Enhancing RansomwareElite App for Detection of Ransomware in Android Applications

Shivangi*, Gautam Sharma†, Anubhav Johri‡, Akshita§, Anurag Goel¶ and Anuradha Gupta‖
Department of Computer Science
Jaypee Institute of Information Technology
Noida, India
Email: *shivangigupta19dec@gmail.com, †gautam.kashyap97@gmail.com, ‡anubhavj22@gmail.com, §akshitag1997@gmail.com
¶anurag.goel@jiit.ac.in and ‖anuradha.gupta@jiit.ac.in

***Abstract*—As the number of android applications (apps) available in the market are increasing rapidly, various types of security attacks using the android apps are also increasing with the same pace. The ransomware attack is one of these kind of security attacks in which the attackers locks the user's phone, encrypts user's data or blocks the user's access to their own data and threatens the user to pay a ransom to gain the access back. This cyber-threat is terrorizing the world from many years as it performs mimicry attacks i.e. combination of encryption & locking attacks. Android devices are more prone to these ransomware attacks compared to Windows and IOS devices. RansomwareElite is an android application which detects the presence of ransomware in the apps installed on an android device by checking the presence of any threatening text in app code or by verifying the permissions requested by the app from the user. In this paper, we focused on improving the performance of RansomwareElite app by extending its features. Now, the RansomwareElite app also searches the presence of any threatening image or file containing threatening text by analyzing the Android Package Kit(APK) file of android app. Moreover, it also detects some specific methods and classes in the code of the APK which could be used for locking the device and checks some specific permissions requested, for uninstalled apps now. Further, it maintains a database on the online server for the records of all the suspicious and ransomware apps detected by RansomwareElite. We have tested RansomwareElite with 9 Test Apps which are manually created based on the features of ransomware family and on 48 android devices. After analyzing the test results, we have found that the performance of RansomwareElite is improved after incorporating the new features and RansomwareElite app detects the presence of ransomware in installed as well as uninstalled apps present on an android's device in an efficient manner.**

***Index Terms*—Android; API; Text Classifier; Ransomware;**

I. Introduction

In today's world, mobile phone is not just remain a device used for calling and sending messages, it has been used for many other purposes e.g. bill payments, online recharges, buying products from e-commerce sites, doing banking transactions etc. This has become possible because of the huge number of mobile apps available in the market. Android operating system covers more than two-third population of the smart phone users. As the android users are increasing rapidly and huge number of android apps is available providing different functionalities to users, thus android operating system remains the major target for the hackers [1,5,6,7]. The web criminals generally attack the smart phone devices by installing some kind of malware during the apps installation. A survey, conducted by F-Secure have shown the results that an android malware which had come from the Google play store contributes only 0.5% [3]. This means if android app installed from trusted party, possibility of malware installed along with the original app is rare. This paper is focused on one of malware ransomware. Ransomware attacks an android device in the form of malicious software. It attacks by locking the screen or encrypting files on the device and later demand to pay a ransom to get the decryption key or unlock the device. The ransom could be paid in the form of bit-coins, vouchers, iTunes, Amazon gift cards and fake legal fines. It has been observed that in the last 4-5 years, the attacks by the ransomware on android devices are growing exponentially [2] . Ransomware can be categorized into crypto ransomware, which encrypts specific files and locker ransomware which locks the entire device[9]. The payment of ransom does not guarantee the complete recovery of data [4]. This created an urge to find measures to this cyber issue.

This paper proposed enhancements in RansomwareElite app [12], an android app which detects the presence of ransomware on an android device. The newly added features make the RansomwareElite app capable of detecting the ransomware even in the uninstalled apps which are residing on the device with the help of their APK. APK is package file format used by the android operating system for distribution and installation of mobile apps and middleware [8]. This file contains all the meta information of the android application such as manifest file, certificate of application, list of resources, application assets, the classes compiled in the dex format, etc in a non user readable format. The early detection of ransomware in uninstalled apps on device, prevents the user from installing the suspicious apps on their device which may lead to ransomware attack later. Another features are added like to detect the presence of threaten text in image, find the function or classes to lock the system and create a database of the suspicious and ransomware app.

II. Background

There are currently two kinds of tools available for detection of ransomware in android: commercial removal/cleanup utilities

(e.g., Avast Ransomware Removal). But many such utilities do not use a generic approach. Thus, it requires a certain effort to keep them up to date with the development of new families. HelDroid[9] is a benchmark research on detection of ransomware in android. It proposes a feature-based detection mechanism using static-analysis techniques directly on the extracted bytecode from APK files. This paper is an extension of our previous work [12], which detects the ransomware by checking the permissions requested by the installed apps and also searches for the presence of any threatening text in the app code. In 2017, [13] performs a sequential process of text extraction which involves detection of text followed by text localization, binarization and ends with text recognition. They have proposed an algorithm which uses Scale Invariant Feature Transform (SIFT). Their work is very useful for understanding and implementing image processing. In a study performed in 2018 [14], the authors have integrated the Tesseract Optical Character Recognition (OCR) engine and the Google Vision library to develop an android application to capture the images using camera and extract the corresponding text. They have used Optical Character Recognition.

III. METHODOLOGY

This paper proposed an approach based on static methods to detect ransomware presence on android device. The motive is to determine whether an android app attempts to threaten the user either in form of text or image, to lock the device or wants to access restricted data or a combination of these actions. To achieve this, we extended four modules into RansomwareElite namely threatening text detector, lock detector, offline permission verification and threatening image detector.

A. Threatening Image Detector

This module is the extended version of previous work and detects the presence of threatening text embedded in an image. These types of images are used by the ransomware app to threaten the victim.

This module extracts images available in all the layout files, mipmap files and drawable files of APK. This is followed by the extraction of texts from the respective image files by using Tesseract OCR[10,11]. Further, this text is preprocessed and stored in a file and provided as an input to naive based classifier by python script which uses NLTK and Textblob libraries. If naive based classifier found any threatening text in image then the image is classified as malicious and the corresponding app is consider to be suspicious.

B. Lock Detector

Ransomware locks the victim's device and demand for ransom to provide the key to unlock it. There are some functions, classes and objects which are used to lock the device. For example, the functions which controls the working of main keys of mobile device like back and home disabled by overwriting the function as blank and in this way the device is locked in some extent. To detect the presence of these kind of function, object and class, this module extracts all .class files and convert them to .java files from the APK and stores content in the form of a string using a python code. Further it check whether the string (.java file of the app) contains

1) Class which extends DeviceAdminReciever.
2) Object of type DevicePolicyManager.
3) Function like wipeNow(), resetPassword() or lockNow().

If such type of functions are overwritten in the code of an app or the objects of some classes are created, then those apps are concluded as suspicious.

C. Offline Permission Verfication Module

We have extended the Permission Verification Module [12] now in the form of desktop application. Our Permission verification module in the previous paper checked all the permissions demanded by the android app. This module now works for the uninstalled apps.

First, it will apply the reverse engineering mechanism and extracts the manifest file of the app which has been an input to the module. It will scan the whole file and list all the permissions demanded by the app and store it with the help of python libraries. Then it will compare these permissions with the suspicious permissions mentioned in the table that we have used from our previous work [12]. We have added 1 more permission i.e. SYSTEM_ALERT_WINDOW based on more researches. If the app demands BIND_DEVICE_ADMIN or SYSTEM_ALERT_WINDOW or if it has malicious score equals to or greater than 13, then the app is concluded as suspicious. In the previous paper [12] the count was 8, but on the basis of more researches and experiments it has been concluded that the appropriate count must be 13 and hence the changes have been reflected.

D. Ransomware Database

The results of RansomwareElite which includes the threatening APK(s) package names, their features, on which the APK was classified as suspicious were stored in the online server. The results of applications were sent to Firebase for storage. Firebase is an online cloud database provided by Google. We have used Requests and Firebase libraries in python to send data from our desktop application to Firebase database, whereas android is using FirebaseAuth, Firebase Database libraries. Application is sending only threatening APK(s) data to the server. Based on this dataset, we can further extend our research to understand the nature of the ransomware embedded apps, predict, and classifying various parameters which are commonly found in ransomware.

IV. IMPLEMENTATION

RansomwareElite will decompile the app and convert the corresponding APK into xml, java, text files and images. Then, offline permission verification will read all the permissions from the android apps manifest file and check whether the manifest has any malicious permissions or not. We increment malicious counter each time we find a malicious permission. If this malicious counter is greater than or equal to 13 we flag the app to be suspicious, else non-suspicious. Further, RansomwareElite

checks for the presence of threatening texts in form of text or threatening images in all the files and folders with the help of two modules i.e. threatening text detector and threatening image detector. For this threatening text detector will extract the texts from the xml, java, txt files and threatening image detector will extracts the text from images of format jpg, png, gif and input them to the text classifier to identifies whether the text is threatening or non threatening. If the text is labeled as threatening then the app is considered to be suspicious. Then, the lock detector will go through all the java files and detect the presence of methods and classes which could be used to lock the navigation in the device and restrict the user to operate the mobile phone. Finally, it merges the results of all the modules and if the result from all modules comes out to be suspicious it will mark the app as Ransomware, else it concludes that it's suspicious but not sure of app being Ransomware. Along with this it provides an external feature in which, the details of these ransomware or suspicious apps found will be recorded in the online server, help to create database of such for future research. Fig. 1 shows the working of RansomwareElite.

V. Experimental Results

To test the efficiency and working of RansomwareElite, two experiments were conducted.

A. Experiment 1 : Sample Test Apps

In this experiment, we manually created 9 android test apps as a ground truth to evaluate the enhanced version of RansomwareElite. These apps were incorporated with the various combinations of the ransomware family features like malicious permission, threatening image, locking feature etc. and provided to RansomwareElite. Further we analyzed whether these apps found to be suspicious or not, as shown in the Table I. Further we checked whether the right entries are getting stored in the database for these results, as shown in Table II.

1) TestApp1 : Contains Threatening Text Only
2) TestApp2 : Contains Threatening Image Only
3) TestApp3 : Contains Suspicious Permissions Only
4) TestApp4 : Contains Locking Functions Only
5) TestApp5 : Contains Both Threatening Image & Threatening Text
6) TestApp6 : Contains Both Suspicious Permission & Threatening Text
7) TestApp7 : Contains Both Suspicious Permissions & Locking Functions
8) TestApp8 : Contains All Features
9) TestApp9 : Contains None of the Features

B. Experiment 2 : Android Devices

To test the efficiency of RansomwareElite, it was tested on 48 android devices to evaluate its performance on real android apps. The first app named 'Wall-E' was found suspicious on the basis of Threatening Image within 10 days. This app is used for applying various wallpapers on the android device. However this app did not contain malicious permissions and the locking functions and hence it is not considered as ransomware but is a suspicious app. Other app found suspicious named Smart Protector which is a locking app that provides security to user data. This data has been detected after a month. This app has been determined suspicious on the grounds of Lock Detector and Offline Permission Verification, but did not possess threatening text or threatening image so it is also categorized as suspicious not ransomware. Both these apps were not downloaded from Google play store. These were the third party apps that have been downloaded from some unauthorized websites. All these results have been stored in the online database along with the condition on which it was found suspicious.

TABLE I: Testing Report of Modules

Test Id	Input	Expected Output	Output	Status
1	TestApp1	Threatening Text Present	ThreateningText Present	Pass
2	TestApp2	Threatening Image Present	Threatening Image Present	Pass
3	TestApp3	Suspicious Permissions Present	Suspicious Permissions Present	Pass
4	TestApp4	Locking Functions Present	Locking Functions Present	Pass
5	TestApp5	Threatening Image & Text Present	Threatening Image & Text Present	Pass
6	TestApp6	Suspicious Permission & Threatening Text Present	Suspicious Permission& Threatening Text Presnt	Pass
7	TestApp7	Suspicious Permission & Locking Functions Present	Suspicious Permission& Locking Functions Presnt	Pass
8	TestApp8	Ransomware Present	Ransomware Present	Pass
9	TestApp9	This App Is Safe	This App Is Safe	Pass

TABLE II: Desktop Ransomware Database Storage

Test Id	Input	Expected Output	Output	Status
1	TestApp1	Entry has been added	Entry has been added	Pass
2	TestApp2	Entry has been added	Entry has been added	Pass
3	TestApp3	Entry has been added	Entry has been added	Pass
4	TestApp4	Entry has been added	Entry has been added	Pass
5	TestApp5	Entry has been added	Entry has been added	Pass
6	TestApp6	Entry has been added	Entry has been added	Pass
7	TestApp7	Entry has been added	Entry has been added	Pass
8	TestApp8	Entry has been added	Entry has been added	Pass
9	TestApp9	Entry has been added	Entry has been added	Pass

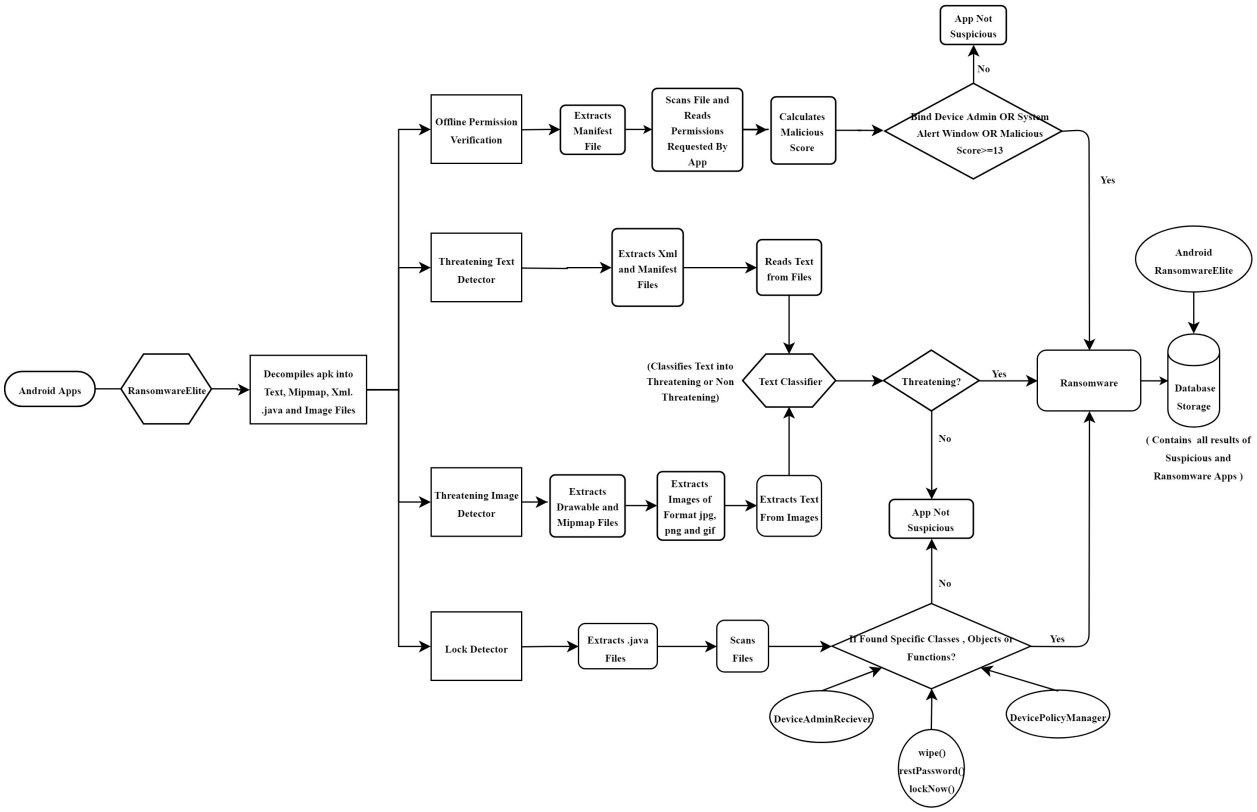

Fig. 1: Working of RansomwareElite

VI. Conclusion

Ransomware has become a major threat as it locks the user's device or encrypts user's data and demands ransom to unlock the device or decrypt the data. In this paper, we enhance RansomwareElite which checks the APK of android apps against ransomware and alerts the user if APK is found suspicious. We have extended the RansomwareElite for the detection of ransomware for uninstalled apps. It consists of 4 modules: threatening image detector which detects any threatening image in APK, offline permission verification which detects suspicious permissions requested by an android app, lock detector which detects methods and classes which could lock the device and threatening text detector which detects whether the android app contains any threatening text. Further, we have created a database which maintains the record of all suspicious and ransomware apps found. We have conducted experiments to test the RansomwareElite and found it performs efficiently to detect the presence of the ransomware.

References

[1] https://blog.barkly.com/ransomware-statistics-2017

[2] https://gbhackers.com/new-ransomware/attackandroidphoneswhichlookslike-a-wannacry/

[3] http://searchsecurity.techtarget.com/definition/ransomware/

[4] https://nakedsecurity.sophos.com/2017/05/19/wannacry-how-safe-is-your-android-phone-from-this-ransomware/

[5] https://www.androidauthority.com/ransomware-attacks-android-increased-751266/

[6] https://cybersecurityventures.com/ransomware-damage-report-2017-5-billion/

[7] https://labs.mwrinfosecurity.com/assets/resourceFiles/mwri- behavioural-ransomware-detection-2017-04-5.pdf

[8] https://en.wikipedia.org/wiki/Android_application_package

[9] N. Andronio, S. Zanero, and F. Maggi, HELDROID: Dissecting and Detecting Mobile Ransomware, IEEE Transactions on Image Processing, Springer International Publishing Switzerland, 2015.

[10] A. Gharib and A. Ghorbani, DnaDroid, IEEE Transactions on Image Processing, Springer International Publishing AG 2017.

[11] C. Zheng (B), N. Dellarocca, N. Andronio, S. Zanero, and F. Maggi, Text Extraction from Text Based Image Using Android, IEEE Transactions on Image Processing, ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2017.

[12] A.Johri, Shivangi, G. Sharma, Akshita, A. Gupta, Machine learning approach to detect presence of Ransomware in android device, 12th INDIACom, 5th International Conference on Computing For Sustainable Global Development 14th to 16th March, 2018.

[13] K. Sahota, L. Awashi and H. Verma , An Empirical Enhancement Using Scale Invariant Feature Transform in Text Extraction from Images,IEEE International Conference on Intelligent Communication and Computational Techniques (ICCT),Manipal University Jaipur, 2017.

[14] A. G. Waghade, A. V. Zopate, A. G. Titare and S. A. Shelke, Text Extraction from Text Based Image Using Android, IEEE International Research Journal of Engineering and Technology (IRJET), vol. 5, issue 3,March 2018.