The International Journal Of Engineering And Science (IJES)
|| Volume || 3 ||Issue|| 4 || Pages || 59-63 || 2014 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

# Malware Classification using Naïve Bayes Classifier for Android OS

[1,] Deepak Koundel, [2,] Suraj Ithape, [3,] Vishakha Khobaragade, [4,] Rajat Jain
*B.E. Computer Science JSPM's JSCOE Pune, India*

**----------ABSTRACT----------**
*In the global world of mobile technology millions of users connect and share on unknown networks without being aware of vulnerability of their confidentiality. Android platform is most popular OS among the smart phones users as well as developers, its open and flexible nature allows a large community to upload and download applications. Such extensive usage makes it an easy target for attack and misuse. A malicious application may steal the confidential data of user and upload it on its server, which is a threat to user's security. In this paper, we propose an approach to classify an application as malware or benign app by using data mining. To categorize an application we use various attributes of an app:(i) the permissions used by an application, (ii) battery usage rating based on permissions and (iii)rating acquired by the application on Android market. We apply Naive Bayes classifier to deduce the results based on the probability of an application being malware or not. These results are uploaded on the cloud where a user can view the results and query an application as being malicious or not to our server.*

**KEYWORDS:** *Android, Malware Detection, Data mining*

---

Date of Submission: 21 March 2014 Date of Publication: 15 April 2014

---

## I. INTRODUCTION

In the era of mobile technology smart phones have become essential part of our life, starting from being a personal assistant for organizing our daily work to adding entertainment in our lives smart phones play a crucial role. As of today many operating systems are available for smart phones. Android OS and iOS dominate the the smart phone world largely, these two OS together had 92.3% of all shipments done for smart phones[1] and Android held 75.0% of market share in smart phone world during the first quarter of 2013, according to IDC[1].

Android is the most widely used and popular OS among the users, which makes it a prominent target for the malware attacks. Malware are malicious applications which intentionally interfere with the system's functionality and causes damage to the software and security of the system. A malware application can send the personal and secret information of the user to an untrustworthy third party. It can cause the system or other applications to behave in an unexpected or malicious way. Unlike other platforms, Android maintains openness and doesn't put much restriction on its users in downloading and uploading apps. This attracts a huge community of developers as well users towards this platform .Unlike apple where app store is the only source of applications, Android allows the users to download apps from third party market. A malware author downloads a legitimate app, repackages it with malware and distributes such app on third party market and websites [2]. Android on its side leaves the security of device in user's hand by letting him take the decision of whether to install an app or not. Unfortunately, due to lack of security awareness and knowledge about Android permissions user is not the right person to judge the intention of an application.

These factors put user in a vulnerable situation where his confidentiality is at risk. To resolve this situation, in this paper we have proposed an approach that classifies the applications in two categories as: 1) malware app and 2) legitimate app using data mining. We make use of Naïve Bayes classifier which is a probabilistic classifier and uses Bayes theorem. It takes parameters such as permissions, battery usage rating and user rating of an application and generates results depending on the probability values. These values are then classified by the classifier and the final results about the status of an application are stored on cloud where user can retrieve them.

Rest of the paper is organized as follows: Section II describes the literature survey and a brief explanation of Naïve Bayes classifier. Section III explains our solution architecture and working of Bayesian classifier with the data set. We have also described the feature extraction method here which is used for gathering input set from the user's phone. Finally in section IV concludes our work.

## II. RELATED WORK

### *A. Android permissions*

Android permissions are an essential part of an application without which no Android app can be considered complete. Every app declares its permissions to user at the time of installation. Android enforces each app to perform only those functions that it has requested and declared. But, no matter how complying this permission model seems, it has some flaws. Using the permissions an app can perform those things in background which we would not have permitted it to do voluntarily. For example, a gaming app requests permissions as android.permission.READ_CONTACT and android.permission.INTERNET, then possibility is that it may read the device's contact and send the data to third party server over internet for advertising purpose. If an app declares android.permission.SEND_SMS, this could allow the app to send message on your behalf and cost you money by sending SMS to for-pay numbers [3].

```
<?xml version="1.0" encoding="utf-
8"?>
<manifest>
<uses-permission />
<permission />
<permission-tree />
<permission-group />
<instrumentation />
<uses-sdk />
<application>
<activity>
<intent-filter>
<action />
<category />
<data />
<activity-alias>
<intent-filter>…</intent-filter>
<meta-data />
</activity-alias>
<service>
<intent-filter>…</intent-filter>
<meta-data/>
</service>
<uses-library />
<uses-configuration />
</application>
</manifest>
```

Figure I. Structure of manisfest file

### *B. Naïve Bayes Classifier*

Bayesian classifier is a supervised learning technique that allow us determine uncertainty of a model by determining probabilities of interdependent events. It is widely used in diagnostic and predictive problems where number of input parameters is very large [4]. It is based on Bayesian algorithm which tries to maximize the probability of an attribute in belonging to a classified category depending on existing training data set. In other words, predicting a suitable class for a tuple based on conditional probabilities of existing data set.

## III. PROPOSED THEME

The main objective of our approach is developing an Android application which enables the user to make a check of the applications installed in his phone. Our app allows user to send a list of applications for analysis and we classify the list of apps sent as malware or legitimate app.

### *A. Basic Architecture*

The basic architecture of our system is shown in Fig II. Our system follows client server architecture where users with smart phones act as clients and our server is setup in cloud. User initiates the process by making a list of applications to be tested on his phone. Then permissions specified in the manifest file of the app are extracted and sent to the server. Our app also calculates a feature called battery usage rating depending on permissions specified in manifest file. If an app uses permissions that drain phone's battery we mark such app with high rating of battery usage. All these features are combined together and sent to our server. The server then parses this file and prepares a .csv file. The .csv file contains the organised dataset having necessary parameters on which the Naïve Bayes classifier can be applied.

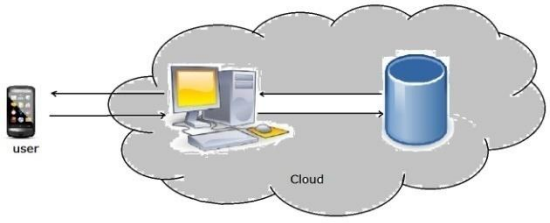


Figure II. Basic Architecture

### *B. Features Extraction*

Retrieving features for preparing a dataset is the most important module of our system. This work is done by our application on client side. Every application contains a *AndroidManifest.xml* file (shown in Fig I) and this file contains necessary parameters required by our system to perform data analysis. To decrypt the application file for extracting features we use the aapt tool (Android Asset Packaging Tool) [5] provided in Android SDK. We extract the following parameters:

- Permissions: the permissions required by an app are identified under <*uses-permission*> tag.
- Name of app: all features extracted are identified by the name of application.
  The name of app can be found at string (package="com.package_name.app_name").
- Battery rating: if an app is consuming or draining phone's battery then it might be doing some heavy task in background that probably requires internet. If an application asks the permission for using internet and GPS of phone then we give the battery usage rating as high else we give low rating.
- User rating: the rating that an application got from users in Android market.

These features are combined together in an object file and sent to server. Finally, on the server side this file is imported and the dataset is prepared ready for Bayes classification. Fig 3 shows the flow of work in features extraction.

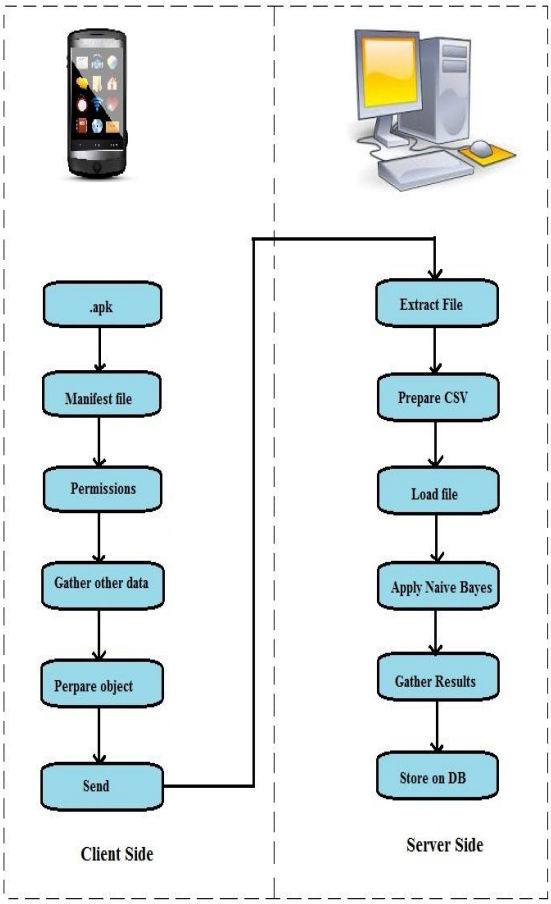


Figure II. Flow of feature extraction

### *C. Applying the Classifier*

The data set consists of various parameters like app name, permissions required by the app, battery rating, etc, shown in Fig 4 Permission can take value as:

- 1 if that permission is found in app
- 0 if that permission is not found in app

Similarly battery rating takes values as low(0), medium(1), high(2).The Bayesian classifier will work as follows:

1. The user sends a query for application testing which is treated as a vector [4] containing a set of n number of parameters. Each vector is a row where each parameter is represented as a column. The query is handled by the Naïve Bayes classifier, which operates on the parameters given in vector. Then according to Bayes theorem, P(CLASS|VECTOR) is proportional to P(VECTOR|CLASS)*P(CLASS).

   Here P(CLASS|VECTOR) is the posterior probability and P(CLASS) is the prior probability.
2. CLASS has two categories MALWARE and LEGITIMATE in which an app can be classified. Given a vector, the classifier will decide that the app belongs to which category on the basis of higher posterior probability, i. e. the Naïve Bayes classifier decides that the vector belongs to the class(MALWARE) if,

   P(MALWARE|VECTOR) > P(LEGITIMATE|VECTOR)
   or belongs to class(MALWARE) if,

   P(LEGITIMATE|VECTOR) > P(MALWARE |VECTOR).

Naïve Bayes works on conditional probability of interdependent parameters. Therefore we calculate conditional probability for individual parameter as,

(Number of malware having that parameter)/ (total number of malwares in dataset), this is repeated for all parameters and all probabilities are multiplied with each other. Same is done for legitimate apps in the data set.

## IV. CONCLUSION

We prepared our own database of malware applications and good legitimate applications. For malware data set we used open source data source of android malwares by androguard [7]. We organized the data to create training data set and testing data set for malware as well as benign apps. After applying the Bayesian classifier on the testing data set we got satisfactory results. However, results were not fully accurate when tested for random real world data. Hence in this system we observed that accuracy of Naïve Bayes classification depends on the quantity of data set.

| **App_ name** | **Permission_ CALL_PHONE** | **Permission_ INTERNET** | **Permission_ READ_SMS** | **Permission_ READ_CONTACTS** | **Permission_ ACCESS_GPS** | **Battery_ Usage** | **Output** |
|---|---|---|---|---|---|---|---|
| BBM | Yes | Yes | Yes | Yes | No | Medium | Not Malware |
| Baseball Superstars 2010 | Yes | Yes | Yes | Yes | Yes | High | Malware |
| MX Player | No | Yes | No | No | No | Low | Not Malware |
| My Tracks | Yes | Yes | No | Yes | Yes | High | Not Malware |
| Dragon Ball Wallpapers | No | Yes | No | No | Yes | High | Malware |

TABLE I. A sample component from our database set.

## REFERENCES


[1] International Data Corporation, http://www.idc.com/getdoc.jsp?containerId=prUS24676414

[2] Rafael Fedler, Christian Banse, Christoph Krauss, and Volker Fusenig, "Android OS Security: Risks and Limitations", Fraunhofer Research Institution for Applied and Integrated Security, May 2012.

[3] Zarni Aung, Win Zaw, "Permission-Based Android Malware Detection", International Journal Of Scientific & Technology Research Volume 2, Issue 3, March 2013.

[4] Mrs.G.Subbalakshmi, Mr. K. Ramesh, Mr. M. Chinna Rao, "Decision Support in Heart Disease Prediction System using Naive Bayes", Indian Journal of Computer Science and Engineering, Vol. 2 No. 2 Apr-May 2011.

[5] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero and Pablo Garcia Bringas, "On the Automatic Categorisation of Android Applications", Consumer Communications and Networking Conference (CCNC), 2012 IEEE.

[6] Webpage: "Android permissions explained, security tips, and avoiding malware", http://androidforums.com/android-applications/36936-android-permissions-explained-security-tips-avoiding-malware.html

[7] Webpage: androguard- Reverse engineering, Malware and goodware analysis of Android applications and more (ninja !) https://code.google.com/p/androguard/wiki/DatabaseAndroidMalwares#Malware_detection.

[8] Webpage: Using Permissions, http://developer.android.com/guide/topics/security/permissions.html.

[9] Webpage : "Android application fundamentals," http://developer.android.com/guide/components/fundamentals.html.

[10] Adrienne Porter Felt, Kate Greenwood, David Wagner, "The Effectiveness of Application Permissions", Proceeding WebApps'11 Proceedings of the 2nd USENIX conference on Web application development, 2011.

[11] Han, J., Kamber, M,:" Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, 2006.

[12] G. F. Cooper and E.Herskovits, "A bayesian method for constructing bayesian belief networks from databases", in proceedings of 1991 conference on uncertainity in artificial intelligence, 1991.

[13] Dhaval Desai "Malware Analysis Report", Kind Sight security labs, October 2011.

[14] Webpage: Malware Alert, Mobile Security, http://appview.mobilesecurity.com/app/473517/Dragon-Ball-Wallpapers#security.

[15] Webpage: Take a sample, leave a sample. Mobile malware mini-dump - July 8 Update http://contagiodump.blogspot.in/2011/03/take-sample-leave-sample-mobile-malware.html.