Automating the process of browsing and downloading APK Files as a prerequisite for the Malware Detection process

Prerna Agrawal[1], Bhushan Trivedi[2]

[1]Faculty of Computer Technology,
GLS University, Ahmedabad, India

Abstract: ***Android has grown to the world's one of the largest mobile platforms. Android applications are capable of performing any functionality in the mobile platform. Android applications are also vulnerable to malware. So our goal in this paper is to collect Malware files of generalized Malware Families which will help us as well as other researchers also to conduct experiments for Malware Detection. Here we propose the process for automating browsing and downloading the APK files. We have also developed and implemented our Crawler to automate the process of browsing and downloading the files. We have downloaded Malware and Benign files from the most popular Android Malware Projects. Using this process we have collected around 15508 Malware files and 4000 Benign Files. We have also discussed our Malware Detection process and we have covered the Android File collection phase in this paper.***

Keywords: Machine Learning, Android, Malware, Android Security, Malware Detection, Mobile Security

1. INTRODUCTION

Conventionally the Android Malware Detection process is static based on various methods like Signature/Pattern, Resources, Components, and Permissions [3]. Machine Learning methods are applied extensively in ascertaining if the given APK file is malware [18]. Machine learning methods are found to be less time consuming and less resource consuming compared to non-machine learning-based techniques [1][18].

A researcher can determine if a given APK file is malware or not. For that, a researcher needs to design his/her own typical set of processes. And for this process, a researcher wants to have his/her dataset for the same. The dataset generation process plays a very important role as it contains a list of features that will help in determining that a given APK file is malware or not. Feature Selection for the dataset is also equally important as various malware files exploit those features. So that exploited features help a researcher in determining the malware file. Our dataset generation process involves Android File Collection, Decompilation, and Feature Mining Phases. In the Android File Collection phase, different malware and benign APK files are downloaded. In the Decompilation phase, reverse engineering of APK files is carried out. In Feature Mining phase the selected features are extracted from the decompiled APK files.

In this paper, we are discussing the Android File collection phase of our dataset generation process. In this paper, we propose the process for automating of browsing and downloading the APK files which we have implemented. We have also developed a Crawler for the following 2 main objectives: 1) To automate the filedownload process 2) To download as many files as possible. Our goal is to collect Malware files of generalized Malware Families which will help us as well as other researchers also to conduct experiments. By using our proposed process we have collected a total of 15508 malware files and 4000 benign files. The Decompilation, Feature Mining Phases will be covered in other papers.

This paper is divided into the following sections. Section 2 describes the overall Process of Malware Detection for our Generalized Detection Engine. Section 3 describes the Android File Collection phase. Section 4 describes the various challenges we faced during the data collection. Section 5 describes some limitations of the file collection process and Section 6 describes the conclusion of the paper.

2. OVERALL PROCESS OF MALWARE DETECTION

This section explains the Overall Process of Malware Detection. Figure 1 explains the logical flow of Android Malware Detection. This Process is mainly divided into 4 phases:

1. Android File Collection
2. Decompilation
3. Feature Mining
4. Machine Learning

The Android File Collection is the first paper that we have covered in this paper. Other phases will be covered in other papers. In the Android File Collection phase, all the Android files are collected. These files are chosen from many popular Android Malware projects like Drebin [8], Kharon [10], Koodous [11], AndroZoo [5], and Android PraGuard [6]. The APK is an Android Package file format that is used to run on devices using the Android Operating System. The APK file contains the Java code to run the application, AndroidManifest.xml, resources, assets, etc. The Java code is compiled into bytecodes which form .dex files which are Dalvik Executable files. Android Operating system uses the Dalvik Virtual Machine (DVM) to run .dex files. The AndroidManifest.xml file describes the name, version of the application, permissions required by the application, intents, intent filters, activities, Broadcast receivers and services needed by the application. In the Decompilation phase, the APK files are reverse engineered.The reverse engineering process separates the Java code and XML files from the APK file.For selecting appropriate features we have performed static analysis on the online malware scanners [2] and we also reviewed the Static Analysis malware reports of Android files provided by Sandroid [16]. We have also studied the datasets available for the detection of Android Malware of famous projects Drebin and MalGenome [17]. The selected features are given as an input in the Feature Mining phase. In Feature Mining Phase the selected features are extracted from the decompiled files and the feature dataset is prepared. The final dataset contains main features like API Calls, Permissions, and Intents. This dataset is given to the Machine Learning phase where various supervised algorithms, Feature Extraction Techniques, and Ensembling Process is used for classification of malware and benign files. Using Machine Learning we will train the dataset first and then based on the trained data all the algorithms will be applied to the testing data which will give more accurate results in malware detection. This will give a better approach to Malware Detection Mechanism.

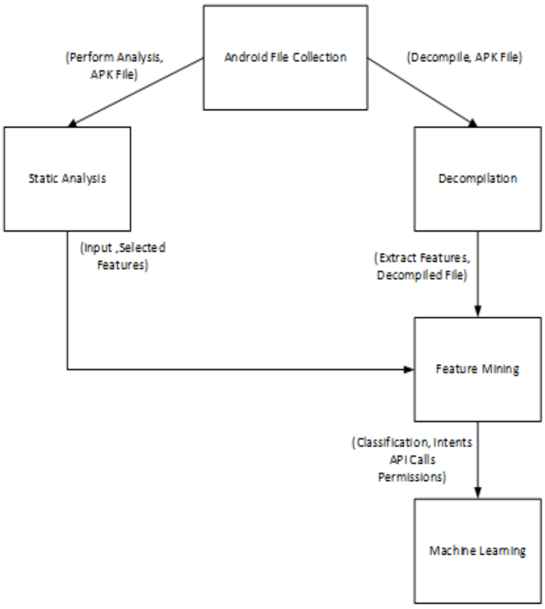
Android File Collection
(Perform Analysis, APK File)
(Decompile, APK File)
Static Analysis
Decompilation
(Input ,Selected Features)
(Extract Features, Decompiled File)
Feature Mining
(Classification, Intents API Calls Permissions)
Machine Learning

Figure 1: Logical Flow of Android Malware Detection

3. ANDROID FILES COLLECTION

This section explains the overall process of the Android file Collection. The collection of Android files is the first step towards Dataset preparation which is explained here.

3.1 File Collection Process

This section presents the architectural flow of the Android Files Collection process. Figure 2 illustrates the Architectural flow of the file collection process. Here user runs the Android File collection module types a specified URL and sends gives malware or benign file request for download. This module connects to a website through a specified URL and after a successful connection, the website connects to the server and sends a file request for download given by the user. The server responds with the file path to the website and finally, the malware or benign file gets downloaded and is saved and stored to a physical location. The malware and benign files are decided based on the worldwide Android Malware datasets and online websites available.

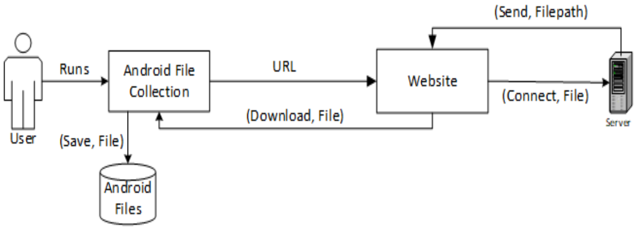

Figure 2: Architectural Flow of Android File Collection Process

We have studied some well-known Android Malware datasetsand Benign Datasets and described our observations in the paper. These datasets are mostly cited, mostly used, and mostly referred by other researchers. Also, these datasets contain a wide range of malware families and so our study is quite reliable indicating that the results that we have obtained are equally reliable.

The Drebin Dataset is a worldwide famous project available for downloading the Android malware dataset [7] [8]. In Kharon dataset malware from a total of 7 families is collected and it contains around 50 applications [9] [10]. Koodous dataset also provides various malware applications for download by using the search criteria by package name, SHA256, hash value, etc [11]. The Androzoo dataset is a very large collection of applications and is growing rapidly day by day [4] [5]. Android PRAguard dataset is also rich and contains 10479 applications from different families and contains samples of MalGenome and ContagioMinidump datasets [6].

We have collected around 15508 Malware applications using the process described in Figure 2. From Drebin [8], around 5490 applications are downloaded. From Kharon [10] and Koodous [11] around 46 applications are downloaded. From Androzoo [5] around 4000 applications are downloaded. From Android PRAguard [6] around 5953 applications are downloaded. We have selected these famous Android Malware Datasets for our process as these datasets contain a wide range of malware families and also contains recent malware samples.

We have collected Benign Applications from datasets, Google Play Store, and online websites. The KuafuDet dataset [13] mainly contains 242,500 benign applications that are downloaded from the Google Play Store and the other 10,400 malicious APK files. The Canadian University of Cybersecurity [14] also provides Malware and Benign Files for download.

We have collected around 4000 benign applications using the process described in Figure 2. From Google Play Store around 250 applications are downloaded. From the KuafuDet dataset [13] around 500 applications are collected. From the Canadian University of Cybersecurity [14] around 1150 applications are collected. We have also developed a Crawler to crawl the files from the websites [15] and around 2150 files are collected from there. We have selected the official Google Play store, some famous datasets and famous online websites for benign files download. Table 1 provides a summary of total applications downloaded from different Android Market places.

Table 1: Summary of Android Files

Market Place	No. of Files Downloaded	File Type	Total Files
Drebin	5490	Malware	15508
Kharon and Kudoos	46	Malware	
Androzoo	4000	Malware	
AndroPRAguard	5953	Malware	
Google Play Store	250	Benign	4000
Canadian University of Cybersecurity	1150	Benign	
Websites	2150	Benign	
KuafuDet	500	Benign	

3.2 Crawling

We have developed a Crawler for crawling the applications from websites [15] using technologies like Cypress Framework and Node package. We have used Cypress 4.0 and Node 10.16.3 with Chrome Browser. We have created a script in Cypress Framework. The script uses the inbuilt functions of the Cypress Framework.cy is the Cypress object used to call the inbuilt functions in the script. We build this script using the following steps:

1. Firstly we have used the function cy.visit (URL) in the script. This function will visit the specified URL of the website.
2. That URL initially shows some APK files and then a button named Load More is placed. On clicking that button other APK files will be shown. Like this, there are many Buttons name Load More are placed. We have also automated this manual process by getting the .loadmore button's click event from the HTML code of the website. We used cy.get(.loadmore).click() function in the script that will get all the .loadmore button events and click them automatically so that all the APK files are listed.
3. Once all the files are listed then the script will collect all APK files URLs by using cy.find() from the HTML code and store all the URL's in an array named data.
4. Then each APK file URL will be visited using cy.visit(URL) function.
5. If the APK file URL exists then we split the URL and the APK file name is extracted from the APK file URL using the cy.split(URL) and stored in a variable named appname.
6. And then that APK file is using function cy.downloadfile(appname). Like this, all APK files will be downloaded automatically using the script.
7. To execute the Script in our local machine we have used the Node Package. We need to open the Command Prompt and move to the root directory of the Script.
8. We need to execute the command npm run cypress: open and the crawler execution will be started.

This crawler provides a simple way of automatic download of bulk applications. We have used the crawler to crawl the benign files from the websites. Figure 3 describes the entire working process of the Crawler developed by us. Firstly the Crawler needs to be initialized with a website URL. If the specified URL is valid then it will fetch that specific Web Page for that URL and parse the web page. While parsing the web page it will look for APK file URL and if the URL exists than it will fetch the APK file URL and download that specific APK file. After downloading the file it will search for another APK file URL and if again URL is found it will fetch URL and download another APK file. This process will be continued until all the APK URLs are parsed and fetched.

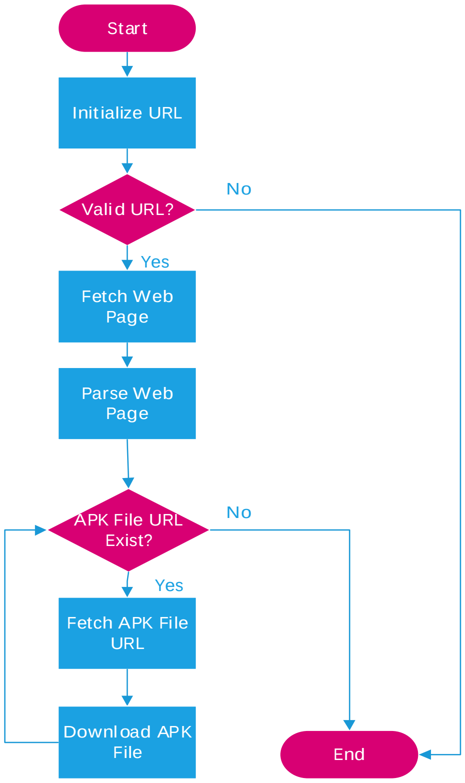

Figure 3: Crawling Process Flow

4. DATA COLLECTION CHALLENGES

We faced many challenges and hurdles while collecting the APK files from different Android Markets.Some general issues are the following:

1. Request for Granting Access to Datasets: Android Malware and Benign APK datasets are openly available but the direct download is not possible. We need to request for the download of the datasets by sending an email for authentication purpose and then the researchers of the datasets allowed us to download the datasets by providing API key, passwords, or download links. All the datasets Drebin [8], Kharon [10], Koodous [11], AndroZoo [5] Android PraGuard [6] KuafuDet dataset [13] The Canadian University of Cybersecurity [14] are downloaded through requests. All these datasets chosen are used very much worldwide by the researchers for their work also these datasets cover various types of malware families and have a huge collection of files.
2. Timeout Situations: When we used our crawler for automatic download of bulk files from the websites

many times there were Timeout situations due to huge file size or lack of stability of HTML code of websites. So we need to start the crawler again after solving this problem.

3. Manual Download of Files:In AndroZoo [4] [5] After receiving the API key access from the researcher to download the files we were not allowed to download the bulk files at a single shot. Androzoo [5] provides an excel file which contains all the details of the malware files like SHA256, Package Name, File Size, Date, Time. SHA256 value is unique for each file. Androzoo[5] gives you the flexibility to download each malware file of your choice. One can download the recent malware as well as other old malware families also. One can have a collection of malware files based on their choice and requirement. One can get details of each file in the excel sheet provided by Androzoo [5].We followed the following steps and downloaded each file manually.

1. Link for Downloading the file
https://androzoo.uni.lu/api/download?apikey=${APIKEY}&sha256=${SHA256}

2. Replace API Key obtained by the Androzoo [4][5].

3. Pick the SHA256 value is given in Excel File and replace the SHA256 value in the link.

4. Copy-paste that whole URL in the browser.

Example:
https://androzoo.uni.lu/api/download?apikey=17c0605b2ead628babafb7b0cd09d85205bba380d73275418c6856b9aaa36f21&sha256=0000003B455A6C7AF837EF90F2EAFFD856E3B5CF49F5E27191430328DE2FA670

5. API Key will remain the same but the SHA256 value will differ for each APK file. So replace SHA256 value for each APK and copy-paste URL for each APK file and download it.

5. LIMITATIONS

The number of Malware Files collected using this process imposes some limitations also. They are as follows:

1. For a researcher to determine a Malware Family for a single Malware file is not possible. We don't provide any extra information regarding the Malware Files.
2. The Malware Files we have collected are from the period between 2011 to 2019.
3. The Crawler we have developed works on specific online websites only.
4. We have collected only those files which are freely available.

6. CONCLUSION

For a researcher to determine whether the given APK file is a malware or not he/she needs to create his/her dataset. We have discussed our Dataset generation process which contains Android File Collection, Decompilation, and Feature Mining Phases. We have discussed the Android File collection phase in this paper and also proposed a process for automating of browsing and downloading the APK files which we have implemented. We have also designed and implemented our Crawler and achieved our objectives. Using this proposed process we have collected a total of 15508 malware files and 4000 benign files.We also discussed some challenges faced during the data collection and some limitations of data collection.

We have accomplished the Collection of Malware and Benign files for dataset preparation in the Android File Collection phase. We have also described the whole process of the Android File Collection and also the working of the Crawler that we developed. Decompilation, Feature Mining, and Machine Learning phases will be covered in the rest of the papers.

ACKNOWLEDGMENTS

We would like to acknowledge our two students Ms.SaberaKadiwala and Ms.Indushree Shetty of GLSICT, GLS University who helped us in manually downloading the files from Androzoo [5]. They downloaded around 4000 files from Androzoo [5] dedicating their time for this task.

REFERENCES

[1] Prerna Agrawal, Bhushan Trivedi, "Machine Learning Classifiers for Android Malware Detection", 4th International Conference on Data Management, Analytics and Innovation (ICDMAI) Springer AISC Series, New Delhi, Jan 2020. (Paper to be Published)

[2] Prerna Agrawal, Bhushan Trivedi, "Analysis of Android Malware Scanning Tools", International Journal of Computer Sciences and Engineering, Vol.7, Issue.3, pp.807-810, Mar 2019.

[3] Prerna Agrawal, Bhushan Trivedi, "A Survey on Android Malware and their Detection Techniques", Third International Conference on Electrical, Computer and Communication Technologies (ICECCT) IEEE, Feb 2019.

[4] Kevin Allix, Jacques Klein, Yves Le Traon, "Androzoo: Collecting Millions of Android Apps for the Research Community", ACM, May 2016. Online Link: https://orbilu.uni.lu/bitstream/10993/27396/1/androzoo.pdf

[5] Androzoo Dataset, Online Link: https://androzoo.uni.lu/

[6] Android PRAGuard Dataset, Online Link: http://pralab.diee.unica.it/en/AndroidPRAGuardDataset

[7] Daniel Arp, MichealSpreitzenbarth "DREBIN: Effective and Explainable Detection of Android in your pocket" NDSS, Feb 2014.

[8] Drebin Files Dataset, Online Link: https://www.sec.cs.tu-bs.de/~danarp/drebin/index.html

[9] Nicolas Kiss, Jean-Francois Lalande, MouradLeslous, Valerie Viet Triem Tong, "Kharon dataset: Android malware under a microscope" The Learning from Authoritative Security Experiment Results (LASER) workshop, May 2016, San Jose, United States. Pp.1-

12.

[10] Kharon Dataset, Online Link: http://kharon.gforge.inria.fr/dataset/index.html

[11] Koodous Dataset Online Link: https://koodous.com/apks

[12] Daniel E Krutz, Andrez Ruiz, Jared Smith, “A Dataset of Open-Source Android Applications” 12th Conference on Mining Software Repositories IEEE, May 2015.

[13] KuafuDet Dataset, Online Link: https://nsec.sjtu.edu.cn/kuafuDet/download.html

[14] UNB Dataset, Online Link: https://www.unb.ca/cic/datasets/invesandmal2019.html

[15] Android Files Download, Online Link: https://apkpure.com/

[16] Android Malware Dataset, Online Link: https://figshare.com/articles/Android_malware_dataset_for_machine_learning_2/5854653

[17] Sandroid Malware Detection Results, Online Link: “http://sanddroid.xjtu.edu.cn:8080/#overview”

[18] Ebtesam J. Alqahtani, RachidZagrouba_and Abdullah Almuhaideb, “A Survey on Android Malware Detection Techniques Using Machine Learning Algorithms”, Sixth International Conference on Software Defined Systems IEEE, 2019.

AUTHORS

Ms.Prerna Agrawal completed her Master of Computer Application from Gujarat University, Ahmedabad, Gujarat, India in 2008. She is currently working as an Assistant Professor in the Faculty of Computer Technology (MCA) at GLS University. She is currently pursuing her Ph.D. from GLS University, Ahmedabad, Gujarat, India. She has total of 8 years of teaching experience and 1-year industry experience. Her main research work focuses on Android Malware Detection and Application Security.

Dr.Bhushan Trivedi completed his Master of Computer Application from MS University, Baroda, Gujarat, India in 1988. He completed his Doctor of Philosophy from Hemchandracharya North Gujarat University, Gujarat, India in 2008. He is currently working as a Director and Dean in Faculty of Computer Technology (MCA) at GLS University. He has more than 20 years of teaching experience. He is acting as a Ph.D. guide at GLS University. His main research area is on Intrusion Detection. He has 4 patents on his name. He has currently 8 research scholars enrolled under him. He has published more than 50 research papers in various national and international journals and conferences.