An Adaptive Dataset for the Evaluation of Android Malware Detection Techniques

Omar Hreirati
Dept. of Electrical and Computer Engineering
Queen's University, Kingston, Ontario, Canada
16oh3@queensu.ca

Shahrear Iqbal, Mohammad Zulkernine
School of Computing
Queen's University, Kingston, Ontario, Canada
{iqbal, mzulker}@cs.queensu.ca

***Abstract*—Android is currently the leading mobile operating system in the world. The huge number of Android devices attracts developers to create applications for it. However, it also attracts attackers that collect sensitive data or make money. This problem has led many researchers to propose malware detection systems and custom versions of Android that can help users against malicious activities. Evaluating these systems is a crucial part of malware prevention research. However, recent datasets that cover different kinds of benign and malicious applications to evaluate the malware detection techniques are often not available. With thousands of newly released applications every day and different new malicious activities discovered, it is difficult to keep malicious application datasets up to date. This paper introduces a recent and adaptive dataset that includes 5,000 applications from different malware categories that can be used by the research community. The applications are selected from more than 5 million applications. To show how the dataset can be used, we deploy a popular malware analysis platform and generate detailed reports on all the applications in an automated way. We also provide the steps to update the dataset and perform the analysis automatically on the updated set of samples. We believe that the adaptiveness of the dataset and the automatic analysis process will help researchers save time in preparing their datasets and focus more on the detection techniques.**

***Index Terms*—Android malware, malware dataset, dynamic analysis.**

I. Introduction

Android is dominating the mobile operating system market worldwide. Unlike iOS which allows users to install only approved apps from the App Store, Android allows users to install applications from many sources. Although Google Play Store holds the biggest source of Android applications, there are many more sources and third-party stores, alongside homebrew, torrent, and direct installations.

This freedom that Android provides comes with a price as it is easier for attackers to publish and distribute malicious applications. The attackers offer simple and sometimes beneficial or entertaining applications while invading the users' privacy in the background. Malicious applications can access personal or sensitive information like the user's location or contact list, send SMS messages or even make phone calls without the user's permission. Fake logins are also popular, as the application might ask the user to provide their credentials to access a specific service or feature, and then send this information to the application developer. Some applications might collect information about the user's device or content, and others can collect a dump of the memory. All these make it almost impossible to completely secure Android devices against malicious applications. Even the official Play Store has published many malicious applications and adware [1], [2].

Android security and the problems with malware applications caused many researchers to work on malware detection techniques for Android. As an alternative to anti-malware applications provided by software security companies like Norton and Kaspersky, researchers found multiple ways to secure the Android environment. One way is to provide a real-time monitoring system that enforces security policies and helps users detect any malicious activities running in the background. Although this method provides accurate results in detecting malware and anomalies [3], in most cases the damage will be done already at the time of detection, as the malicious code would be already running on the device. The other proposed methods are mainly for analyzing and detecting Android malware prior to the installation, these methods use static and dynamic analysis techniques. One of those detection systems performs a broad static analysis across the operating system, gathering all the features of the installed applications and scans for any malware patterns [4]. In [5], the authors proposed a hybrid detection system that performs anomaly detection using dynamic analysis, in addition to signature detection. In a more recent research [6], the authors proposed a component that uses a context management system to track the context of the phone and enforce corresponding security policies automatically.

The problem that most researchers face when designing such techniques is the evaluation, as there are many different application sources and different replicas for each application. This creates the need for a dataset that the research community can use to evaluate the performance of their techniques. Since creating a dataset can take a huge amount of time and effort, researchers tend to use the dataset they can find in the literature, which can be many years old. This is what triggered the motivation for this project, to help the research community by providing an up to date dataset that has around 5,000 Android applications including benign and malicious applications from several malware categories. We then show how researchers can use our dataset with malware analysis platforms. Specifically, we use CuckooDroid (a dynamic analysis platform for Android applications) to execute all the applications automatically in a

virtualized environment and generate reports on the activities of the applications.

To overcome the limitations of the currently used datasets, the paper also serves as a manual that outlines the main steps to create an updated version of the dataset without going through the hassle of researching and troubleshooting. This is what makes the dataset adaptive, as it is easy to be customized based on the researchers' needs. The process starts with over 5 million applications and the result is a dataset of about 5,000 applications. All the resources, tools, and code that we used to create the dataset is released and available in http://research.cs.queensu.ca/~qrst/androidmalwaredataset/.

In particular, the contributions of the paper are as follows:

- We create an up to date Android malware dataset from millions of Android applications available across multiple stores and sources. The final dataset includes 5,000 applications, including 500 benign applications, and 4,500 malware applications from the 22 most popular malware categories.
- We show how researchers can use our dataset with malware analysis performs in an automated way and generate reports.
- We provide clear guidelines for creating a newer version of the dataset and corresponding analysis reports.

The rest of the paper is organized as follows. Section II describes the related work. In Section III, we explain the methodology of creating the dataset and how to create an updated version of it from scratch. We describe the details of the dataset in Section IV and finally, we conclude in Section V.

II. Related Work

Allix et al. [7] proposed Androzoo, a growing collection of Android applications collected for the research community from different sources and stores including the official Google Play Store. At the time of writing this paper, Androzoo has 5,796,118 different APK files (Application Package Kits for Android apps). In addition to the APK files, the authors also include a number of metadata about each application. For example, Androzoo uses VirusTotal [8][1] to analyze all the applications and include how many antiviruses detect the applications as malware. Androzoo also updates the database weekly. The snapshot used in this project was extracted in June 2017 and includes 5,544,082 Android applications. Androzoo database can be downloaded as a CSV file.

In [9], the authors proposed a tool called Euphony that goes through the Androzoo database and analyzes the VirusTotal reports to identify the malware names and types found in the applications. They publish a list called *Labels* that provides the names and types of the malware -if any- inside the APK files. We use this list to select the top malware categories for our dataset.

Although Androzoo is more comprehensive, we prepare our dataset to provide an easier way for researchers to evaluate their detection techniques as dealing with several million applications in most cases would be impossible. Our dataset is also adaptive, so researchers can easily extract the applications from Androzoo again to suit their needs. For example, they can customize the categories, application sources, and the number of applications selected from each category or the total number of applications inside the dataset.

[1]Provides output from more than 60 well-known antiviruses about an app.

III. Methodology

This section provides a step-by-step guide to create our dataset and how to update it with more recent Android Applications. The process includes obtaining the Androzoo database, processing the database, selecting the apps, and downloading the APK files. We also describe how to setup the CuckooDroid malware analysis platform and generate comprehensive reports for each application. Figure 1 shows the steps of our methodology. The blue boxes represent the creation of our dataset and the green boxes represent the generation of the analysis reports.

A. Creating the Dataset

Setting up the database. The first step is to set up the preferred database to be able to import the data and design the experiment. We use the PostgreSQL database for simplicity and the speed advantage in terms of importing large data. It is also recommended by the CuckooDroid team.

Downloading the Androzoo database. The two databases of interest (Androzoo and Euphony labels) can be downloaded from the Androzoo official website [10]. The first one is called *latest.csv*. As mentioned before, this file includes many useful properties such as the sha256 checksum values, package names, VirusTotal scores, source markets, and many more. The second one is called *Labels* which includes the malware types and names.

Importing the downloaded files into the database. We import the two files into two tables. Since Androzoo contains both benign and malicious applications, it is normal for the *labels* table to have fewer entries, because it only includes malicious applications.

Merging the databases. Once the data is imported, it is time to merge the data of the two tables into one table. The best way to do so is to copy the `names` column from the *labels* table to the other table whenever the sha256 fields match. Otherwise, the value of the name should be "null" because it means that the application is benign.

Extracting the top malware categories. After merging the data into a single table with all the properties of all the applications, the top malware categories can now be easily extracted through SQL commands.

Selecting the apps. At this stage, we should decide which malware categories to include in the dataset and how to select applications. This includes the total number of Android applications in the resulted dataset, the proportion of benign to infected applications, the number of studied categories, and the number of applications within each category. Depending on the purpose of the research, and the intended usage of the dataset, these properties can be customized.

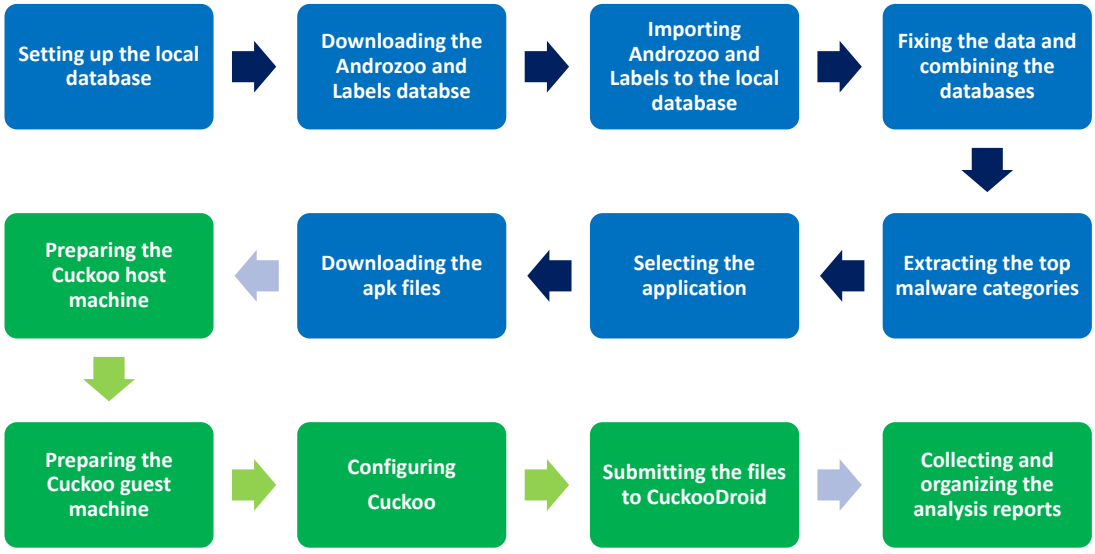

Fig. 1. The steps of our methodology for creating the dataset and generating malware analysis reports.

Downloading the app APK files. After the selection of the apps, we need to download the APK files from the Androzoo server. Androzoo requires a unique key which will allow researchers to access the server and download APK files. The key can be obtained by contacting the Androzoo team and requesting access. We then download the APK files using the sha256 checksum value which identifies individual applications.

B. Generating Malware Analysis Reports

In this section, we explain how CuckooDroid works and how to perform the analysis to generate the malware analysis reports. The setup we used is an Ubuntu machine as the host and a virtual machine running Ubuntu as the guest.

CuckooDroid. CuckooDroid [11] is an extension of the cuckoo sandbox. Cuckoo sandbox [12] is one of the leading open source malware analysis systems designed for automated malware detection and profiling. Cuckoo can be used to execute and analyze files inside a controlled and isolated environment (guest virtual machine), and generate a comprehensive report outlining the behavior of the malware along with their activities. Some examples of the results are traces of API calls performed by the processes, file operations like creating or deleting or downloading files, memory dumps of the malware processes, network traffic trace provided in PCAP files, screenshots during execution, and full memory dumps of the machine.

CuckooDroid brings the ability to execute Android APK files and perform analysis using the Cuckoo sandbox. Android emulators execute the applications inside the guest machines. In addition, the guest machines are also configured to use the Droidmon monitoring module that is based on the Xposed framework [13]. CuckooDroid provides all the analysis information in a comprehensive report for each analyzed application. This information is helpful in detecting any suspicious activity performed by the analyzed applications in the background without the user's knowledge.

Preparing the Cuckoo host and guest machines. The first step is to install Cuckoo on the host machine. Then, we create a virtual machine and install Ubuntu in it as the guest. Cuckoo supports multiple virtual machines at the same time. However, since Android emulation is usually very resource hungry, we used only one guest machine during our experiment. Nonetheless, several machines can be replicated easily, if resources permit.

After installing the Ubuntu, we establish a connection between the host and guest machines. There are two adapters that should be set up. The first is for forwarding the internet connection from the host machine and the second adapter is for the communication between the host and guest machines. To ensure the network connectivity between the host and the guest, a ping command should ensure a successful connection. Next, Android SDK is installed in the guest. Then, we create a new device using the Android Virtual Device Manager and copy the following two files from the Cuckoo directory into the guest machine:

1. utils/android_emulator_creator/create_guest_android_on_linux.sh

2. agent/agent.py

We start the Android emulator and run the *create_guest_android_on_linux.sh* script and after that the *agent.py* script. The guest is now ready for incoming connections from Cuckoo. It is necessary to take a snapshot of the guest virtual machine at this stage, with both the emulator and the agent running. CuckooDroid will use the snapshot to start the guest machine and begin analysis right away.

If more than one virtual machine is used, the machine can be easily cloned. The only change needed is the IP address in the Ubuntu configuration.

Cuckoo Configuration. Back in the host machine, the last step

is to tweak the configuration files for Cuckoo. The files can be found in the *conf* folder inside the Cuckoo directory. We highlight the necessary changes for each file in our technical report [14], and the full details can be found in CuckooDroid's manual.

Submitting APK files to CuckooDroid. APK files from our dataset are now submitted to CuckooDroid. Cuckoo offers two methods to analyze the samples. The first is using the web interface, which will be explained in section IV. The second is through the utility script provided by CuckooDroid. The web interface is more user-friendly, however only useful for individual file submission. For a large number of files, the script is the fastest way to submit the samples. The script can be found in the Cuckoo directory under *utils/submit.py*. Providing the script with a path of a folder that includes multiple APK files will result in queueing all the files in the folder to Cuckoo with a unique task ID for each file. Task IDs are generated sequentially starting from Task 1.

At this stage, if Cuckoo is running, the analysis will start right away. Otherwise, all the tasks will be queued and saved and cuckoo will run the analysis once it is started. The analysis will automatically start the snapshot of the virtual machine and install the apks in the emulator. Then, the application will be executed and the analysis report will be generated.

IV. Dataset

In this section, we describe the details of our dataset and the CuckooDroid analysis reports. Then, we discuss the characteristics of a good dataset and our limitations.

A. Selected Applications

For our dataset, we picked 5,000 Android applications out of the 5,544,082 total applications, 500 of which are benign, and the other 4,500 are malicious. The number of applications for each malware type reflects how popular the attack is and how important it is to detect it.

To provide a balanced dataset that can be beneficial for the research community, the dataset covers the top 22 malware categories in Androzoo. We divided the applications into 3 groups and sorted them according to the number of occurrences for each malware. The distribution of the selected malware apps across multiple categories is shown in Table I.

TABLE I
THE DISTRIBUTION OF THE MALWARE APPLICATIONS IN THE DATASET

Group	Category	Number of Applications in Androzoo	Number of Applications per Category	Total Number of Applications Per Group
1	dowgin	262,057	500	1,500
	kuguo	107,114		
	airpush	100,415		
2	revmob	74,419	200	2,200
	youmi	51,762		
	artemis	37,706		
	droidkungfu	37,214		
	leadbolt	31,491		
	adwo	28,733		
	jiagu	27,526		
	wapsx	23,781		
	deng	23,538		
	startapp	21,468		
	genpua	21,155		
3	admogo	19,940	100	800
	waps	18,287		
	anydown	17,459		
	domob	16,224		
	umeng	13,999		
	utchi	13,262		
	igexin	12,448		
	smspay	11,458		
			TOTAL	**4,500**

B. CuckooDroid Analysis Reports

CuckooDroid analysis reports can be accessed in two ways. Below, we describe both of them.

CuckooDroid web interface. The web interface is accessed by running *utils/web.py* which is inside the Cuckoo directory. The web interface provides the functionality of submitting individual files. It is a more user-friendly approach than the manual method for a small number of files.

The web page will show a list of tasks in the Cuckoo database, which includes completed and pending tasks, indicated by the assigned task ID. Once the analysis is completed and the reports are generated for a task, they can be accessed by clicking on the corresponding task ID in the web interface. This will open an HTML version of the report, with all the details and screenshots available in an organized fashion. The HTML reports give details about the analysis such as the date and duration of the analysis process, file details, Android application information, activities, services, receivers, permissions, signatures, dynamic analysis results, screenshots, and network analysis reports.

File structure of the analysis result folder. While the web interface provides an easy way to access the reports, it is not very useful if the analysis needs to be done for a large number of files. Going through thousands of pages and extracting the information manually is impossible. Therefore, Cuckoo keeps the results in an organized file structure. Inside the *storage/analyses* folder in cuckoo directory, the results are organized by task IDs, each task has its own folder named according to its ID. Inside each folder, the following files and folders can be found [15]:

- analysis.log: This is a log generated by the analyzer and it contains the trace of the execution inside the guest machine. All the outputs and error messages that were displayed during the analysis process can be found here.
- dump.pcap: Network dump generated by tcpdump.
- dump_stored.pcap: A sorted version of dump.pcap, which allows the web interface to lookup TCP stream.
- Files: This folder contains whatever Cuckoo was able to isolate and dump from the files that the malware operated on.
- Logs: The logs folder contains all the logs that were generated by Cuckoo's process monitoring.
- Reports: This folder includes two files. *report.html* is the same report that is displayed in the web interface of Cuckoo. *report.json* is a comprehensive report that includes all the analysis details, such as the VirusTotal status for every provider, the yara hits, network activity,

malicious activity, etc.

Our dataset contains all the analysis files mentioned here for 5,000 Android applications and the corresponding APK files organized in three groups.

C. Discussion

For a malware dataset to be useful for the research community, it should satisfy a number of characteristics and qualities. For example, it should include benign as well as malicious applications from different categories. Another important property for a dataset is whether it contains the newer malware. As thousands of applications are being added to different sources every day, it is essential for the evaluation to be performed on a recent dataset. The dataset should include any new malware categories as well as new versions of the applications. Any dataset will become irrelevant after few years, and there will be a need for a new one.

A good dataset should also be manageable and easy to use. Having a lot of applications can make the process of generating the analysis reports very time-consuming. An unstructured dataset makes analyzing the results confusing and might lead to inaccurate findings.

We incorporate all these characteristics in our dataset. Additionally, our dataset is completely adaptive and can easily be recreated from Androzoo to include newer benign and malicious applications as Androzoo continually updates itself.

D. Limitations

We use the metadata Androzoo provides for each application. However, we do not verify that the information in the Androzoo database is accurate. Also, we included the 22 most frequent malware categories when we created the dataset. However, the number of infected applications may not necessarily represent how serious the attack is. Important categories can be missed if this process is performed without some background information about each category.

V. Conclusion

We have presented an up to date Android malware dataset for researchers. Researchers are struggling to find recent datasets to evaluate their malware analysis and detection techniques. As millions of Android applications are available from many different sources, attackers are finding new ways to gain access to devices. New malware types are frequently found in Android applications and using an obsolete dataset weakens the evaluation of a malware detection technique. Researchers tend to avoid creating new datasets because it is very time consuming.

Our dataset is based on a huge collection of Android applications. It starts with 5.5 million applications collected in 2017 which covers the most popular sources of Android applications and the result is a dataset with 5,000 applications (500 benign and 4,500 malicious from the 22 most common malware categories). The dataset has benign applications to help eliminate false positives. We have also shown how to use our dataset in an automated fashion using the CuckooDroid dynamic analysis platform. We have made the dataset and the malware analysis results public for the research community.

To make further contributions and avoid the inevitable state of the dataset being outdated after few years, we also give a step-by-step instructions on how to create an updated version of the dataset in the future, making the dataset adaptive. Researchers may use our method to create a customized version of the dataset. Furthermore, researchers that are interested in a specific type of malware can create their own dataset with that specific type only. We plan to analyze all the generated reports from CuckooDroid and devise novel techniques to detect malware using our dataset in the future.

Acknowledgment

This work is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Research Chairs (CRC) program.

References

[1] D. Palmer, "Phoney android security apps in google play store found distributing malware, tracking users." [Online]. Available: https://goo.gl/FKx5uq, Jan 2018. (Accessed on 2018-02-28).

[2] P. H. O'Neill, "Google killed 700,000 malicious apps in the play store in 2017." [Online]. Available: https://www.cyberscoop.com/chris-wlaschin-es-s-voting-security/, Jan 2018. (Accessed on 2018-02-28).

[3] D. Schreckling, J. Köstler, and M. Schaff, "Kynoid: Real-time enforcement of fine-grained, user-defined, and data-centric security policies for android," *Information Security Technical Report*, vol. 17, pp. 71 – 80, Feb 2013.

[4] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "Drebin: Effective and explainable detection of android malware in your pocket," in *In Proceedings of Network and Distributed System Security Symposium (NDSS)*, Feb 2014.

[5] X. Wang, Y. Yang, and Y. Zeng, "Accurate mobile malware detection and classification in the cloud," *SpringerPlus*, vol. 4, p. 583, Oct 2015.

[6] M. S. Iqbal and M. Zulkernine, "Droid mood swing (dms): Automatic security modes based on contexts," in *In Proceedings of the 20th Information Security Conference (ISC)*, pp. 329–347, Springer International Publishing, Oct 2017.

[7] K. Allix, T. F. Bissyandé, J. Klein, and Y. Le Traon, "Androzoo: Collecting millions of android apps for the research community," in *In Proceedings of the 13th International Conference on Mining Software Repositories*, pp. 468–471, ACM, May 2016.

[8] VirusTotal, "Virustotal is a free service that analyzes suspicious files and urls and facilitates the quick detection of viruses, worms, trojans, and all kinds of malware.." [Online]. Available: https://www.virustotal.com/, 2017. Accessed: 2017-08-03.

[9] M. Hurier, G. Suarez-Tangil, S. K. Dash, T. F. Bissyandé, Y. L. Traon, J. Klein, and L. Cavallaro, "Euphony: Harmonious unification of cacophonous anti-virus vendor labels for android malware," in *In Proceedings of 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pp. 425–435, IEEE, May 2017.

[10] U. du Luxembourg, "Androzoo." [Online]. Available: https://androzoo.uni.lu. (Accessed on 2017-06-30).

[11] "Cuckoodroid - automated android malware analysis with cuckoo sandbox." [Online]. Available: https://github.com/idanr1986/cuckoo-droid. (Accessed on 2017-09-30).

[12] "Cuckoo sandbox - automated malware analysis." [Online]. Available: https://cuckoosandbox.org/. (Accessed on 2017-08-31).

[13] "Guest machine architecture — cuckoodroid v1.0 book." [Online]. Available: http://cuckoo-droid.readthedocs.io/en/latest/installation/guest_android_on_linux/architecture/. (Accessed on 2017-08-31).

[14] O. Hreirati, "An Adaptive Dataset for Android Malware Research." [Technical Report]. Detpartment of Electrical and Computer Engineering, Queen's University, May 2018.

[15] "Analysis results — cuckoo sandbox v2.0.0 book." [Online]. Available: https://docs.cuckoosandbox.org/en/latest/usage/results/. (Accessed on 2017-11-30).