IPASJ International Journal of Computer Science (IIJCS)
Web Site: http://www.ipasj.org/IIJCS/IIJCS.htm
A Publisher for Research Motivation ....
Email:editoriijcs@ipasj.org
Volume 5, Issue 8, August 2017
ISSN 2321-5992

Android Application Pen-testing Framework

Dr. Digvijaysinh Rathod

Institute of Forensic Science Gujarat Forensic Sciences University

ABSTRACT

Android OS running Smartphones are widely accepted and popular in the recent years and as the vogue of using the Android applications in the android phone are inspiring the Android developers to build verities ofAndroid applications. There are 2.8 million android applications in Android play store itself. Security pet-testing of Android mobile application is challengeable and complicated for pen-tester because of various versions of Android OS and mobile phone fragmentation. The two major problems found in Android applications were Insecure Communication over the network and Code Mitigating which includes altering some portion of code which leads certain benefits to the attacker. There are various other ways to penetrate the android applications and find innumerable vulnerabilities and bugs which might lead to critical organizational fail. I used SantokuOS which is Linux based open source operating and Genymotion to configure virtual envirnmentwith DIVA (*Damn insecure and vulnerable App*) to perform mobile application penetration testing. I elaborated broad categories of mobile application vulnerabilities and demonstrate practically vary crucial security loophole - insecure data storage, insecure communication and data leakage.

Keywords: Android, Pen testing, Vulnerability, Malware, Mobile Forensics, Mobile Security

1. INTRODUCTION

Smart Phones are fastest growing consumer technology, with worldwide unit sales is about 1.5 billion till 4th quarter of 2016 and 1.7 billion are expected until 2020 [9]. iOS and Android OS are two leading platform as far as mobile technologies concern. Android is a Linux based platform developed by Google and the open Handset Alliance. Java is an official programming language to develop android application. The Android operating system software stack consists of Java applications running on a Dalvik Virtual Machine (DVM). The current version of April 2017 is Android 7 Nougat.

Android has certain feature inbuilt as part of architecture that ensure the protection of user’s and applicationdata, protection of system resources and make sure that one application cannot access the data of another application. These security featuressome time prevent security pen-tester from gaining access to necessary data. All applications on Android run in an application sandbox, by default, an Android application can only access a limited range of system resources. The system manages Android application access to resources that, if used incorrectly or maliciously, could adversely impact the user experience, the network, or data on the device. These restrictions are implemented in a variety of different forms. Some capabilities are restricted by an intentional lack of APIs to the sensitive functionality (e.g. there is no Android API for directly manipulating the SIM card). In some instances, separation of roles provides a security measure, as with the per-application isolation of storage. In other instances, the sensitive APIs are intended for use by trusted applications and protected through a security mechanism known as permissions [10].

Mobile phones and mobile applications plays an important role in the life of everyone by simplifying messaging, emails, document sharing, collaboration, and online bankingetc,.It is a responsibilities of mobile application development firm to perform security penetration testing of mobile application because there are lots of security risks are associated with it, which may results in sensitive data leakage or security breaches. Following is the set of problems and complications faced by Android application penetration tester,

I. **OS version:** There are various versions of Android OS is available and some mobile companiescome up with customized version of Android OS as per their recruitment. Each version has different set of vulnerabilities and pen-tester needsto prepare verities test cases by keeping in mind all possible vulnerabilities.

II. **Device fragmentation problem:** As android application runs on verities of mobile devices, the task of security pen-testing of mobile application is very difficult.

2. RELATED WORK

The previous research activities were introduced with similar intensions to build the reliable framework to find the vulnerabilities and bug from android applications, the concentrated issues found from previous research were up to privacy and security within the applications.

Suyash Jadhav, Tae Oh,Young Ho Kim, Joeng Nyeo Kim describes the mobile device evaluation and testing platform to evaluate the mobile malware and to create safe environment for malware researchers and pen testers. It provides details of multiple required tools and framework to penetrate mobile security [1].

Bhabani Prasad Swain, Rahul Kumar Sinha, Keshava Murthy provides various cost-effective processes and techniques to improve efficiency of mobile application security and summarize the cyber threat on mobile applications [2].

William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri seeks for better understanding of android application by introducing the ded decompile which recovers the android application source-code from the installation image, it enumerates various possibilities of Code Mitigating [3].

Pasquale Stirparo enumerates the secure coding concepts in mobile application development; it briefs where application or user data exists, how personal data is handled in smart phones, and how can one care about security and privacy of mobile applications [4].

Research work mention above focused on Android mobile security or privacy of mobile applications or talks about cyber threat on mobile application. The key contribution of this research paper is to focus on very important vulnerability of mobile application security such as insecure data storage, insecure communication and data leakage.

3. METHODOLOGY

There exist a lot of different approaches on how a penetration test should conducted and generally as shown in Figure 1,it starts with planning, information gathering and discovery, execution and a final report. Penetration-testing is an iterative process and need to be conduct every time as and when there will be a change request in the application.

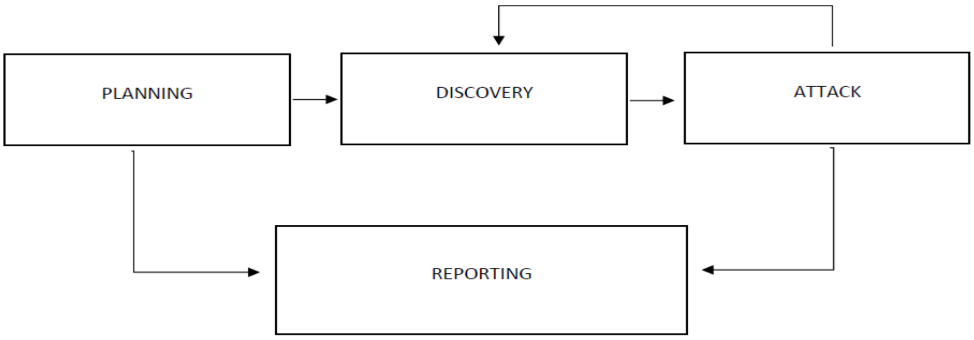


Figure 1 Iterative Process of Penetration Testing

I broadly categories the security vulnerabilities ofmobile application in four categories,

I. **Data Storage:**Need to check that data is encrypted and stored at a secure location.

II. **Authentications and session management:** Authentication is performed with user's credential at remote endpoint and after pre-defined inactivity, does sessions are terminated at the remote endpoint?

III. **Network communication:** Need to check that transmitted data is encrypted or not.

IV. **Client-side entry point:** Need to check that all the potential client-side entry points are validated and secured.

In this research paper, I discussedand demonstratedvery important mobile application security vulnerabilities such as insecure data storage, insecure communication and data leakage.

I used the following methods for evaluation:

I. **Identification of Target:** Before preparing a plan for security pen-testing, understanding the target application is very important such as its purpose, way of functioning, data need to input and navigation of activity etc,.

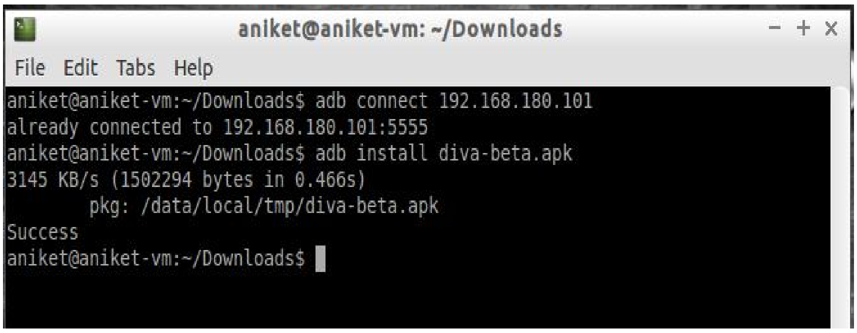
II. **Data Population:** Mobile application needs to bepopulated with required data, so proper security testing can be carried out and according test cases can be prepared.

III. **Data Acquisition:** Data acquisition is carried out with ADB(Android Debug Bridge) for analysis purpose.

IV. **Analysis:**Once data is acquired,I can do the analysis of data with the intension to find vulnerabilities.

4. CONFIGURATION

I configured the virtual laboratory with Santoku operating system and Genymotion. The Santoku is the dedicated to mobile security, analysis and forensics which is Linux based open source OS. In Santoku various inbuilt technologies for mobile security are available like Android SDK, Eclipse IDE, DroidBox, Android Emulators, etc. There are various pen testing tools are also available i.e. APK Tool, Dex2Jar, Java Decompiler, DirBuster, Nmap, Burp Suit, Metasploit Console, SQL Map and many more[11].Genymotion is an Android emulator which comprises a complete set of sensors and features in order to interact with a virtual Android environment. With Genymotion, I can test Android applications on a wide range of virtual devices for development, test and demonstration purposes. Genymotion is fast, simple to install and powerful thanks to user-friendly sensor widgets and interaction features, there are multiple APIs available in Genymotion to create a virtual device to penetrate [12].DIVA (Damn insecure and vulnerable App) is installed (Figure 2) with Santoku OS and it is target for security pen-testing. DIVA is an App intentionally designed to be insecure [13].



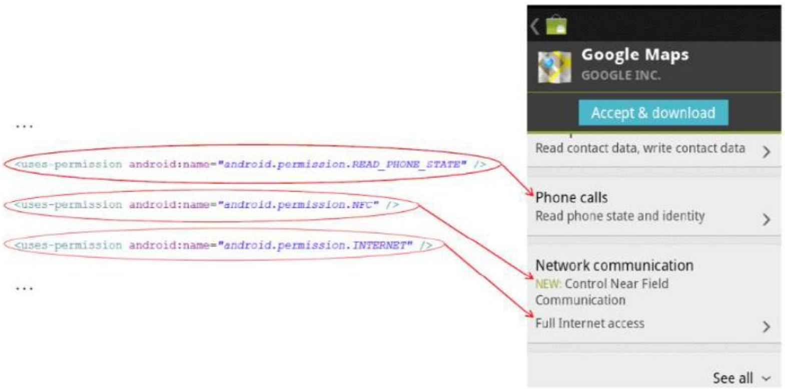
```
aniket@aniket-vm:~/Downloads$ adb connect 192.168.180.101
already connected to 192.168.180.101:5555
aniket@aniket-vm:~/Downloads$ adb install diva-beta.apk
3145 KB/s (1502294 bytes in 0.466s)
        pkg: /data/local/tmp/diva-beta.apk
Success
aniket@aniket-vm:~/Downloads$
```

Figure 2. Installing DIVA application in Emulator

5. RESULT ANALYSIS

Android application are installed from android play store which has self-signed applications, and application permission is granted on user installation time (Figure 3), an permission might be the rights given to application to access particular segment like internet, send sms, read or write external storage, etc [10].

Android Permission Model:

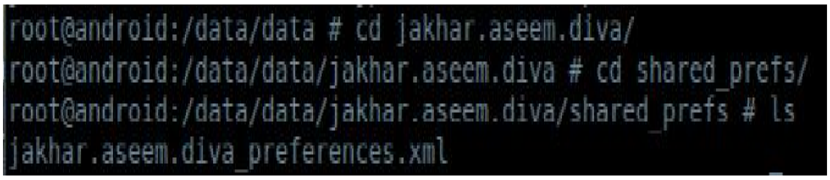


```
...
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.NFC" />
<uses-permission android:name="android.permission.INTERNET" />
...
```

Google Maps
GOOGLE INC.
Accept & download
Read contact data, write contact data
Phone calls
Read phone state and identity
Network communication
NEW: Control Near Field Communication
Full Internet access
See all

Figure 3 . Android Permission Model

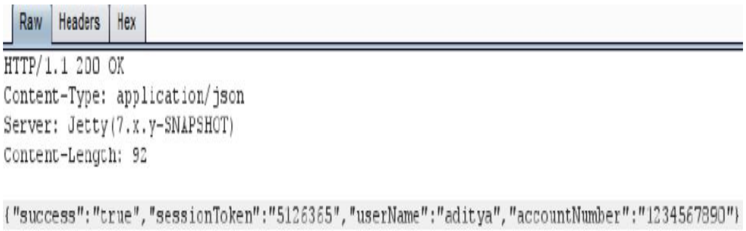
Another concern was found in android application is the insecure storage (Figure 4) of data where application data are stored in unsecuremanner; It is either stored as plain text or not properly encrypted. Various critical data related to configuration settings are found in shared_preferences which can easily viewed by the attacker. The shared_preferences file resides in data/data/packagename/shared_prefsin theXMLformat.



```
root@android:/data/data # cd jakhar.aseem.diva/
root@android:/data/data/jakhar.aseem.diva # cd shared_prefs/
root@android:/data/data/jakhar.aseem.diva/shared_prefs # ls
jakhar.aseem.diva_preferences.xml
```

Figure 4.shared_pref file

I tried to highlight some issues regarding Insecure Transmission of data and ways to exploit those issues. I took the help of tools such as Burpsuit which traces the package being sent and received, and the data in between and can be altered by using the ipaddress of the user's phone. I used the application named Goatdroid and HerdFinancial to transmit the data (Figure 5).

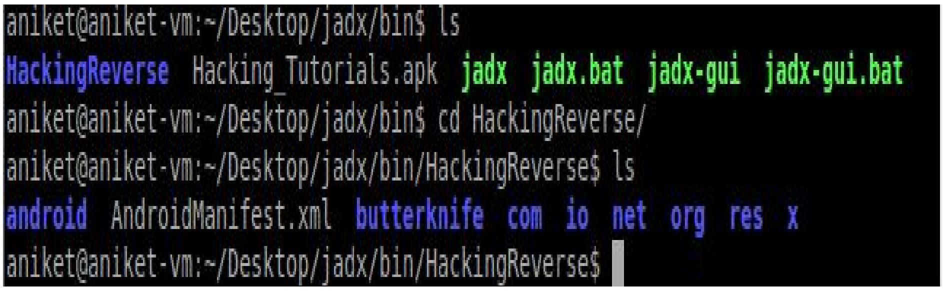


```
HTTP/1.1 200 OK
Content-Type: application/json
Server: Jetty(7.x.y-SNAPSHOT)
Content-Length: 92

{"success":"true","sessionToken":"5126365","userName":"aditya","accountNumber":"1234567890"}
```

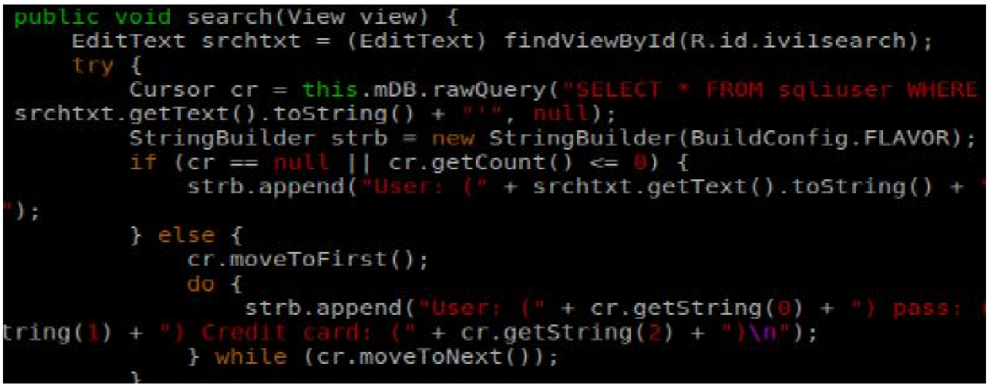
Figure 5. Burpsuit Intercepting data

As in result another most critical vulnerability would be reversing engineering (Figure 6,7) the application and tempering the code to access certain restricted functionality or to gain certain benefits. Many times information from database can be retrieved using this method.



```
aniket@aniket-vm:~/Desktop/jadx/bin$ ls
HackingReverse  Hacking_Tutorials.apk  jadx  jadx.bat  jadx-gui  jadx-gui.bat
aniket@aniket-vm:~/Desktop/jadx/bin$ cd HackingReverse/
aniket@aniket-vm:~/Desktop/jadx/bin/HackingReverse$ ls
android  AndroidManifest.xml  butterknife  com  io  net  org  res  x
aniket@aniket-vm:~/Desktop/jadx/bin/HackingReverse$
```

Figure 6. Reverse Engineering Application using jadx tool



```
public void search(View view) {
    EditText srchtxt = (EditText) findViewById(R.id.ivilsearch);
    try {
        Cursor cr = this.mDB.rawQuery("SELECT * FROM sqliuser WHERE
srchtxt.getText().toString() + "'", null);
        StringBuilder strb = new StringBuilder(BuildConfig.FLAVOR);
        if (cr == null || cr.getCount() <= 0) {
            strb.append("User: (" + srchtxt.getText().toString() + "
");
        } else {
            cr.moveToFirst();
            do {
                strb.append("User: (" + cr.getString(0) + ") pass: (
tring(1) + ") Credit card: (" + cr.getString(2) + ")\n");
            } while (cr.moveToNext());
        }
```

Figure 7. Mitigating Code of an Application

6. CONCLUSION

Android is one of the most popular and widely accepted smart mobile phone operating system and inbuilt securities of Android operating system provides security to user's sensitive data, application's sensitive data, and system resources, and protect the access of one application's data to another application data. In-built and user installed mobile application make every person's life very easy by providing facilities such as email, messaging, document sharing and online banking etc. if these mobile application will not be tested for security may result in data leakage or sensitive data loss for end user. With this serious concern,in this research I elaborate broad categories of mobile application vulnerabilities and demonstrated practicallyvary crucial security loophole - insecure data storage, insecure communication and data leakage.

References

[1]. Mobile Device Penetration Testing Framework and Platform for the Mobile Device Security by Suyash Jadhav, Tae Oh, Young Ho Kim, JoengNyeo Kim from Dept. of Information Sciences and Technologies, Dept. of Computing Security, Rochester Institute of Technology, 152 Lomb Memorial Dr., Rochester, NY, USA

[2]. Mobile Security Testing by Bhabani Prasad Swain Rahul Kumar Sinha Keshava Murthy from the Tavant Technologies.

[3]. A Study of Android Application Security by William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri from Systems and Internet Infrastructure Security Laboratory Department of Computer Science and Engineering the Pennsylvania State University.

[4]. Security and Privacy of Personal Data in Mobile Applications by Pasquale Stirparo from Mobileak.

[5]. Mobile Application Security Study 2013 report by Hewlett Packard.

[6]. Problems and solutions in mobile application testing by Triin Samuel from Software Engineering Curriculum, Institute of Computer Science, University of Tartu.

[7]. Android Environment Security by Fredrik Andersson and Gustaf Andersson from School of Computer Science, Linnaeus University.

[8]. Penetration Testing for Android Smartphones by Okolie C; Oladeji F.A; Benjamin B.C; Alakiri H. A; Olisa O. from IOSR Journal of Computer Engineering.

[9]. There are officially more mobile devices than people in the world by ZACHARY DAVIES BOREN from www.indipendent.co.uk (http://www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html).

[10]. A Permission Verification Approach for Android Mobile Applications by Dimitris Geneiatakisa,, Igor Nai Fovinoa, Ioannis Kounelisa B, and Paquale Stirparoa from Joint Research Centre (JRC), European Commission, Ispra (VA), Italy.

[11]. https://santoku-linux.com/

[12]. https://www.genymotion.com/desktop/

[13]. http://payatu.com/damn-insecure-and-vulnerable-app/