agency**
- ▢ **Ad Provider Purchasing APP Bundle**
- ▢ **Hosting Ad in APP Bundle**
- ▢ **Third-Party Service Provider**

6.9 MODULE EXPLANATION:

6.9.1 Adding Applications in Ad Agency:

In-order to use our application the users need to do user registration. After Ad Agency user Authentication the users or application developers will add their applications to the Ad-Agency portal. While adding an application they has to give basic information about the app like number of downloads, number of daily user logins, ratings, and App Id is mandatory. After that when an Ad-Agency admin sign-in into the portal the requested applications for advertisements will be shown. Here the admin has the privilege to set amount for the requested applications.

6.9.2 Ad Provider Purchasing APP Bundle:

After the Ad Provider User registers into Ad Agency, The Provider as well as Agency has to generate new bank account using our Bank application. After accounts created successfully they can do money transfer for any desired accounts. Then the Ad Provider has to sign-in the Ad-Agency page and look for apps satisfying their criteria. They can select the apps in a bundle by applying filters based on their type of advertisement content, ratings,

cost , number of downloads and number of daily users. After filtration an app bundle will be formed with Agency's applied offers. So now the Ad Provider has to purchase the app bundle.

6.9.3 Hosting Ad in APP Bundle:

After bundle purchase the Provider has to upload the suitable ad content which is of types transport, education or health. After uploading the ad content into Agency, it will be automatically loaded into the end applications whose app id is there in the selected bundle. For each category of application we will be showing a related category Android Application. .For example we run a health related app. Once an advertisement gets hosted into that application the ad content will start initiating its malicious behavior. If the application uses permission like sms or contacts the ad-content will make use of it and send all user's sms or contacts to their ad-Server automatically without user's knowledge in background.

6.9.4 Third-Party Service Provider:

We implement a Third Party Service Provider in Ad-Agency side to verify the malicious contents of incoming advertisements .So whenever a new adcomponent is loaded into the empty spaces allocated for specific ads. The ads get loaded into certain locations inside the host apps. We need to do a preprocessing technique in our proposed Third party Service Provider. So when an advertisement is loaded first into the application the Service provider will check or process all the running components using decompilation and find out the possible permission access that the ads going to get from the hosted application.A popup will be shown to the app user and get the authorized access. Hereby we can restrict the ad components from getting access to user's sensitive data. Thus we can handle the ads from misbehaviors. This is known as confinement of advertisements.

CHAPTER 7

CODING AND TESTING

7.1 CODING

Once the design aspect of the system is finalized, the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required; it easily screws into the system.

7.2 CODING STANDARDS

Coding standards are guidelines to programming that focus on the physical structure and appearance of the program. They make the code easier to read, understand, and maintain. This phase of the system actually implements the blueprint developed during the design phase.

7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. These conventions are as follows:

7.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

7.2.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the

code.

7.2.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

7.3 TEST PROCEDURE SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. Testing checks for the errors, as a whole of the project testing involves the following test cases:

7.4 TEST DATA AND OUTPUT

7.4.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

7.4.2 FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Stress Test
- Structure Test

7.5 TESTING TECHNIQUES / TESTING STRATEGIES

7.5.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences.

7.5.2 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

7.5.3 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

7.5.4 PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords

are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output.

7.5.5 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- * The function or performance characteristics confirm to specifications and are accepted.
- * A validation from specification is uncovered and a deficiency created.

CHAPTER 8

RESULTS AND DISCUSSION

Thus we propose a third party Server to monitor the advertisement library, Here the User whenever install any new application the third party server will start checking the incoming adds. So before loading a new advertisement into the application our system will analyze the triggering components inside the ads and its behavior.

If there is any chance of accessing the user details or files or contacts, thus an add can get unauthorized access permissions from the hosted app. Our system is capable of detecting that kind of ads and restricts their access, then pop up the required access permission list to the user.

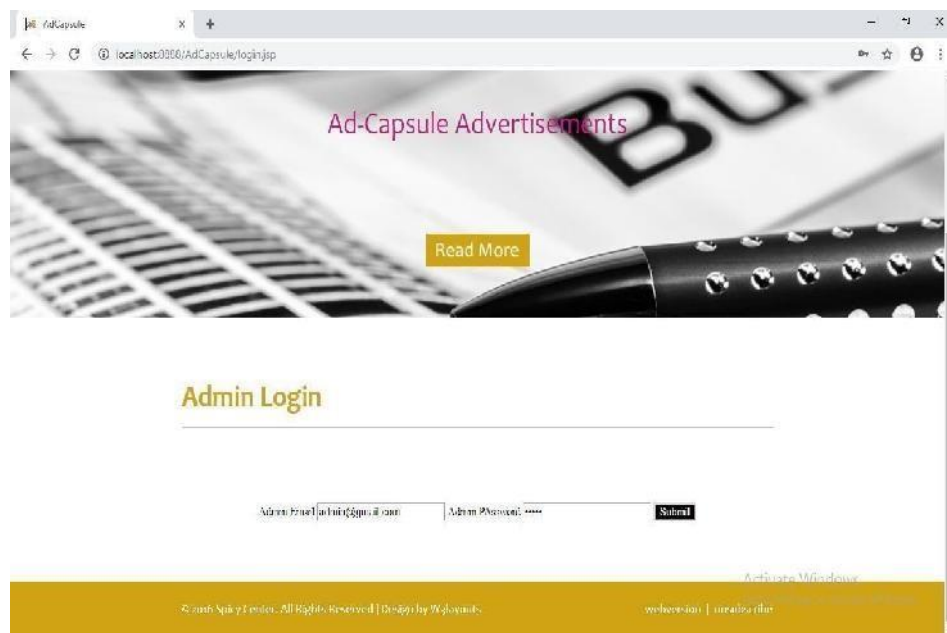
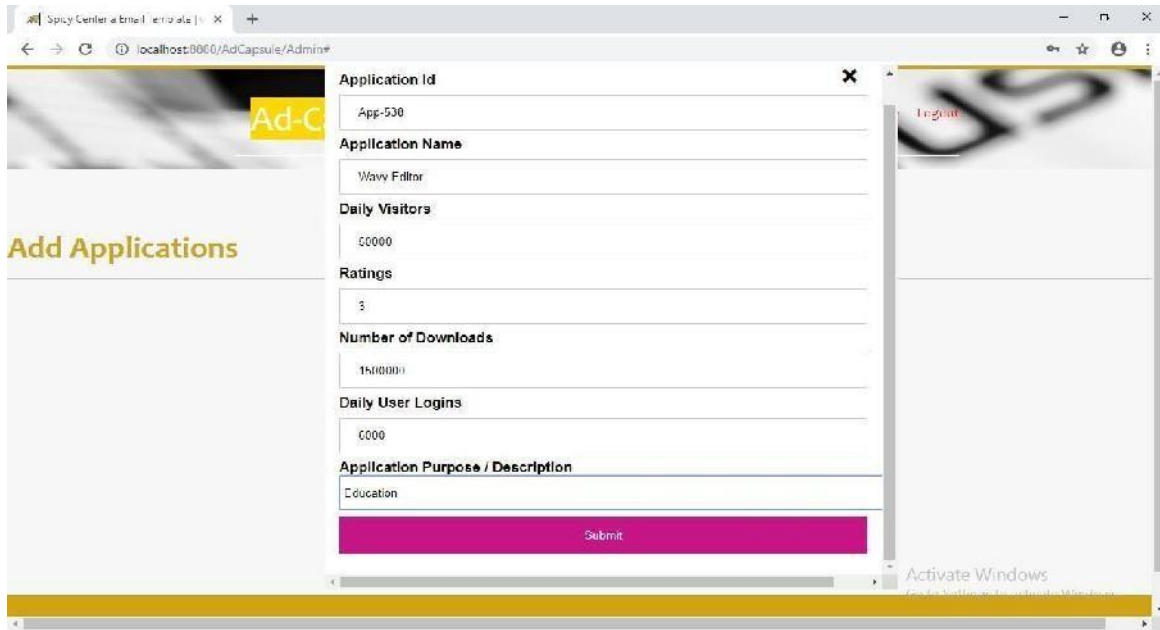


Fig 8.1 Admin login page

In-order to use our application the users need to do user registration. After Ad Agency user Authentication the users or application developers will add their applications to the Ad-Agency portal. While adding an application they has to give basic information about the app like number of downloads, number of daily user logins, ratings, and App Id is mandatory. After that when an Ad-Agency admin sign-in into the portal the requested applications for advertisements will be shown. Here the admin has the privilege to set amount

for the requested applications.



The screenshot shows a web browser window with the URL `localhost:8080/AdCapsule/Admin#`. The page has a yellow header with the text "Add Applications" and a logo "Ad-C". The main content area contains a form with the following fields:

- Application Id**: `App-538`
- Application Name**: `Wavy Editor`
- Daily Visitors**: `00000`
- Ratings**: `3`
- Number of Downloads**: `1500000`
- Daily User Logins**: `0000`
- Application Purpose / Description**: `Education`

A pink "Submit" button is located at the bottom of the form. The browser window also shows a "Login" button in the top right corner and an "Activate Windows" watermark in the bottom right corner.

Fig 8.2-Registering Application

After the Ad Provider User registers into Ad Agency, The Provider as well as Agency has to generate new bank account using our Bank application. After accounts created successfully they can do money transfer for any desired accounts. Then the Ad Provider has to sign-in the Ad-Agency page and look for apps satisfying their criteria.

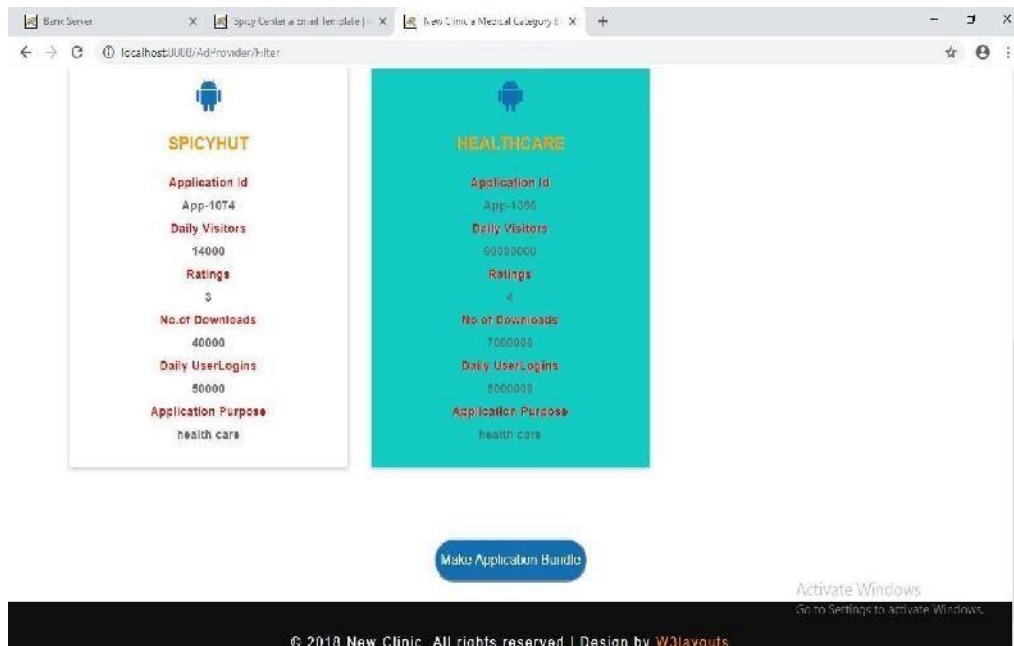


Fig 8.3-Health care app



Fig 8.4 App installed mobile

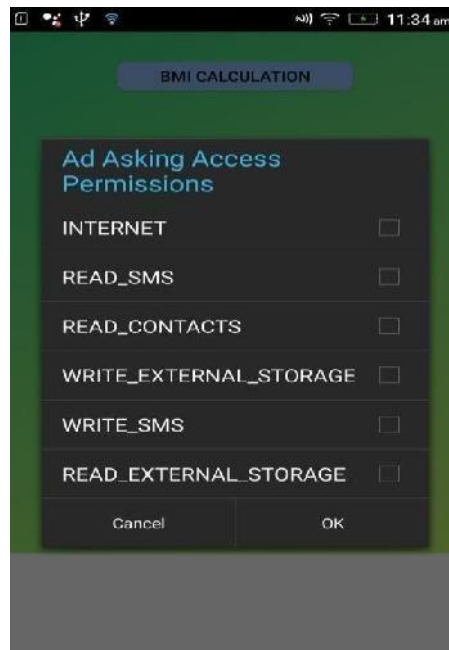


Fig 8.5 Ad asking access permissions

After bundle purchase the Provider has to upload the suitable ad content which is of types transport, education or health. After uploading the ad content into Agency, it will be automatically loaded into the end applications whose app id is there in the selected bundle. For example we run a health related app. Once an advertisement gets hosted into that application the ad content will start initiating its malicious behavior. If the application uses permission like sms or contacts the ad-content will make use of it and send all user's sms or contacts to their ad-Server automatically without user's knowledge in background.

CHAPTER 9

CONCLUSION AND FUTURE WORK

Thus we implement a Third Party Service Provider in Ad-Agency side to verify the malicious contents of incoming advertisements .So whenever a new adcomponent is loaded into the empty spaces allocated for specific ads. The ads get loaded into certain locations inside the host apps. We need to do a preprocessing technique in our proposed Third party Service Provider.So when an advertisement is loaded first into the application the Service provider will check or process all the running components using decompilation and find out the possible permission access that the ads going to get from the hosted application.

Then the Third party application will send a request to the application user in order to grant permission for authorized access. A popup will be shown to the app user and get the authorized access. Hereby we can restrict the ad components from getting access to user's sensitive data. Thus we can handle the ads from misbehaviors. This is known as confinement of advertisements.

REFERENCES

1. "Number of Android applications." <https://www.appbrain.com/stats/number-of-android-apps>.
2. "Free vs. paid Android apps." <https://www.appbrain.com/stats/free-and-paid-android-applications>.
3. "Security Alert: New Stealthy Android Spyware – Plankton – Found in Official Android Market." <https://www.csc.ncsu.edu/faculty/jiang/Plankton/>, 2011.
4. "PontiFlex Ad Library - Remote JavaScript Command Execution." <https://labs.mwrinfosecurity.com/blog/pontiflex-ad-library-remote-javascript-Cmmandexecution/>, 2013.
5. "Setting the Record Straight on Moplus SDK and the Wormhole Vulnerability." <http://blog.trendmicro.com/trendlabs-securityintelligence/setting-the-record-straight-on-moplus-sdk-and-the-wormhole-vulnerability/>, 2015.
6. S. Demetriou, W. Merrill, W. Yang, A. Zhang, and C. A. Gunter, "Free for All! Assessing User Data Exposure to Advertising Libraries on Android," in Proceedings of the 23rd Annual Symposium on Network and Distributed System Security, NDSS, 2016.
7. M. Grace, W. Zhou, X. Jiang, and A.-R. Sadeghi, "Unsafe Exposure Analysis of Mobile In-App Advertisements," in Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM Wi sec, 2012.
8. S. Son, D. Kim, and V. Shmatikov, "What Mobile Ads Know about Mobile Users," in Proceedings of the 23rd Annual Symposium on Network and Distributed System Security, NDSS, 2016.
9. N. Daswani and M. Stoppelman, "The anatomy of clickbot. a," in Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, pp. 11–11, USENIX Association, 2007.
10. A. Dewald, T. Holz, and F. C. Freiling, "Adsandbox: Sandboxing javascript to fight malicious websites," in Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1859–1864, ACM, 2010.
11. A. Zarras, A. Kapravelos, G. Stringhini, T. Holz, C. Kruegel, and G. Vigna, "The dark alleys of Madison avenue: Understanding malicious advertisements," in Proceedings

of the 2014 Conference on Internet Measurement Conference, pp. 373–380, ACM, 2014.

12. X. Dong, M. Tran, Z. Liang, and X. Jiang, “Adsentry: comprehensive and flexible confinement of javascriptbased advertisements,” in Proceedings of the 27th Annual Computer Security Applications Conference, pp. 297–306, ACM, 2011.
13. T. Book and D. S. Wallach, “An empirical study of mobile ad targeting,” CoRR, vol. abs/1502.06577, 2015.
14. S. Han, J. Jung, and D. Wetherall, “A study of thirdparty tracking by mobile apps in the wild.”
15. “How to inject JavaScript into an Android web view for a more dynamic UX.” - <http://www.techrepublic.com/article/pro-tip-inject-javascript-into-an-androidweb-view-for-a-more-dynamic-ux/>, 2014.
16. S. Poeplau, Y. Fratantonio, A. Bianchi, C. Kruegel, andG. Vigna, “Execute this! analyzing unsafe and maliciousdynamic code loading in android applications.,”in NDSS, vol. 14, pp. 23–26, 2014.

APPENDIX

A.PLAGIRISM REPORT

PaperWork.docx			
ORIGINALITY REPORT			
7%	3%	6%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	"Anomaly Detection on Android System", International Journal of Innovative Technology and Exploring Engineering, 2019 Publication	3%	
2	www.i-scholar.in Internet Source	2%	
3	Xiaonan Zhu, Jinku Li, Yajin Zhou, Jianfeng Ma. "AdCapsule: Practical Confinement of Advertisements in Android Applications", IEEE Transactions on Dependable and Secure Computing, 2018 Publication	1%	
4	azure.microsoft.com Internet Source	1%	
5	yajin.org Internet Source	1%	
Exclude quotes: On Exclude matches: Off			

B.JOURNAL PAPER

ARRESTING THE ADVERTISEMENT'S BEHAVIOR IN ANDROID APPLICATIONS

² P.Sandhya Rani
Department of computer
science and engineering,
Sathyabama institute of science
and technology, Chennai, India
sandhyapadavala5798@gmail.com

² P.Gayatri
Department of computer
science and engineering,
Sathyabama institute of
science and
technology, Chennai, India
Pajlurigavathri123@gmail.com

² Dr. L. Lakshmanan
Department of Computer Science
and Engineering, Sathyabama
Institute of science and
technology, Chennai, India
lakshmanan.cse@sathyabama.ac.in

Abstract –

Because of the multiplication of cell phones as of late there are numerous favorable circumstances and a few impediments in advanced cells utilization. Numerous applications like Ecommerce, Social media and games gain access to a client's authorization to use a few benefits. Ordinarily while introducing any application we used to concede consent to the specific application to get to all the benefits. So, when the applications which we introduce in our advanced cells have some tie-up with any notice organization. So that for the advantages of the application designer those includes will run in our application. Here right now promotions can carry on vindictively a few times. We propose an outsider Server to approve the commercial before the include get stacked into the application. We have executed a model of AdCapsule. Our assessment results show that AdCapsule can effectively uphold security arrangements to square endeavors of getting to private data or controlling documents of the host utilisation.

Keywords—Security and privacy protection, ANDROID, AdCapsule.

I. INTRODUCTION

Likewise, analysts have proposed a few arrangements to settle the security and protection worries of promotion libraries. Adcapsule is a feature for the Android platform that isolates the advertising mechanism. Adcapsule offerings api for promotion libraries & relating authorizations.

Adcapsule segregates advertisement libraries into another procedure with various client distinguishing proof. Promotion lib & host application stand 2 diverse applications, henceforth this might use android application UID component to seclude them from one another. The Frame further confines process/consent, yet additionally the sources of info and show. These frameworks may tackle the issue. Be that as it may, one genuine constraint, i.e., the adjustment to the Android structure, blocks handy arrangement of these frameworks. Clients need to streak custom ROMs to embrace these frameworks, which is about unthinkable for ordinary clients. Indeed, even for the situation that telephone merchants need to incorporate these frameworks, the notable Android fracture issue makes the arrangement to various telephone models and Android forms a tedious procedure. Interestingly, PEDAL influences bytecode changing system to rework employs authentication techniques tailored to the API in question. It isn't necessary to make any changes to the advertising library or the secret Android system, so sending it is easy. In any case, this framework is certainly not a total arrangement, and could be avoided because of its structure. To begin with, static code reworking might skirted by promotion libraries utilizing dynamic code execution

system, Present promotion repositories [3, 7] have shown this to be so. Second, sensitive details may be deduced by vindictive promotions by perusing private documentation of the host application or transparent records on SDCard, without invoking any API security[8]. Right now, propose , adcapsule a client level answer for restrict promotions in Android applications. Our answer For this function, it employs two sandboxes; the consent and record sandboxes.

We suggest two client-level sandboxes: an authorization sandbox to decouple advertising libraries' protection-related operations, and a record sandbox to restrict promotion libraries' document tasks and advertisement substance. We've developed and tested an adcapsule model. The appraisal findings demonstrate adcapsule's feasibility and similarity. Furthermore, it brings with it a plethora of benefits.

II. LITERATURE SURVEY

Son et al [1] indicated that noxious advertisement substance could induce program history by getting to reserved pictures and HTML documents on outer capacity. In particular, to decrease organize traffic, the Dolphin program reserves pictures in outside potential, and advertisement substance could utilize JavaScript to decide if a record of a site exists on the capability. Thusly, vindictive advertisement substance could decide if a specific site has been gotten to by clients utilizing the Dolphin program. This is only one of the proposed assaults appeared in the paper. Different assaults incorporate approaches to construe social diagram, sex inclinations for dating accomplices, client directions, and so on.

Grace et al [2] defined these applications, identifying 100 agent in-app promotion libraries and developing AdRisk, a platform for systematically identifying possible dangers. The outcomes indicated that most existing advertisement libraries forcefully gathered private data, including area, call

logs, telephone numbers, and so forth. Another work gathered an example of 114, 000 applications and afterward extricated and grouped the implanted advertisement libraries.

Zhang et Al[3] Proposed Aframe authorises and disengages promotion libraries from the host application. However, both of these structures requires a modification to the Android layout, which obstructs the practical scheme. Controls, approaching, and active information are all part of the Information Sluice scheme, which is targeted at each application. Unwanted advertising expectations or vexing popups are no longer a problem.

III. SYSTEM IMPLEMENTATION

A. Existing System

Privacy-conscious activities were part of the ongoing work. As a consequence, when the software we run on our devices have a commercial arrangement with a marketing agency. As a result, those ads will appear in our software for the benefit of the app maker. Advertisements in this case may sometimes be deceptive. This malicious activity harvests sensitive data such as the user's contact information and files.

A. Proposed System

We propose the TPS to monitor about advertisement library, Here User When you install a new programme, the third-party server begins scanning for new advertisements. If there is any chance of accessing the user details or files or contacts, thus an add can get unauthorized access permissions from the hosted app. Our system is capable of detecting that kind of ads and restricts their access, then pop up the required access permission list to the user. Only when the user accepts those permission it will allow the add to run inside the hosted app.



Fig.1 System Architecture

C'.EN WW

T*he modules Imlrtded In the project are:

libraries for which the Ad Broker system must file. The specialist must complete the appropriate installation process in order to obtain authorised access to the Ad Libraries. Here, the instalment scheme must be authorised access to any of the listed ads until

In-order to use our application the users need to do user registration. After Ad

ogpticinions ;W W W-Army porlnl: ¥h1le

adding an application they has to give basic information about the app like number of downloads, number of daily user logins, ratings, and App-Id is mandatory. Here the admin has the privilege to set amount for the

AD PROVIDER PURCHASING APP

After the Ad Provider User registers into Ad Agency, The Provider just as Agency

search for applications fulfilling their criteria. They can choose the applications in a pack by applying channels dependent on their kind of ad content, applications, and number of

amount, an application pack can be compared with Agency's applied offers. So now the Ad Provider needs to buy the application group.

After group buy the Provider needs to transfer the reasonable promotion content which is of types transport, training or wellbeing. Subsequent to transferring the promotion content the Agency's offers are naturally stacked into the end applications

whose application id is there in the chosen

utilization we will show a related classification Android Application. For instance we run a wellbeing related

facilitated into that application the

m<ing its pernicious' sonnet. In ihc eieiti ihei ilie apptirziun utilin

promotion substance will utilize it and send every one of client's sms or contacts to their advertisement Server naturally without client's information in foundation.

THIRD-PARTY SERVICE PROVIDER

Third-Party Service Provider to verify the noxious content of upcoming notices. So, if

empty spaces dispensed for overt ads, another adcomponent is stacked into such spaces.

Third meeting, we would need to do some preprocessing in third Providing Assistance. So when a promotion is stacked first into the application the Service supplier will check or procedure all the running segments utilizing decompilation and discover the conceivable authorization get to that the advertisements going to get from the facilitated application. At that point the Third party application will send a solicitation to the application client so as to give consent for approved access. A popup will be appeared to the application client and get the approved access. Thus we can confine the promotion parts from gaining admittance to client's touchy information. In this way we can deal with the promotions from mischievous activities. This is known as constraintment of notices.

IV.Results and Discussion

Thus we propose a third party Server to monitor the advertisement library, Here the User whenever install any new application the third party server will start checking the incoming adds. So before loading a new advertisement into the application our system will analyze the triggering components inside the ads and its behavior.

If there is any chance of accessing the user details or files or contacts, thus an add can get unauthorized access permissions from the hosted app. Our system is capable of detecting that kind of ads and restricts their access, then pop up the required access permission list to the user.



Fig.2 Admin login page- Users must complete user registration in ad agency in

order to use our app and for adding the applications.



Fig.3 Registering Application- While adding an application they has to give basic information about the app like number of downloads, number of daily user logins, ratings, and App Id is mandatory.



Fig.4 Health care app- After bundle purchase the Provider has to upload the suitable ad content which is of types transport, education or health.



Fig.5 App installed-Health care app should be installed to our mobile.



Fig.6 Ad asking access permissions- As a result, when an advertising is first loaded into the programme, the Service Provider will decompile all of the operating components to determine the potential authorization permissions that the advertisements will provide from the hosted application.

IV. Conclusion

Right now, propose a client level answer for keep notices, including promotion libraries and advertisement substance. Our framework can be delivered quickly and does not include the root benefit. It employs two sandboxes to limit security-related operations and log commercial duties. We've implemented a model of our system, and the evaluation findings demonstrate its adequacy and similarity, with minimal presentation overhead.

V. References

[1] S. Son, D. Kim, Shmatikov, "What Mobile Ads Know about Mobile Users," in Proceedings of the 23rd Annual Symposium on Network and Distributed System Security, NDSS, 2016.

[2] "Unsafe Exposure Analysis of Mobile In-App Advertisements," in Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM WiSec, 2012. M. Grace, W. Zhou, X. Jiand A.-R. Sadeghi.

[3] X. Zhang, A. Ahlawat, and W. Du, "AFRAME: Isolating Advertisements from Mobile Applications in Android," in Proceedings of the 29th Annual Computer Security Applications Conference, ACM ACSAC, 2013.

[4] A. Dewald, T. Holz, and F. C. Freiling, "Adsandbox: Sandboxing javascript to fight malicious websites," in Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1859–1864, ACM, 2010.

[5] S. Demetriou, W. Merrill, W. Yang, A. Zhang, and C. A. Gunter, "Free for All! Assessing User Data Exposure to Advertising Libraries on Android," in Proceedings of the 23rd Annual Symposium on Network and Distributed System Security, NDSS, 2016.

[6] X. Dong, M. Tran and Z. Liang., "Adsentry: comprehensive and flexible confinement of javascriptbased advertisements," ACM, 2011, Proceedings of the 27th Annual Computer Security Applications Conference, pp. 297–306.

[7] T. Book and D. S. Wallach, "An empirical study of mobile ad targeting," CoRR, vol. abs/1502.06577, 2015.

[8] "Injecting JavaScript into an Android page view for a more interactive user experience." <http://www.techrepublic.com/article/pro-tip-inject-javascript-into-an-androidweb-view-for-a-more-dynamic-ux/2014>.

[9] "Moplus SDK and the Wormhole Vulnerability: Getting the Facts Straight." Setting the record straight on Moplusdk and the Wormhole Vulnerability, Trend Micro, 2015.

C.SOURCE CODE

Applicationdetails.java

```
package logics;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.http.servlet.HttpServlet;

public class ApplicationDetails extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String appid=request.getParameter("1");
        String appname=request.getParameter("2");
        String dv=request.getParameter("3"); String
        ratings=request.getParameter("4"); String
        down=request.getParameter("5"); String
        logins=request.getParameter("6"); String
        purpose=request.getParameter("7");
        ApplicationPojo ap=new ApplicationPojo();
        ap.setAppId(appid);
        ap.setAppName(appname);
        ap.setDailyVisitors(dv);
        ap.setRatings(ratings);
        ap.setNumOfDownloads(down);
        ap.setDailyUserLogins(logins);
        ap.setAppPurpose(purpose);
        boolean result=AppDao.idCheck(appid);
        if(result!=true)
```

```

    {
    int status=AppDao.saveApplicationDetails(ap);
    if(status>0)
    {
        request.setAttribute("msg",
        "Application details are added!");
        RequestDispatcher
        rd=request.getRequestDispatcher("
        menu.jsp"); rd.forward(request,
        response);

    }
    }
    else
    {
        request.setAttribute("msg",
        "Application details are not
        added!"); RequestDispatcher
        rd=request.getRequestDispatcher("
        menu.jsp"); rd.forward(request,
        response);

    }

    request.setAttribute("msg",
    "Application id already exists!");
    RequestDispatcher
    rd=request.getRequestDispatcher("
    menu.jsp"); rd.forward(request,
    response);

    out.close();
    }
}

```

AppDao.java

```
package logics;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import logics.AppDao;
```

```

public class AppDao
{
    public static Connection getConnection(){
        Connection con=null;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost/adcapsule",      "root",
"root");
            //System.out.println("-----CLOUD USER CREATED      ");
        }catch(Exception e){System.out.println(e);}
        return con;
    }

    public static int saveApplicationDetails(ApplicationPojo ap)
    {
        int status=0;
        try
        {
            Connection con=AppDao.getConnection();
            String      query="insert      into
application(appid,appname,daily_visitors,ratings,no_of_downloads,user_logins,app_purpose,
price) values (?,?,?,?,?,?,?,?)";
            PreparedStatement ps=con.prepareStatement(query);
            ps.setString(1, ap.getAppId());
            ps.setString(2, ap.getAppName());
            ps.setString(3, ap.getDailyVisitors());
            ps.setString(4, ap.getRatings());
            ps.setString(5, ap.getNumOfDownloads());
            ps.setString(6, ap.getDailyUserLogins());
            ps.setString(7, ap.getAppPurpose());
            ps.setString(8, null);
            status=ps.executeUpdate();
            con.close();
        }
    }
}

```

```

    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return status;
}

public static List<ApplicationPojo> getApplicationData()
{
    ArrayList<ApplicationPojo> list=new ArrayList<ApplicationPojo>();
    try
    {
        Connection con=AppDao.getConnection();
        String query="select * from application";
        PreparedStatement ps=con.prepareStatement(query);
        ResultSet rs=ps.executeQuery();
        while(rs.next())
        {
            ApplicationPojo aps=new ApplicationPojo();
            aps.setAppId(rs.getString("appid"));
            aps.setAppName(rs.getString("appname"));
            aps.setDailyVisitors(rs.getString("daily_visitors"));
            aps.setRatings(rs.getString("ratings"));
            aps.setNumOfDownloads(rs.getString("no_of_downloads"));
            aps.setDailyUserLogins(rs.getString("user_logins"));
            aps.setAppPurpose(rs.getString("app_purpose"));
            aps.setPrice(rs.getString("price"));
            list.add(aps);
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

```

```
        return list;
    }
}
```

```
public static boolean idCheck(String appid)
{
    boolean result=false;
    try
    {
        Connection con=AppDao.getConnection();
        String query="select * from application where appid=?";
        PreparedStatement st=con.prepareStatement(query);
        st.setString(1, appid);
        ResultSet rs=st.executeQuery();
        if(rs.next())
        {
            result=true;
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return result;
}
```

```
public static int saveAgency(AgencyPojo app)
{
    int status=0;
    try
    {
        Connection con=AppDao.getConnection();
        String query="insert into
values (?, ?, ?, ?, ?)";
        agency(name,email,password,mobile,status)
```

```
PreparedStatement  
ps=con.prepareStatement(query);
```

```

        ps.setString(1, app.getName());
        ps.setString(2, app.getEmail());
        ps.setString(3, app.getPassword());
        ps.setString(4, app.getMobile());
        ps.setString(5, "off");
        status=ps.executeUpdate();
        con.close();
    }
    catch (Exception e) {
        e.printStackTrace();
    }

    return status;
}

public static boolean agencyCheck(String email)
{
    boolean result=false;
    try
    {
        Connection con=AppDao.getConnection();
        String query="select * from agency where email=?";
        PreparedStatement st=con.prepareStatement(query);
        st.setString(1, email);
        ResultSet rs=st.executeQuery();
        if(rs.next())
        {
            result=true;
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

        return result;
    }

    public static String agencyLogin(String email, String password)
    {
        String result="no";
        try
        {
            Connection con=AppDao.getConnection();
            String query="select * from agency where email=? and password=?";
            PreparedStatement st=con.prepareStatement(query);
            st.setString(1, email);
            st.setString(2, password);
            ResultSet rs=st.executeQuery();
            if(rs.next())
            {
                result=rs.getString("name");
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return result;
    }
}

```

```

    public static int saveAgencyPrice(String appid, String price)
    {
        int status=0;
        try
        {
            Connection con=AppDao.getConnection();
            String query="update application set price=? where appid=?";
            PreparedStatement ps=con.prepareStatement(query);

```

```

        ps.setString(1, price);
        ps.setString(2, appid);
        status=ps.executeUpdate();
        con.close();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return status;
}

public static int onOrOff(String status, String name)
{
    int result=0;
    try
    {
        Connection con=AppDao.getConnection();
        String query="update agency set status=? where name=?";
        PreparedStatement ps=con.prepareStatement(query);
        ps.setString(1, status);
        ps.setString(2, name);
        result=ps.executeUpdate();
        con.close();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return result;
}

public static String statusCheck()
{
    String res="";
    try {
        Connection con=AppDao.getConnection();

```

```
String query="select * from agency";
PreparedStatement ps=con.prepareStatement(query);
ResultSet rs=ps.executeQuery();
if (rs.next()) {
    res=rs.getString("status");
}
else {
    res="error";
}

} catch (Exception e) {
    e.printStackTrace();
}
return res;
}
}
```