Proceedings of the 12th INDIACom; INDIACom-2018; IEEE Conference ID: 42835
2018 5th International Conference on "Computing for Sustainable Global Development", 14th - 16th March, 2018
Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA)

Machine Learning Approach to Detect Presence of Ransomware in Android Device

Anubhav Johri[1] , Akshita[1], Gautam Sharma[1], Shivangi Gupta[1], Anuradha Gupta[2]
Jaypee Institute of Information Technology
Noida, India
[1]{ anubhavj22, akshitag1997, gautam.kashyap97, shivangigupta19dec }@gmail.com
[2]anuradha.gupta@jiit.ac.in

***Abstract*—Android mobile devices are not immune to ransomware attacks; rather they have less security than windows and IOS. The world has been terrorized by this cyber-threat as it is intractable and only the prevention and detection can give a solution to this crime. Ransomware is a type of malicious software which encrypts or locks the victim's data and threatens the victim's data or perpetually blocks access to it unless a ransom is paid. This paper is focused to detect the ransomware in android application(app), which are installed on user's mobile device and generate an alert to inform the user about its presence. To detect the presence of ransomware, we made RansomwareElite, it's an android app which detects the ransomware by checking the permissions requested by the android app and also searches the presence of any threatening text in an app code. RansomwareElite reads the manifest file of every android app present on the user mobile device, to know whether any app is requesting for such permissions which could act maliciously and also detect any threatening text in the code of the mobile app that can be later used to threaten the user to pay the ransom. Our work is applicable to all the android app that are either installed or are being installed on the device. We have tested RansomwareElite in 2 stages. First, we tested it with 6 Testapps. Second, we execute RansomwareElite on 66 android devices and detect two malicious apps. We found that RansomwareElite is an efficient way to detect the presence of the ransomware.**

Keywords— Android; API; Malware; Permissions; Ransomware; Text Classifier.

I. INTRODUCTION

Android operating system is the most extensively used operating system by the Smartphone industry. The number of android users has increased manifolds in past few years [1]. Due to this, android became major attraction for the hackers and web criminals to target it. Ransomware attacks on android devices are growing exponentially [2]. Google maintains that dangerous malware is actually extremely rare, and a survey conducted by F-Secure showed that only 0.5% of android malware reported had come from the Google Play store [3]. Ransomware is a form of malicious software that locks the screen or encrypts files on your device, and demands that you pay to get your files back. The payment is made through bit-coins, iTunes, vouchers, and Amazon gift cards etc. Ransomware malware can be broadly categorized as crypto ransomware and locker ransomware that encrypts specific files and locks the entire system respectively. However paying ransom doesn't ensure 100% recovery of data in most cases [4].

The first ransomware attacking android was found in the middle of 2014[5]. It has spread immensely within three years, according to cyber security researchers at ESET (Essential Security against Evolving Threats), android ransomware attacks have risen by over 50% in just a year, peaking in the first half of 2016. In 2016 ransomware exploded about a billion dollars and it is predicted that the damage caused by android ransomware globally could exceed $5 billion in 2017[6, 7]. This infection has caused a huge economic loss especially to US which are most likely to be followed by India because of extreme usage of android mobile phones by the Indians.

The two major approaches for the detection of the ransomware are static and dynamic. Static-based analysis means analyzing an android app) code prior to its execution to determine if it is capable of any malicious activities. If the static analysis finds any malicious code, the executable will be stopped from launching. Dynamic-based analysis detection entails the live monitoring of processes, to determine any malicious behavior. Any malicious behavior process will be flagged as dangerous and terminated [8]. In this research work, we propose RansomwareElite, an android app which detects the presence of ransomware in an android device and provides a solution in order to alert the user regarding the presence of ransomware.

II. RELATED WORK

In literature both static and dynamic analysis approaches have been used for the detection of ransomware. Tianda Yang and Yu Yang proposed a solution of static and dynamic model.

Their model used features: permissions access checking, sequence of API invoking, APK structure and encrypted resource files, checking critical path and data flow, malicious domain access, malicious charges and bypassing the android permissions. But it lacked the requirements to build such detection apps. After that HelDroid [10] was proposed which is a fast efficient and fully automatic approach was presented that recognized unknown scareware and ransomware. The HelDroid, in a general way detected whether an app is attempting to lock or encrypt the device without the user's consent, and if ransom requests are displayed on the screen. It mainly has three different detectors: locking detector, encryption detector, text detector. It uses a text classifier that applies linguistic features to detect threatening text. It uses a fast small emulation to detect locking capabilities and identifies the presence of encryption by using taint analysis which is computationally demanding. Another work [12] was recognized in 2016 which proposed a method that monitors file activities while ransomware accesses and copies them. So the technique detects and eradicates the ransomware by using CPU, I/O usage as well as the information stored in the database. This approach was propped to detect the ransomware in its early stages of malicious activities. R-PackDroid [13] is a supervised machine learning system for the detection of android ransomware was given the spotlight. It introduced a method of using a list of system API packages that can help understand various types of malicious actions. It automatically separates ransomware from genetic malware and trusted files with high accuracy also recognize novel ransomware samples. But they could not analyze the possible attacks on the machine learning algorithm as the main idea was to understand if API packages could really help recognizing new ransomware samples. DnaDroid [14], another approach that overcome the shortcomings of HelDroid used two modules, static and dynamic. This module used text classification, image classification and permission module to categorize the malware as ransomware or not. Once the malware is suspicious then the dynamic module checks for the API calls and then identifies the ransomware. Another approach for the enhancement of HelDroid using static-taint analysis tool, on which the encryption detector is used. For instance, preventing decryption flows from being erroneously considered as malicious, lowering the number of false positives, identified a different set of sources and sinks that allows the detector to identify encryption flows independently of the particular folder that contains the target files and augmented HelDroid for detecting the abuse of admin APIs, which are used by modern ransomware to urge victims to effectively lock the device. In addition to that, also proposed a heuristic to statically resolve the method invoked via the most common reflection patterns, even in the presence of lightweight method name obfuscation. Authors have implemented a pre-filter that aims to reduce the overhead of HelDroid by recognizing good ware.

III. METHODOLOGY

The RansomwareElite consists of an approach based on static methods that is able to detect the presence of ransomware on android devices with the help of its two modules, Permission verification and Threatening text detector. First RansomwareElite examines all the permissions requested by the android apps, because to encrypt or lock the device ransomware containing app will require read and write permissions and second RansomwareElite checks for the threatening text which is used by ransomware writer to threaten the victim and demand the ransom.

A. Permission verification module

Permission verification checks all the permission of an android app. For this we have considered both the android apps a) android app which has already been installed in the system b) those that are requesting to get installed in the system. All these permissions are available in the manifest file of an app. A manifest file is a file containing metadata of a group of xml files and java classes that are part of an app. It contains the names of all android permissions accessed by an app. RansomwareElite uses a "PackageManager" class to extract a list of all the packages in the system. Then it will access the manifest file of the packages and extracts all the permissions with the help of a function getPackageInfo() and store it in a list. Then it compares all the permissions in that list with permissions, mentioned in the Table 1. It maintains a malicious score of an app that increases by 1, if permission matches with the suspicious permissions in the Table 1. After that we count the score of each android app, if the malicious score of any app equals or exceeds the malicious score of 8 or if it contains "BIND_DEVICE_ADMIN" permission then the app is concluded as suspicious and corresponding message will be shown on the screen. Ransomware is a malware which requires certain specific permissions to get control on any android device and lock it. Those permissions are termed as suspicious permissions.

Table 1 Suspicious and Non-Suspicious Permissions

Permission Classification	**CLASSIFICATION**	
TYPE	*PERMISSION*	*SUSPICIOUS*
SYSTEM	GET_TASK	X
	WRITE_SETTING	X
	SYSTEM_ALERT_WINDOW	X
	RECEIVE_BOOT_COMPLETED	✓
	READ_PHONE_STATE	✓
	READ_EXTERNAL_STORAGE	✓
	WRITE_EXTERNAL_STORE	✓
	WAKE_LOCK	✓
	GET_ACCOUNTS	✓
	BIND_DEVICE_ADMIN	✓
	DISABLE_KEYGUARD	✓
	CAMERA	X
	INSTALL_SHORTCUT	✓
SMS	RECEIVE_SMS	X
	SEND_SMS	X
	READ_SMS	✓

CONTACT	READ_CONTACTS	X
	READ_CALL_LOG	X
	CALL_PHONE	X
NETWORK	INTERNET	✓
	ACCESS_NETWORK_STATE	✓
	READ_HISTORY__BOOKMARKS	✓
	ACCESS_WIFI_STATE	✓
LOCATION	ACCESS_COARSE_LOCATION	✓
	ACCESS_FINE_LOCATION	✓

B. Threatening Text Detector

This module uses a python script to detect whether the text file in android app package is threatening or non-threatening. We have implemented Natural Language Processing by using NLTK and Textblob libraries in python. We have used Naive Bayes Classifier to implement NLP classification. This module consists of two phase explained in details.

1) Checking installed apps: We have used our desktop machine as a server and have done server side programming in java to receive and send the text file. Firstly, the rawextract() method is called which extracts the text file from the raw folder in the app package, then getParams() method sends the text file to the desktop machine over same network and creates a file in any drive in our desktop machine. The text in the text file is then taken as an input by the python script, which classifies using the text as threatening or non-threatening and the result is extracted and sent *back to our app as a response.*

2) Checking uninstalled apps: It inputs an apk of an app and does reverse engineering on it using apktool. Apktool is a tool that extracts all the xml files and a manifest file of the apk provided and stores all these files in a folder in a drive in our desktop machine. Our module then extracts all the text from all the extracted xml files of an app. All the extracted texts are stored in a list which is then classified one by one using the text classifier. The module then gives the output if the app has threatening text or not.

IV. IMPLEMENTATION AND RESULT

Firstly, the RansomwareElite has to be installed on a user's device and then it will read all the permissions from an android app to be checked and after that it verifies that is there any malicious permission found or not. If for any android application we get malicious greater than and equal to 8, then it will simply consider it as suspicious further RansomwareElite checks for threatening text. For this RansomwareElite extracts text statements from the xml files of an android app and feeds them to the text classifier that identifies whether the text is 'threatening' or not with the help of machine learning. If the text is labelled as 'threatening' then the app is considered to be suspicious.

To check the efficiency and for the testing of the RansomwareElite, we implemented 2 tests in stages one by one. In the first test, we manually created 6 android apps which we call "Testapps". These Testapps has some malicious which features like asking for malicious permission and threating text. We applied our two modules on these Testapps and analysed the results and check whether the result declares these apps "suspicious" or not. These test apps have been developed on the features possessed by any malware belonging to the ransomware family. Some test apps contains suspicious permissions while some contains threatening text or both. These were made to ensure that the device on which the test is to be carried out remains safe and protected from the ransomware attack.

After the satisfactory results from our first test i.e. from the test apps, we moved to the next test. In our second test, we tested RansomwareElite in real life scenario to check its efficiency. We installed RansomwareElite in 66 different android mobile devices in different surroundings i.e. in devices of people of different age groups in different parts of Delhi- NCR and Haryana. Our app provided two options i.e. either to scan the whole device or to check specific apps. After 10 days, we found one android app unsafe. The suspicious app found was "Mini Militia" a gaming app which was not taken from Google Play store and hence was not genuine. Later a month, to our result, we found 1 more app suspicious. That app was some system locking app of android device which was found to be granted suspicious permission. This app was not found to be responsible for threatening text. Therefore, we concluded out two apps found suspicious in all 66 android devices which were tested.

Hence we have tested RansomwareElite in all aspects and surroundings to check its efficiency and working properly. These results will help the user in realizing the contributor of ransomware before it completely attacks the device and hence the corresponding app will be stopped or uninstalled to prevent damage and loss of data.

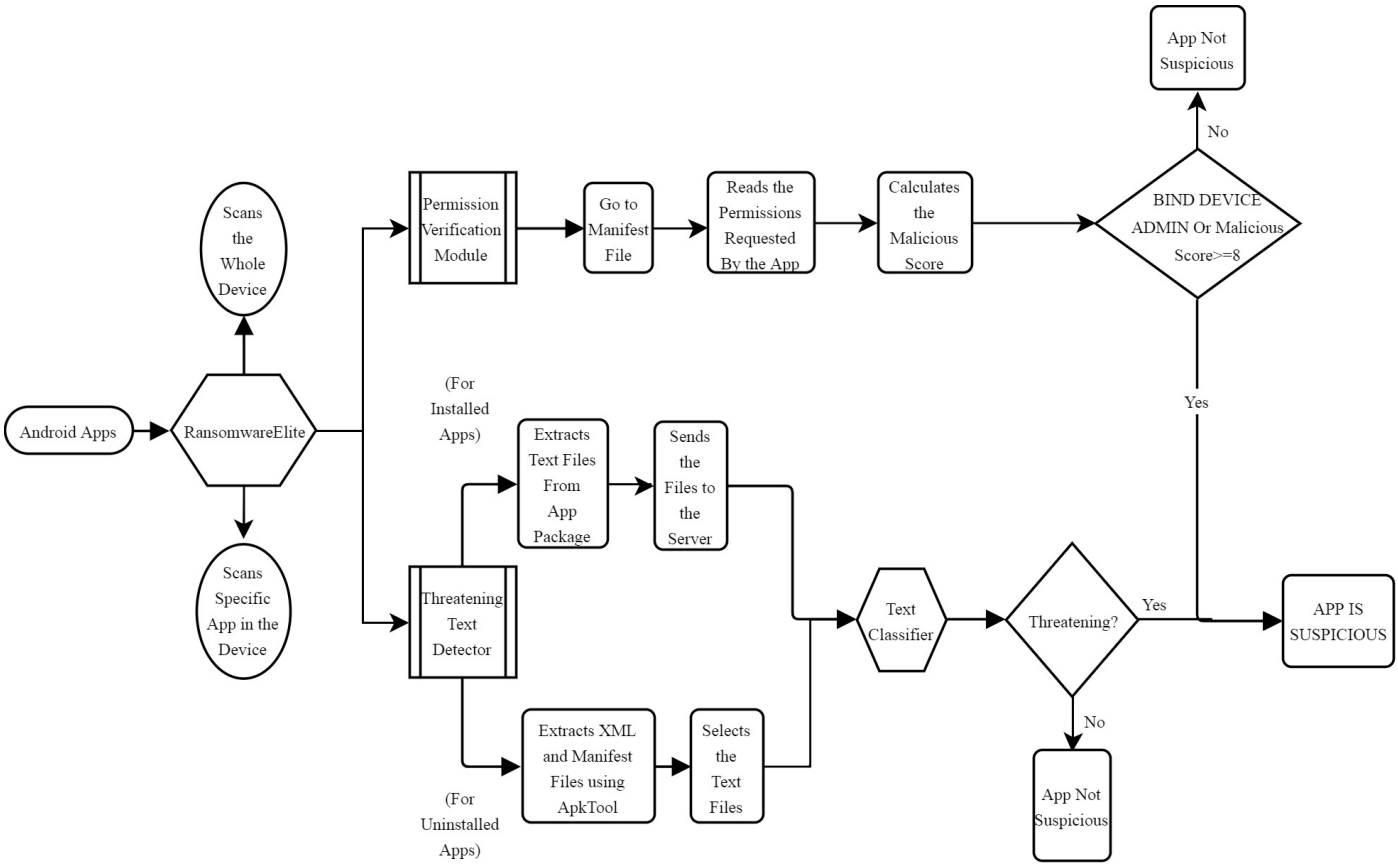

Fig.1 RansomwareElite Working Model

V. CONCLUSION

Ransomware that has become a major threat nowadays, demands money by locking or encrypting data of the user. In this paper, we propose a RansomwareElite android app which checks the user system against ransomware and if it finds any suspicious activity then it will generate an alarm. For the detection of the ransomware, we have created the RansomwareElite app which consists of two modules: Permission verification and Threatening text detector which detects suspicious permission request in an app and whether that app contains any threatening text in it or not. We tested RansomwareElite with 6 Testpps and an experiment was performed on 66 android devices. In our result we found TestApps were detected by ransomware and 2 android app were detected by RansomwareElite. Since the android market is extremely vast and as mentioned above ransomware is an ever increasing and dangerous threat worldwide, there is an urgent need for a proper detection and preventing method for this problem.

REFERENCES

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

[2] https://gbhackers.com/new-ransomware-attackandroidphoneswhich-lookslike-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://cybersecurityventures.com/ransomware-damage-report-2017-5-billion/

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

[9] T. Yang, Y. Yang, K. Qian, D Chia-Tien, “Automated Detection and Analysis for Android Ransomware”, 1338-1343. 10.1109/HPCC-CSS-ICESS.2015.39

[10] N. Andronio, S. Zanero, and F. Maggi (B), “HELDROID: Dissecting and Detecting Mobile Ransomware”, IEEE Transactions on Image Processing, Springer International Publishing Switzerland, 2015

[11] F. Mercaldo (B), V. Nardone, A. Santone, and C. A. Visaggio, “Ransomware Steals Your Phone. Formal Methods Rescue It”, IEEE Transactions on Image Processing, IFIP International Federation for Information Processing 2016

[12] S. Song, B. Kim, and S. Lee, “Effective Ransomware Prevention Technique Using Process Monitoring on Android Platform”, IEEE Transactions on Image Processing, March, 2016

[13] D. Maiorca, F. Mercaldo, G. Giacinto, C. A. Visaggio, F. Martinelli, “R-PackDroid”, IEEE Transactions on Image Processing, April 03-07, 2017

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

[15] C. Zheng (B), N. Dellarocca, N. Andronio, S. Zanero, and F. Maggi, “GreatEatlon”, IEEE Transactions on Image Processing, ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2017