Android Smartphone Vulnerabilities : A Survey

Jignesh Joshi
M.Tech (Cyber Security)
Raksha Shakti University
Ahmedabad, India
jignesh.joshi2504@gmail.com

Chandresh Parekh
Department Of Telecommunication
Raksha Shakti University
Ahmedabad, India
rishi1745@gmail.com

***Abstract*—Round the globe mobile devices like Smartphone, PDAs & tablets are playing an essential role in every person day to day lives. Various operating systems such as android, ios, blackberry etc provides a platform for smart devices. Google's Android is a one of the most popular and user friendly open source software platform for mobile devices. Along with its convenience people are likely to download and install malicious applications developed to misguide the user which will create a security gap to penetrate for the attacker. Hackers are inclined to discover and exploit the new vulnerabilities which will bring forth with the latest version of Android. In this paper, we concentrate on examining and understanding the vulnerabilities exist in android operating system. We will also suggest a metadata model which gives the information of all the related terms required for vulnerability assessment. Here, analyzing data is extracted from Open Source Vulnerability Database (OSVDB) and National Vulnerability Database (NVD).**

Keywords— Android Vulnerability, Vulnerability Assessment, Metadata Model, NVD, OSVDB, Android Security

I. INTRODUCTION

Smart phones are the latest multipurpose gadget to which people are getting addicted. As per IDC (International Data Corporation) survey Android dominated the smart phone market with a share of 82.8% in 2015. Android platform is open source based on linux kernel and it provides developers huge opportunity to develop, sell and distribute applications. To develop an application, security developers use different types of SDK tools like Android SDK manager, AVD manager and many more using Java programming language. As we know that Android architecture is categorized into five parts: the Linux Kernel, System Libraries, Android Runtime, Application Framework and Application layer [1].

In today's cyber world, cyber attacks are going on everywhere. Security is major concern for this crucial environment. Recently, Android version 6.0 named Marshmallow has been released in market. This version runs on Linux kernel version 3.18 which handles security management for the operating system. As per security aspect concern we focus android security architecture.

Android security architecture represents the security issue associated with android applications. To isolate apps data and codes from other apps the sandbox concept is used. Android does this by giving each app a unique user id (a UID). This concept is also known as privilege separation. It gives the surety that one application does not have permission to access other applications. Android application structure and content provider concept should be discussed for understanding the android security architecture [2].

- Application Structure

Android device application is developed using Java programming language. This program files are in compressed form which contains resources, source program, data files required to run this kind of programs on Android platform. Java source program has been compiled to make classes.dex file. In res folder, resources are available such as images, strings, and User Interface screens. This resource related information is kept in resources.arsc file. The electronic signature files have been holding by META-INF. SHA1 Hash algorithm is used by electronic signature. Developed program is going to be installed in smart device after signing by private key of developer. The most important file named AndroidMenifest.xml which contains the essential information of application such as name of application, permission, activities, Android API version, service and content provider information. Before the execution of app this .xml file should be notified to Android system [3].

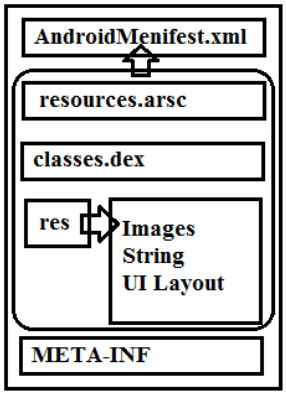

Fig.1. Application Structure

978-1-5090-0673-1/16/$31.00 ©2016 IEEE

➢ Permission and Content Provider

On android device, to perform database operation such as storing, retrieving and manipulating data from the database, SQLite database is used. Every app has their own location and structure and based on that they can access their database. One app can't access the database of other apps. Content provider concept which lies on client server model offered by android for accessing other apps databases. To share the databases of server apps should make use of Content Providers. URIs is used for identification of databases externally by server apps. The client apps that are intended to use the database, and thus as database identifier the server allocates URI in AndroidManifest.xml or Content Provider which will make request to transmit queries to server apps through given URI. So that client apps can be communicated with server apps database through URI identifier.

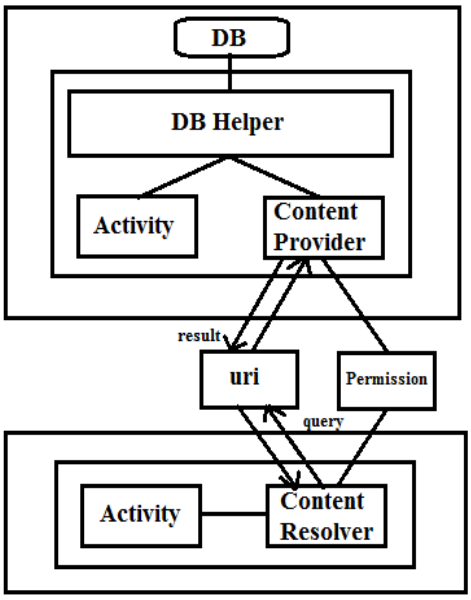

Fig.2. Database sharing between apps

Content Resolvers is used at client side and Content Provider is at server side to communicate with server database by client apps. For that, client apps send queries and URIs to Content Provider through Content Resolvers. Query results are given to Content Provider by database through DB Helper. These results send back to Content Resolver at client side. Sharing databases between server and client apps is possible because of Client Resolver and Client Provider. The whole process and structure diagram are depicted in Fig.2.

When apps required major system resources or functions then they need to send request for permission of Android system through AndroidMenifest.xml file as shown in Fig.3. For example, to access the internet, an application needs to acquire the android.permission.INTERNET permission from Android system. Android system enables the developers to use their own permissions over 200 permissions. Server applications defined in AndroidMenifest.xml file use permission to bound accesses, permission for reading and writing to share their databases. Client apps request to the system for taking the permission to access the database assigned in AndroidMenifest.xml file. Permissions are verified for accessing particular databases when client apps run by sending queries to Content Provider from Content Resolvers. Service is denied if the client apps do not own required permission to access those databases [3].

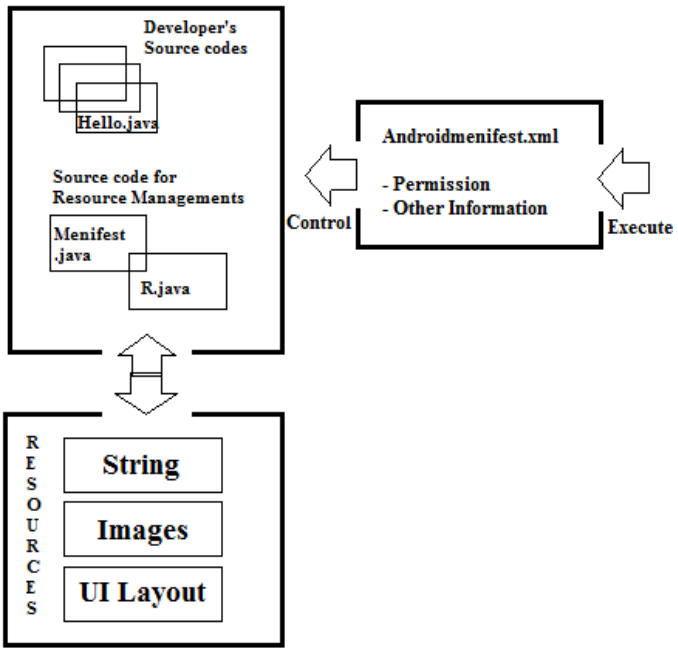

Fig.3. Relationship between apps and permissions

II. VULNERABILITY SCENERIO

In the mobile world the popularity and market share of android is continuously increases. As the number of always-on, always-connected Smartphone increase, so does the amount of personal and sensitive information they collect and transmit. Thus it becomes critical to secure these mobile devices from the so-called cyber attack. Smart devices including android mobiles and tablets are subject to vulnerabilities across the globe. We hereby dynamically analyze the vulnerabilities which exists in the android smart devices especially Android. National Vulnerability Database (NVD) and the Open Sourced Vulnerability Database (OSVDB) are the databases project came up to provide us the gathered and revealed latest detected software vulnerabilities [4].

The website named http://www.cvedetails.com lists all the security vulnerabilities present in the world. Here, we are introducing a table in which total no. of vulnerabilities present in android smart devices by every year are listed. The data given in table is analyzed uptil January 2016 [5].

TABLE I. VULNERABILITY PROGRESSION OVER TIME

Year	# Of Vulnerabilities
2009	5
2010	1
2011	9
2012	8
2013	7
2014	11
2015	130
2016	13

According to the survey all over world there are different kinds of vulnerabilities with which one can exploit the android devices. Some of the vulnerabilities are discussed below:

A. Code Execution

Without given any permission malicious code is going to be run in user's device by attacker. This vulnerability technique is known as code execution which is more terrible. After 18 month this patch was fixed.

B. Denial of Service

. Around 95% of Android devices are affected by DoS vulnerability. DoS vulnerability can be exploited either by a malicious app installed on the affected device, or through a specially crafted website. In this vulnerability attack, service continues to loop even if the malicious app is terminated until system resources aren't depleted. DoS attack is also possible when Wi-Fi scanning process is started.

C. Overflow

This type of vulnerability in android can make the phone apparently dead- unable to make calls, silent, , with a lifeless screen.

D. Memory Corruption

Memory exploitation bugs affect native Android executables even in the midst of protections such as ASLR, Stack Guard, and SE Linux.

E. SQL Injection

SQL Injection attack is very much popular & prominent attack in android devices. Database is used in Android device is SQLite. In this attack, malicious code is inserted into strings that are later passed to an instance of SQLite for execution.

F. XSS

XSS vulnerability can be used to gain access and stealing credentials by injecting JavaScript into web pages loaded in browsers like Opera mini, Google chrome etc. It is also known as Cross Application Scripting.

G. Directory Traversal

In the context of a user application, Directory Traversal can allow attackers to perpetrate a path traversal attack for accessing read/write files and also can view restricted files inside internal storage.

I. Http Response Splitting

Http response splitting arises when data penetrates into web applications through entrusted source like http request. In that an attacker passes malicious data to a vulnerable application.

J. Bypass Something

To get complete access of the device the password protected lock screen can be bypassed. In this exploit, the device will be buffered too much and thus choked in by entering number of characters into the password field by the intruder.

K. Gain Information/Gain Privilege

Android device fails to perform adequate boundary checks user-supplied data. So that attackers can exploit this vulnerability by executing arbitrary code within the context of the affected device for getting credentials & personal information of the user.

L. Cross Site Request Forgery (CSRF)

Web browser applications store cookies which are used to maintain credentials of the user. In this attack, the user is tricked by submitting forged request using the cookies which store credentials associated with the browser applications.

M. File Inclusion

File inclusion vulnerability allows an attacker to include pre-created file, usually through a script on the web server at the time of application installation.

III. METADATA MODEL

For Mobile Vulnerability Market Analysis, we suggest metadata model as shown in below figure. In that, there are four modules [6].

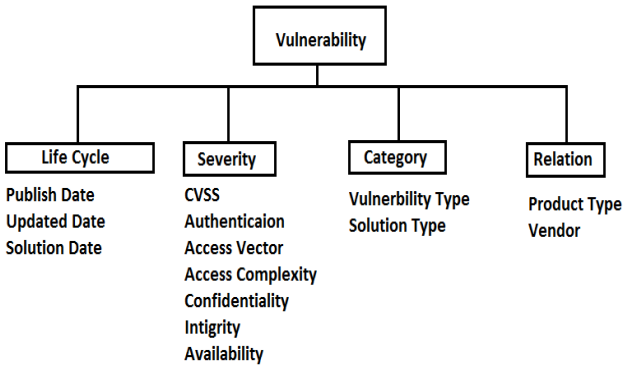

Fig.4. Metadata Model

- Life Cycle
  A. Publish Date
     The date in vulnerability is collected by the platform and disclosed to the public.
  B. Updated Date
     The date the vulnerability was last modified or repaired by the service.
  C. Solution Date
     The date on patching for the vulnerability is released by the vendor.

- Severity
  The risk level of any kind of vulnerability is known as severity level. This severity level is based on self-calculated CVSS score for specific vulnerability.

  - CVSS
    Common Vulnerability Scoring System is a vulnerability scoring system designed to provide end users with an overall composite score representing the severity and risk of a vulnerability. CVSS is a real number between 0 and 10 in which level 10 being the most severe.
  - Authentication
    In order to exploit the vulnerability, express the number of times attacker commit authenticate to destination target.
  - Access Vector
    Illustrate the approach to exploit the vulnerability.
  - Access Complexity
    The level of complexity depicts to attack the software through vulnerability.
  - Confidentiality Impact Ratio
    The impact ratio on confidentiality of a successful exploitation is defined.
  - Integrity Impact Ratio
    The impact ratio on integrity of a successful exploitation is defined.
  - Availability Impact Ratio
    The impact ratio on availability of a successful exploitation is defined.

The information about discovered vulnerabilities can get from different kinds of vulnerability databases such as the National Vulnerability Database U.S (NVD) [7] and Open Source Vulnerability Database (OSVDB) [8] which consist of huge bunch of identified vulnerability. The unique identifiers are used for publicly known security vulnerabilities are known as Common Vulnerabilities and Exposure Identification (CVE-ID) [9]. CVE-ID can be assign with year (4 digit) plus arbitrary digits (4-6 digit). Another Common Weakness Enumeration Identification (CWE-ID) provides a common language of discourse for discussing, finding and dealing with the causes of security vulnerabilities as they are found weaknesses in source code, design, or system architecture. This identification system is maintained by American organization name MITRE Corporation. Using different kinds of website, we extract the relevant information of particular vulnerability like CVE-ID, CWE-ID, vulnerability type, publish date, update date, score and other affected factor.

IV. PROPOSED ARCHITECTURE

Most of the attacks are performed on devices due to malicious application. These applications are vulnerable because they are mostly downloaded from third party sources. There are numerous vulnerabilities present in the previous and latest versions of android applications. So, to work on all the vulnerabilities is quite tedious and expensive task. Concentrating on a single vulnerability present in latest android versions, we will try to find out that vulnerability using application testing process. For that, we convert .apk file into java file using reverse engineering process [10].

My Proposed architecture [11] as shown in figure 5 suggests working on latest known vulnerability present in android application and thus providing secured android application. This architecture illustrates the process of application testing to find out vulnerability present in the application and fix it. After that we are going to generate keystore for authentication. Then, we again rebuild the application and thus verify it by signing it through authentication certificate provided by authorized entity which will allow us to upload the rebuilt application in hosting servers.

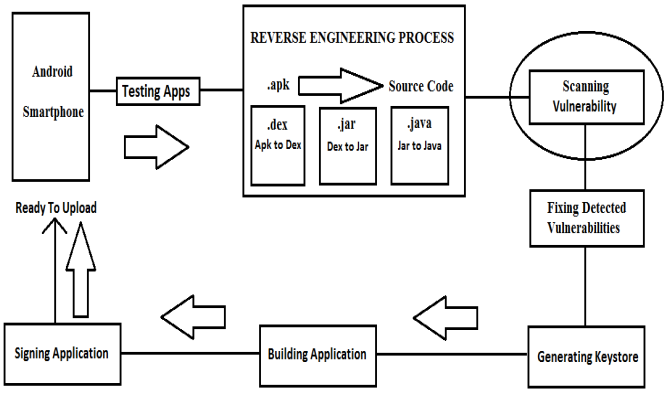

Fig.5. Android Application Vulnerability Scanning Tool

V. FUTURE WORK

A Mobile and android devices particularly are not designed with the privacy in mind. In fact they are designed to make it user friendly and easy for third parties, telecommunications companies, adversaries and even malicious minds to gain all manner of personal information from the user. Even worse by default users are unaware about what information is being collected and how. Users download various malicious apps from the app stores which contains bugs or malware which will negatively affect your device and personal information. To avoid such consequences the security platform which must offer us to scan the vulnerabilities present in any smart devices.

We will make one application vulnerability scanning tool for detecting the vulnerabilities from the application and try to fix it accordingly.

VI. CONCLUSION

Android smart phone environment has been playing an essential role in users' day-to-day life. Along with that security also play an important role in the latest susceptible corporate environment. We have introduced here a metadata model for understanding the relevant terms of android vulnerability like discovered date, solution date and severity effect of that particular vulnerability. This vulnerability data gathered from various vulnerability database sources which will significantly provide us the details about the vulnerability characteristics and its impact on the security aspects such as permissions, privileges, and access control. The research here intends to study about application testing and make a patching management system for fixing the vulnerabilities. The goal is to identify the vulnerabilities for new released version and to fix that vulnerabilities present in application by using application testing technique for improving the security of the mobile application.

REFERENCES

[1] Himanshu Shewale, Sameer Patil, Vaibhav Deshmukh and Pragya Singh "Analysis Of Android Vulnerabilities And Modern Exploitation Techniques," Ictact Journal On Communication Technology, March 2014, Volume: 05, Issue: 01.

[2] Ankita Khandelwal, A K Mohapatra "An Insight into the Security Issues and Their Solutions for Android Phones," 2nd International Conference on Computing for Sustainable Global Development (INDIACom) IEEE 2015.

[3] P. D. Meshram, Dr. R.C. Thool, "A Survey Paper on Vulnerabilities in Android OS and Security of Android Devices," Global Conference on Wireless Computing and Networking (GCWCN) IEEE 2014.

[4] Eko Sugiono, Yudistira Asnar, Inggriani Liem, "Android Security Assessment Based on Reported Vulnerability" 2014 IEEE.

[5] Android Vulnerability Statistics, http://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224

[6] Keman Huang, Jia Zhang, Wei Tan, Zhiyong Feng, "An Empirical Analysis of Contemporary Android Mobile Vulnerability Market" International Conference on Mobile Services IEEE 2015.

[7] Android Vulnerability NVD Results https://web.nvd.nist.gov/view/vuln/search-results?adv_search=true&cpe=cpe%3A%2Fo%3Agoogle%3Aandroid

[8] OSVDB, https://blog.osvdb.org/category/vulnerability-databases/

[9] CVE-ID Syntex, https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures

[10] Darko Hrestak, Stjepan Picek, Željko Rumenjak, "Improving the Android Smartphone Security against Various Malware Threats" MIPRO 2015, 25-29 May 2015, Opatija, Croatia.

[11] R.Dhaya, M.Poongodj, "Detecting Software Vulnerabilities in Android Using Static Analysis" 2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT).