

Twitter data analysis to enhance Android malware detection

by

Gauresh Singh Rajawat

B.E., Ujjain Engineering College, India, 2015

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2020

Approved by:

Major Professor
Dr.Doina Caragea

Copyright

© Gauresh Singh Rajawat 2020.

Abstract

In recent years, we have witnessed a proliferation of mobile applications (or apps), including useful, benign apps, and also malicious apps (or malware). Identifying malicious apps is a challenging but urgent problem, as malicious apps can cause significant damage and financial losses to their users. Most systems for identifying malware rely on features extracted from the code of the apps themselves using static or dynamic analysis. However, many zero-day malware apps still evade such systems and enter the market. To complement the information contained in the code and facilitate the detection of zero-day Android malware apps, we propose to use social media information, specifically, Twitter to identify tweets that talk about Android malware, in particular those that may contribute to the spread of the malware. The assumption is that users who try to advertise and/or spread malware share the characteristics of spam users. We have used Twitter Developer's APIs to crawl a large number of tweets that contain URLs corresponding to Android apps. The tweets, together with meta-information about their retweets/favorites and about their users, have been stored in a MongoDB database. The URLs in the collection of tweets collected have been matched with Android apps using information crawled from Google PlayStore. Furthermore, the apps found in tweets that were matched to apps in Google PlayStore have been labeled as benign or malware using a platform called AndroZoo, which uses anti-virus programs such as VirusTotal to identify malware. Finally, Twitter users who post malware are being studied to identify patterns characteristic of spam users, which could potentially be used to identify zero-day malware.

Table of Contents

List of Figures	vi
List of Tables	vii
Acknowledgements	viii
Dedication	ix
1 Introduction	1
1.1 Why Twitter?	3
2 Data collection and pre-processing	4
2.1 Creating a MongoDB database	5
2.2 General statistics about tweets	7
2.2.1 Top 5 most popular languages	7
2.2.2 Users with maximum statuses count	8
2.3 Tweets containing URLs	8
2.4 Data from AndroZoo	9
2.5 Data from Google PlayStore	13
2.6 User's metadata	15
3 Analysis and results	16
3.1 Analysis of Twitter app URLs	16
3.2 Analysis of AndroZoo data	17
3.3 Analysis of Google PlayStore data	19

3.4 Analysis of user behavior	20
4 Conclusions and future work	24
Bibliography	26

List of Figures

2.1	Tweet captured in JSON format	5
2.2	Command to import data on Mongo Database	6
2.3	Database on Mongo displaying Tweets information	7
2.4	Most Popular Languages	8
2.5	Users with largest status count	9
2.6	Distribution of apps by VT detection count.	10
2.7	Distribution of apps by the year of their VT scan.	11
2.8	MongoDB entry corresponding to an AndroZoo file	12
2.9	MongoDB Query to get the count of apps based on their Scan date	12
2.10	Package names of applications in AndroZoo	13
2.11	Package names converted to links	13
2.12	Work-flow of the project	14
2.13	Application URLs from Google PlayStore	15
3.1	MongoDB query to extract URLs from Twitter data	16
3.2	Results of running the query on MongoDB	17
3.3	Most frequently tweeted application links	18
3.4	MongoDB Query to get the results for VT Detection count equals to zero	18
3.5	Mongo Query to get the results for VT Detection count greater than Zero	19
3.6	Number of malicious and benign applications labeled vs total application links on Twitter	20
3.7	Total Number of apps matched on each platform	21
3.8	Count of followers of an account whose tweets contain malware app links.	23

List of Tables

2.1	Tweet fields included in the MongoDB database	6
3.1	Count of total tweets versus tweets containing URLs	17
3.2	User behavior comparison based on field names	22

Acknowledgments

I would like to communicate my incredible thankfulness to Dr.Doina Caragea for her continuous support and constructive feedback during the planning and development of the project. Her willingness to help and vast knowledge on the subject, have helped in shaping the outcomes in a much profound way. Her patience with me the entire time is very much appreciable.

I would also like to thank the Support team of the Computer Science Department for helping me setting up the account on servers and providing me with the required software to run the project.

Finally, I would like to express my gratitude to my friend, Akash for introducing me to this wonderful field and supporting and encouraging me throughout my master's studies.

Dedication

I want to dedicate this work to my parents, who trusted me that I can graduate with a degree in a field that is completely different from my past career. I also would like to dedicate this work to Dr. Doina Caragea, without whose ideas and suggestions I would have lost my way.

Chapter 1

Introduction

With the advent of technology, recent years have witnessed an ample amount of growth in the field of mobile applications. There are about 3.5 billion smartphone users in the world¹, and 74.13% of the users use Android smartphones. The number of Android applications has risen from 2.6 million in 2018 to 3.04 million as of September 2020 in Google PlayStore alone². These applications (or apps for short) available in the markets can either be benign (no signs of compromise or spam) or malicious. With the usage of the apps in the day-to-day life, malicious apps impose a serious threat to the user, which often includes collecting the GPS coordinates, contact list of the user, browsing history, or even stealing credit card numbers or other financial information, and causing significant damage to users. Hence it is important to segregate the malicious applications from the benign ones. There are tools and systems created for this purpose, and many of them rely on features/information extracted from the code of the apps themselves using dynamic or static analysis.

However, these systems can not accurately detect if zero-day malware apps enter the markets. The most common and adaptable route for an application to enter the market is through social media. Once the application is available to download, spam users on multiple social media platforms, try to advertise the app to attract the users. Oftentimes, they post the link to the Android market from where the app can be downloaded. Such posts on social media can potentially be useful in identifying zero-day malware as soon as they enter the

markets, before the more traditional anti-virus systems, can detect them.

To test this hypothesis, the goal of my thesis is to utilize/scrutinize a popular social media platform, Twitter, to find information about zero-day malware, and use that information to enhance malware detection systems that rely only on information extracted from the code. Twitter is used in this research, as it is a popular and easily accessible social media platform, and at the same time provides an API for crawling data.

To facilitate the classification of Android apps as malware or benign, tweets were collected using Twitter Streaming APIs, which is available for public use. All the tweets along with their metadata, which includes retweets and user's information, have been stored in a database created over Mongo, for easy access and querying. Tweets containing URLs to Android apps are filtered from the tweets that do not contain URLs. Given that Google PlayStore is widely used to download applications belonging to different categories ranging from games, fashion to educational and informational apps, apps found in the tweets are then matched with apps received from crawling the Google PlayStore. Python scripts have been written to crawl the app data, specifically the Android app links available there. The apps found at Twitter are labeled as benign or malware using a platform named AndroZoo, which uses multiple anti-virus programs, including VirusTotal, to detect the malware or malicious behaviours, and label apps accordingly. Additionally, all labelled links found in the tweets are also matched with the data of Google PlayStore. The reason for matching apps mentioned on Twitter with apps in Google PlayStore is to see if any of the apps on Google PlayStore may be labeled as malware.

More precisely, tweets containing the mention of malicious apps have been separated from the pool of tweets to study user characteristics. The assumption is that the users who try to advertise and/or spread malware, share the characteristics of spam users. The patterns formed can then be potentially used to identify zero-day malware.

1.1 Why Twitter?

Twitter has been a very popular and powerful social media platform for more than a decade. From 400,000 tweets posted per quarter in 2007 to 330 million registered users as of today, Twitter has been successful in gaining the attention of the mass population worldwide³. A typical user spends about 3.39 minutes on the platform per session. One of the interesting things to note here is that about 40 percent of Twitter users purchase some product after seeing it on Twitter. When it comes to reviewing a product, be it the launch of a mobile phone, software application, or clothing accessory, people make posts on Twitter, thus providing their reviews on the product. In many cases, this proves to be helpful as an honest review is shared by the one after using it. But nowadays, with the growth of competition in all fields, many people post untrue reviews, thus degrading the market image of the product. Bots are also trained to perform such tasks. The process of differentiating the posts done by a human from those done by bots is oftentimes not successful. Hence Twitter is chosen over any other social media platform.

Chapter 2

Data collection and pre-processing

Twitter provides streaming APIs⁴⁵, which helps in capturing tweets from all around the world in nearly real-time. To get access to the developer's streaming API, one needs to complete a form available on Twitter's website. The crawler for capturing tweets is written in Python and includes an API Key and a Token. Tweets for this project are collected during two time periods: from March 15, 2020, to May 21, 2020, and from August 31, 2020 to September 25, 2020. Since the project focuses on the tweets related to Android apps, we collected the tweets using keywords, such as android, app, application, playstore, security, mobile application, malware, malicious, and apk. The tweets crawled were saved in multiple JSON files, labeled based on dates. The Computer Science department's server was used for running the crawling script, which provides an ample amount of storage to store these files.

The JSON files crawled consist of several pieces of information, including the date when the tweet was created, the username (the name of the user who posted the tweet), geo-location (if this feature was enabled by the user), followers count (count of followers the account has), favorite count (total count of tweets liked by the user), whether the tweet received any replies, tweet text, and others. Some of these fields give the information about the user, such as its favorites and followers count, whether the user is verified, the number of total tweets posted, etc. This data about users can potentially be used to determine whether a user is genuine or a bot or spam. Figure 2.1 below shows the format in which tweets were

received:

```
{
  "created_at": "Fri Apr 17 08:52:06 +0000
  2020",
  "id": 1251070922766929928,
  "id_str": "1251070922766929928",
  "text": "https://
  /t.co/sQvbj8ZL0o",
  "source": "\u003ca href=\"http://twitter.com/#!/download
  /ipad\" rel=\"nofollow\"\u003eTwitter for iPad\u003c/a
  \u003e",
  "truncated": false,
  "in_reply_to_status_id": null,
  "in_reply_to_status_id_str": null,
  "in_reply_to_user_id": null,
  "in_reply_to_user_id_str": null,
  "in_reply_to_scre
  n_name": null,
  "user": {
    "id": 1077659047464235009,
    "id_str": "1077659047464235009",
    "name": "Jay",
    "screen_name": "Jay41477382",
    "location": null,
    "url": null,
    "description": null,
    "translator_type": "n
    one",
    "protected": false,
    "verified": false,
    "followers_count": 43,
    "friends_count": 184,
    "listed_count": 0,
    "favourites_count": 14697,
    "statuses_count": 7295,
    "created_at": "Tue
    Dec 25 20:15:13 +0000
    2018",
    "utc_offset": null,
    "time_zone": null,
    "geo_enabled": false,
    "lang": null,
    "contribu
    tors_enabled": false,
    "is_translator": false,
    "profile_background_color": "F5F8FA",
    "pro
    file_background_image_url": "",
    "profile_background_image_url_https": "",
    "profile_bac
    kground_tile": false,
    "profile_link_color": "1DA1F2",
    "profile_sidebar_border_color": "
    C0DEED",
    "profile_sidebar_fill_color": "DDEEF6",
    "profile_text_color": "333333",
    "profi
    le_use_background_image": true,
    "profile_image_url": "http://abs.twimg.com/sticky
    /default_profile_images
    /default_profile_normal.png",
    "profile_image_url_https": "https://abs.twimg.com
    /sticky/default_profile_images
    /default_profile_normal.png",
    "default_profile": true,
    "default_profile_image": false
    },
    "following": null,
    "follow_request_sent": null,
    "notifications": null,
    "geo": null,
    "coo
    rdinates": null,
    "place": null,
    "contributors": null,
    "is_quote_status": false,
    "quote_cou
    nt": 0,
    "reply_count": 0,
    "retweet_count": 0,
    "favorite_count": 0,
    "entities": {
      "hashtags": [],
      "urls": [
        {
          "url": "https://t.co/sQvbj8ZL0o",
          "expanded_url": "http:
          //Google.com",
          "display_url": "Google.com",
          "indices": [0, 23]
        }
      ],
      "user_mentions":
      [],
      "symbols":
      []
    },
    "favorited": false,
    "retweeted": false,
    "possibly_sensitive": true,
    "filter_level": "
    low",
    "lang": "und",
    "timestamp_ms": "1587113526418"
  }
}
```

Figure 2.1: *Tweet captured in JSON format*

2.1 Creating a MongoDB database

Since the tweet data collected was originally in the JSON format, it makes it difficult to analyze the contents and find information about Android malware apps. Therefore, to better analyze the data and perform the required operations, a MongoDB database was created. To import the JSON files to the database, the command in Figure 2.2 was used:

Creating a MongoDB database provided us not only with the ease to query, but it also

```
C:\Program Files\MongoDB\Server\4.2\bin>mongoimport --db twitter --collection tweets --file C:\twitterData.json
```

Figure 2.2: *Command to import data on Mongo Database*

fieldName	Description
createdAt	Denotes the date and time tweet is created.
user.lang	This field contains the acronym of the language in which the tweet is done. For instance, "en" for English, "ja" for Japanese, and so on.
user.url	This field contains the URL of the application, which has been mentioned by the user.
user.protected	If TRUE, indicates that the user has chosen to protect their tweets
user.verified	If TRUE, indicates that the user's account is verified
user.followers_count	This field indicates the number of followers this account has
user.friends_count	The number of users this account is following on Twitter
user.favourites_count	This field denotes the number of tweets this account has liked since the day it is created
user.statuses	Total number of tweets committed by the user, including the retweets
user.geo_enabled	This field if set to TRUE, enables users to share their location
user.default_profile	If TRUE, indicates that the user is still using the same default theme or background of their user profile

Table 2.1: *Tweet fields included in the MongoDB database*

enabled safe storage of the data for future studies. Since there are more than 100 columns for each tweet, most of the columns were discarded as they did not contain the information needed for this project. Included columns are: tweet creation date, user object fields including language used, URL, protected, verified, count of followers, count of friends, count of favorites, total statuses, whether the geological location is enabled, and if the profile is the default. These field names are listed and described in more detail in Table 2.1⁶:

Fig.2.3 depicts as how fields are displayed over the database created on Mongo:

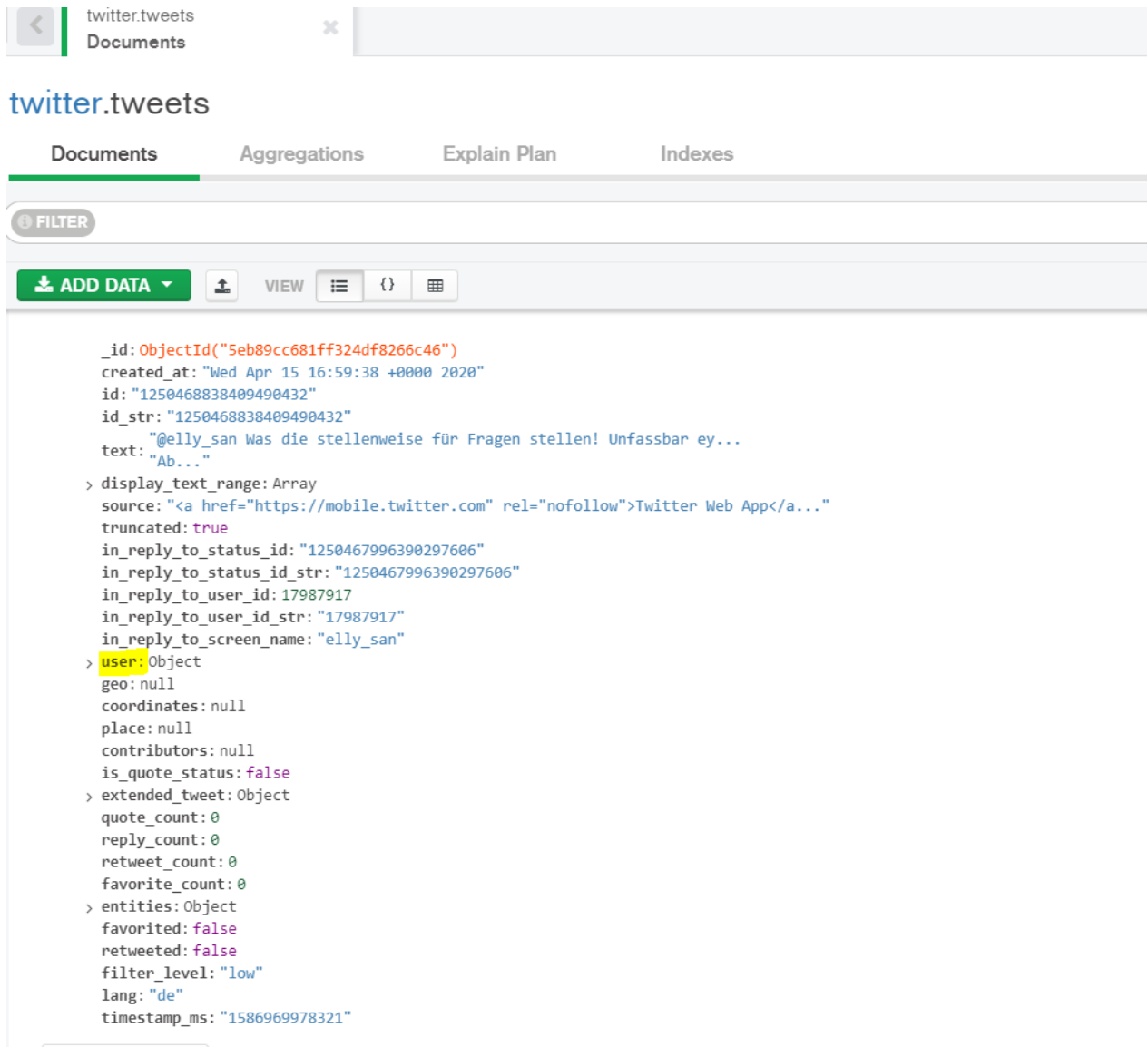


Figure 2.3: Database on Mongo displaying Tweets information

2.2 General statistics about tweets

On studying the data received after crawling Twitter using research specific keywords, some general statistics of these tweets are obtained.

2.2.1 Top 5 most popular languages

English is found to be the most used language on Twitter with around 40.1% of the users tweeting in English, abbreviated as “en” on their platform, followed by Japanese, abbrevi-

ated as “ja” with 23.8%. Figure 2.4 provides information about the most frequently used languages in our security-related dataset.

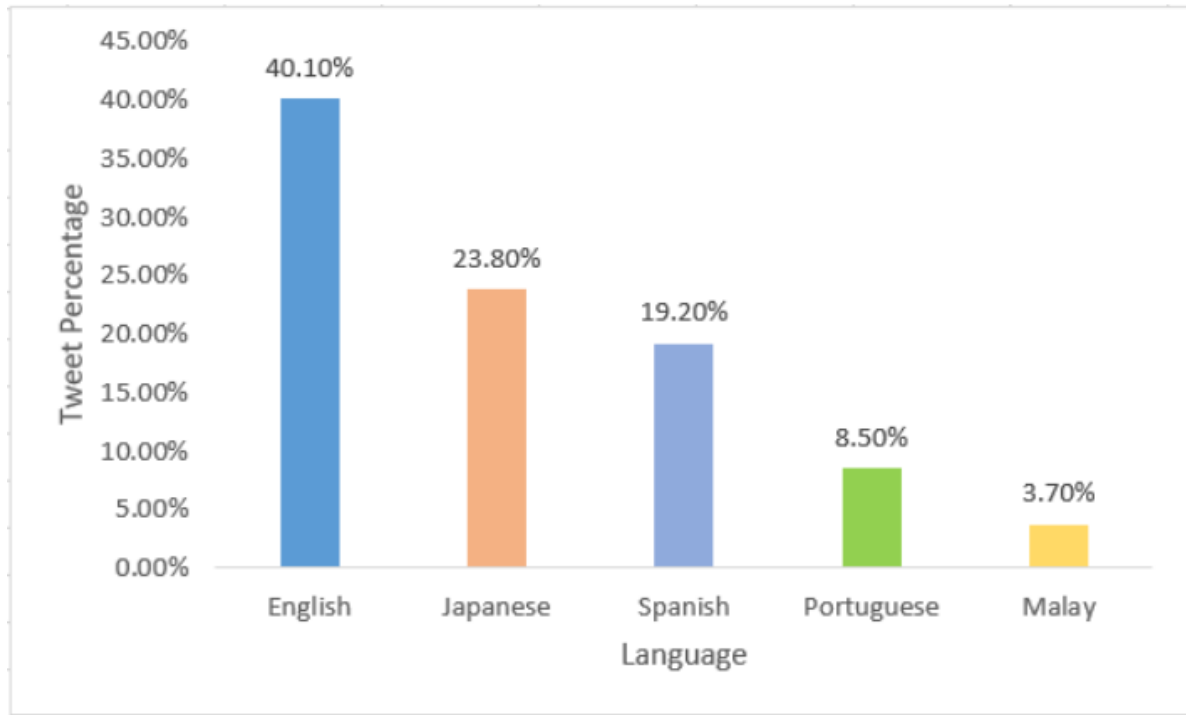


Figure 2.4: *Most Popular Languages*

2.2.2 Users with maximum statuses count

On running the query on MongoDB, I also found out the users, who had the largest number of statuses on the platform. User with the screen name “famima_reply” has statuses count of 30,206,797, followed by “ElNacionalWeb” with count 7,351,378. The top five users have been displayed below in Figure 2.5.

2.3 Tweets containing URLs

Out of 9.1 million tweets crawled on Twitter, around 2.3 million tweets contain URLs to some web page or application. For this research, I only shortlisted the tweets which contain links in the description of the tweet. These links, in some cases, also indicate the name of

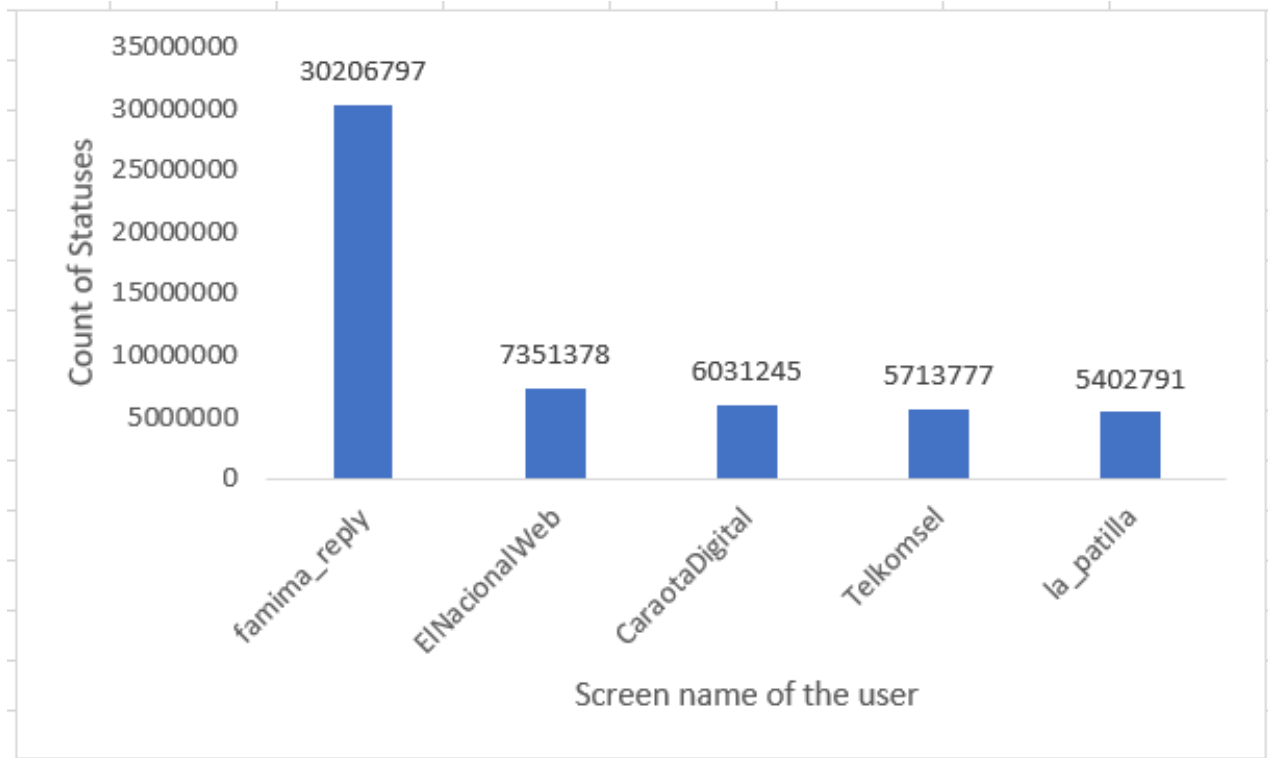


Figure 2.5: *Users with largest status count*

the market pool from where it was downloaded. For instance:

<https://play.google.com/store/apps/details?id=com.instagram.android>.

The application here is “Instagram” and the market from where it is downloaded is “Google PlayStore”. These URLs are then compared with the existing trusted data to get the labels predicted as malware or benign.

2.4 Data from AndroZoo

AndroZoo is an open platform that maintains a large collection of Android applications. These applications are analyzed by different Anti-Virus tools, including VirusTotal. The label of whether the application listed is malicious or benign is then published⁷.

AndroZoo has got a collection of approximately 11 million URLs. Along with the URL of each app in this collection, AndroZoo makes available the VirusTotal predictions, denoted as VT detection count, and the name of the market from where an app was downloaded.

The VT detection count signifies the count of Anti-Virus vendors who detected the application as malicious. It ranges from 1 to 100, but the file's maximum count is found to be 57. If the count is more than or equal to 10, we consider the application as malicious. The apps with VT count lying in the range of 1 and 10 are considered to form a grey area, as they can be either false-positives (for some vendors) or false-negatives (for other vendors). However, intending to identify potential zero-day malware entering the markets, I have also included those apps in the analysis and I considered them to be malware. Figure 2.6 shows the number of apps with zero VT count, with VT count between 1 and 10, and with VT count greater than 10.

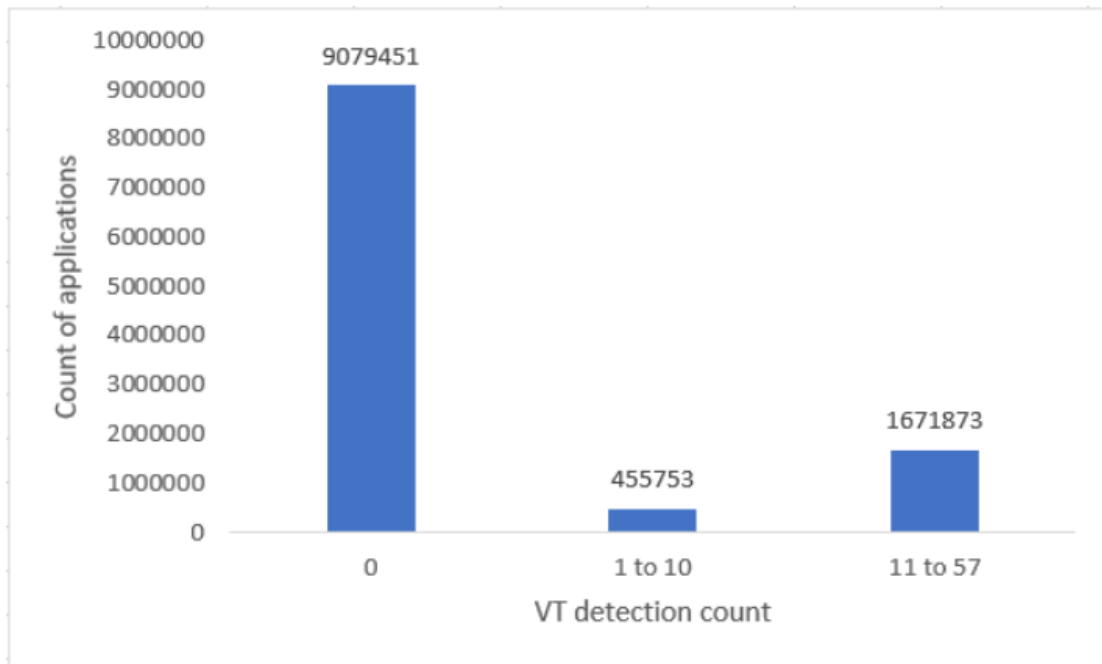


Figure 2.6: *Distribution of apps by VT detection count.*

Different applications have been scanned and added to the AndroZoo database in different years. Many of these scans date back to 2014, while some applications have also been scanned in 2020. Figure 2.7 shows the graphical representation of the count of applications by the year of their VT scan.

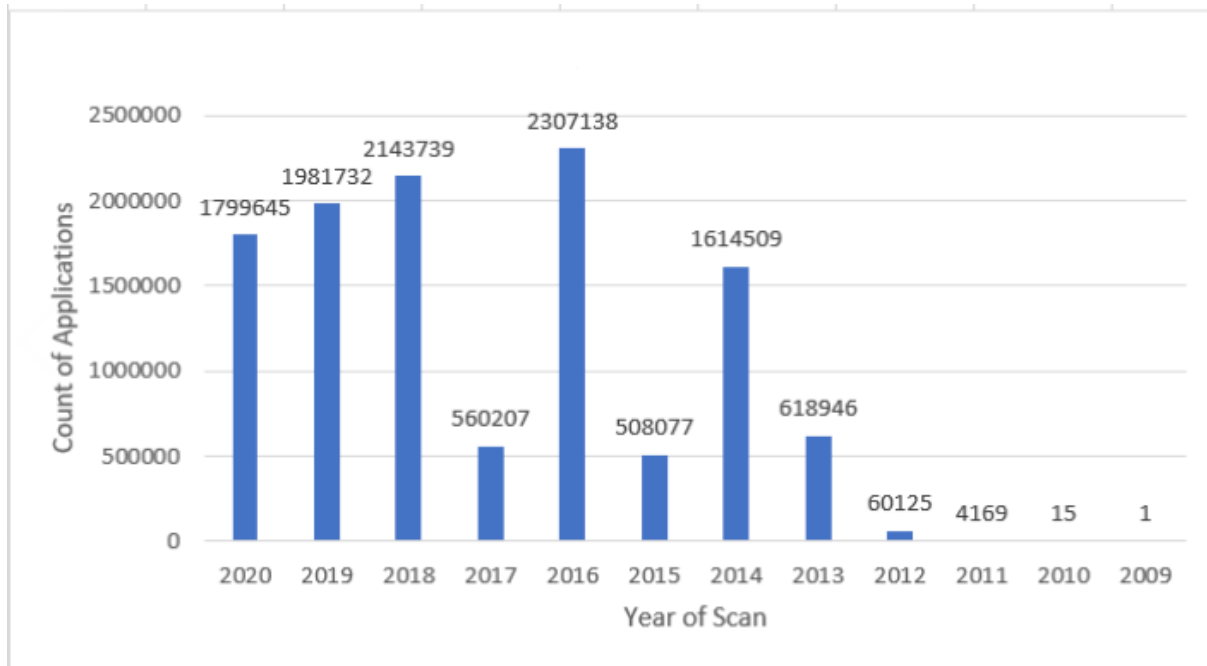


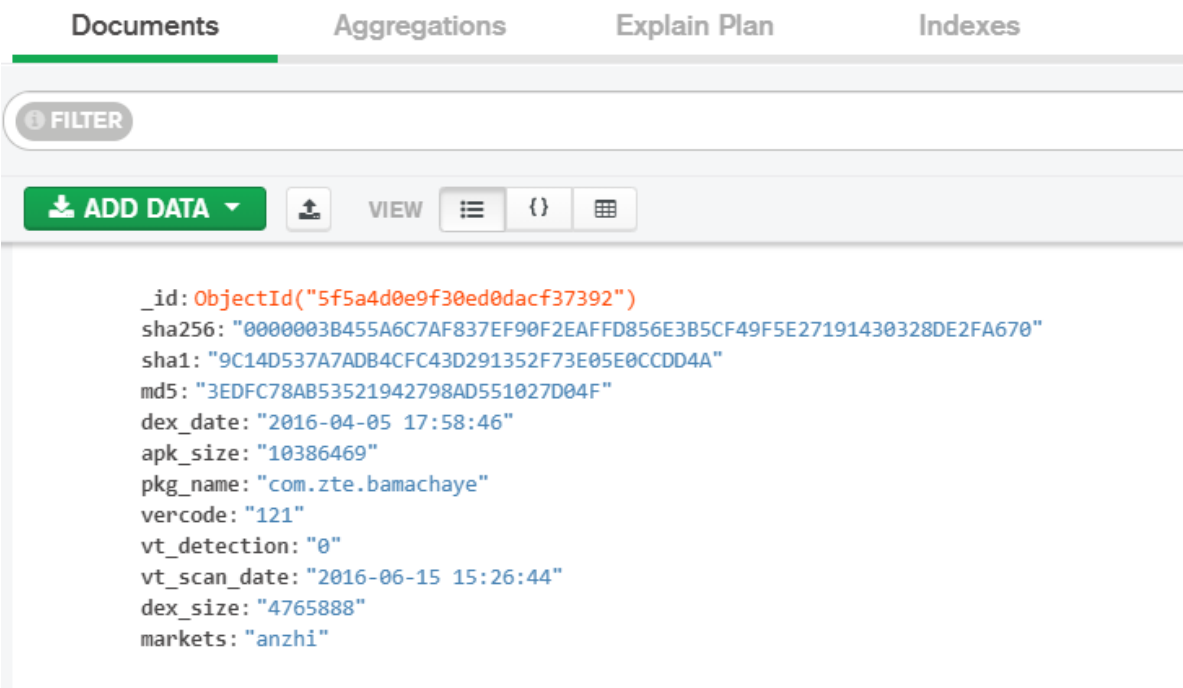
Figure 2.7: *Distribution of apps by the year of their VT scan.*

A MongoDB database was created for the file received from AndroZoo. This helped in analyzing the structure of the file, the information it contains, and to run queries to receive the results. An entry corresponding to an AndroZoo app is shown in Figure 2.8. The MongoDB query in Figure 2.9 was used to get the count of applications based on the scan date.

The data corresponding to an application contains the “package name” of the application, instead of its URL (following the naming convention of the Android application market). Thus, to match and compare it with the list of applications mentioned in Twitter (application URLs), it was required to reverse the package names, and thus produce identifiers in the form of URLs. For instance, one of the records in the AndroZoo database is “com.deperu”, a package name, which should spell backward as “deperu.com” to make the matching script work. Therefore, a Python script was developed to convert the package fields in a file to URLs. Below are the steps followed:

1. A file was extracted from the Mongo database in CSV format. An example of the results retrieved originally is displayed in figure 2.10.

twitter.URL Labels-AndroZoo



The screenshot shows the MongoDB interface with the following elements:

- Navigation tabs: Documents (selected), Aggregations, Explain Plan, Indexes.
- A FILTER button.
- A toolbar with an ADD DATA button, an upload icon, and VIEW options (list, JSON, grid).
- A document entry with the following fields:

```
_id: ObjectId("5f5a4d0e9f30ed0dacf37392")
sha256: "0000003B455A6C7AF837EF90F2EAFFD856E3B5CF49F5E27191430328DE2FA670"
sha1: "9C14D537A7ADB4CFC43D291352F73E05E0CCDD4A"
md5: "3EDFC78AB53521942798AD551027D04F"
dex_date: "2016-04-05 17:58:46"
apk_size: "10386469"
pkg_name: "com.zte.bamachaye"
vercode: "121"
vt_detection: "0"
vt_scan_date: "2016-06-15 15:26:44"
dex_size: "4765888"
markets: "anzhi"
```

Figure 2.8: MongoDB entry corresponding to an AndroZoo file

```
{vt_scan_date : {$gte : ("2020-01-01 00:00:00")}}
```

Figure 2.9: MongoDB Query to get the count of apps based on their Scan date

2. Package names were converted into URLs using a Python script. Figure 2.11 shows an example of the results received from this step.
3. Another Python script was written to match the file containing links from AndroZoo with the file containing links identified in the Twitter database that I created. This returned the matching results, i.e. number of URLs from the Twitter database found in AndroZoo.
4. The results obtained were also verified using a VLOOKUP formula in the excel sheet.

pkg_name
bmthx.god102409paperi
cinema.release.dates
com.tadu.android.androidread
cn.dreamjet.newplanet.android
com.egorkudraviy.JigsawPuzzleDakarKamaz
com.colorme.game.jisushuzipai
com.kezhan.buildAndsign
com.wSpeedtest1A

Figure 2.10: *Package names of applications in AndroZoo*

```

pkg_name
god102409paperi.bmthx
dates.release.cinema
androidread.android.tadu.com
android.newplanet.dreamjet.cn
JigsawPuzzleDakarKamazTruck.egorkudraviy.com
jisushuzipai.game.colorme.com
buildAndsign.kezhan.com
wSpeedtest1A.com

```

Figure 2.11: *Package names converted to links*

2.5 Data from Google PlayStore

The Google PlayStore data is also used in the analysis in this thesis. This is to get insights into the number or percentage of applications available on the Google platform, ones which are present in the file created after matching URLs in tweets with that of AndroZoo to find out if the apps common to both Google PlayStore and tweets are either malware or benign. Since the PlayStore is most widely used for downloading applications on the Android phones and is also one of the most trusted sources for Android apps, the presence of any malicious applications there can pose a significant threat to the security of the platform.

Figure 2.12 explains that the initial file handled was from AndroZoo and it was used to obtain ground for further analysis. URLs from tweets were compared with apps in the AndroZoo file. The results obtained from this, i.e., labeled apps, were then compared with the Google PlayStore data.

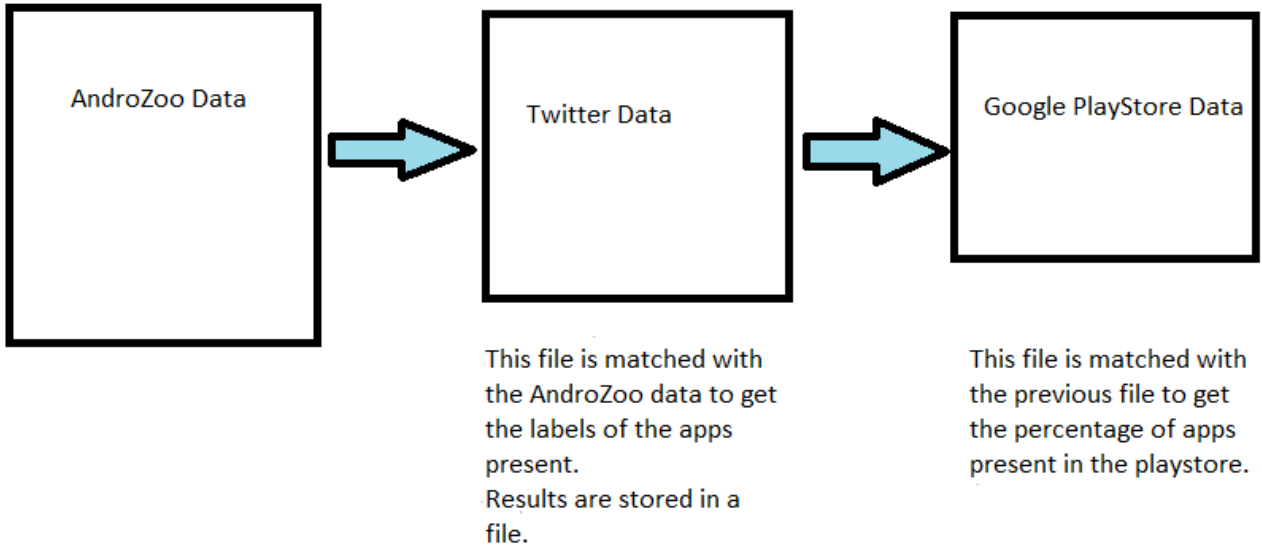


Figure 2.12: *Work-flow of the project*

Towards these goals, a crawler was developed to crawl web-pages from Google Play Store. This crawler collected the links of the applications present in the market. The crawler was developed using the Scrapy library of Python3 on the Computer Science department’s server. The following steps were performed:

- A project was created initially with the name “googlePlay” using the below command:
python3-scrapy startproject googlePlay

This came with a configuration file and a subfolder with the name “spiders” along with other Python files including items, middlewares, pipelines, and settings.

- The code developed for crawling was kept inside the spiders sub-folder.
- Following is the command that was used to run the file and obtain the desired output.
python3-scrapy crawl googleplayapi

- An output file was created with the name “googleplay-api” in the text format, inside the main project, after the successful execution of the crawler.

This file contains the URLs of the Android applications available on the Google Play-Store. Below is an example of a few of the results received from running the crawler:

https://play.google.com/store/apps/details?id=com.facebook.katana,com.facebook.katana
https://play.google.com/store/apps/details?id=com.instagram.android,com.instagram.android
https://play.google.com/store/apps/details?id=com.zhiliaoapp.musically,com.zhiliaoapp.musically
https://play.google.com/store/apps/details?id=com.twitter.android,com.twitter.android
https://play.google.com/store/apps/details?id=com.spotify.music,com.spotify.music
https://play.google.com/store/apps/details?id=com.pinterest,com.pinterest
https://play.google.com/store/apps/details?id=com.tocaboca.tocalifeworld,com.tocaboca.tocalifeworld
https://play.google.com/store/apps/details?id=com.roblox.client,com.roblox.client

Figure 2.13: *Application URLs from Google PlayStore*

This data received was also stored in the database. The total number of applications on the PlayStore currently is 2.8 million⁸ and the list received contains the URLs of around 700,000 applications. This is because of the restriction imposed by Google in terms of access to the Developer’s account. This list was then compared with the list of the malicious applications to determine the percentage of those available on the platform. To facilitate the matching, initial lines were removed and the one displaying the name of an app was then stored in a separate file. For instance, in

“https://play.google.com/store/apps/details?id=com.whatsapp,com.whatsapp”,
the string “https://play.google.com/store/apps/details?id=” was removed and the later part was kept i.e., “com.whatsapp”. Since the later part also denoted the package name, the Python script used earlier was re-used over the file to reverse its string data to transform it into a URL field.

2.6 User’s metadata

Additionally, I looked into the user’s information, in particular, the information of tweets that contain URLs of malicious applications. This was done to draw some conclusions on the user behavior, and detect anomalies occurring.

Chapter 3

Analysis and results

3.1 Analysis of Twitter app URLs

The URLs segregated from the Twitter data were stored in a separate file. Query shown in Figure 3.1 was used on MongoDB to extract the URLs.

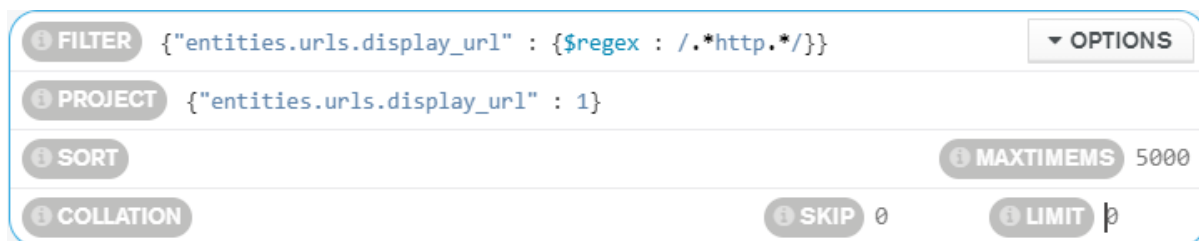


Figure 3.1: MongoDB query to extract URLs from Twitter data

URL is a field inside the “user” object. Hence while querying, “URL”: example.com cannot be used directly. For the extraction process to be accurate, “Http*” was used as a keyword (all URLs start with either HTTP or HTTPS), so that all fields starting with this keyword can be retrieved. Figure 3.2 shows the result obtained on running the query:

The result is shown in tabular form to make it more readable. MongoDB provides an option to view the result either in JSON format or tabular form.

Out of 9.1 million tweets obtained by crawling with the specified keywords, 2.3 million of the tweets had the URL field not null. The exact numbers are shown in Table 3.1 below.

	tweets	entities { }	urls []	urls.0 { }
		<code>didp0byentIdString</code>		
7	<code>5eba1cd96c50a33ea26b4176</code>			<code>"google.com/url?q=https://..."</code>
8	<code>5eba1cdc6c50a33ea26b5c6b</code>			<code>"fkdeals.app/?l=https://sto..."</code>
9	<code>5eba1cdc6c50a33ea26b60e1</code>			<code>"fkdeals.app/?l=https://sto..."</code>
10	<code>5eba1cdf6c50a33ea26b7749</code>			<code>"b.hatena.ne.jp/entry?url=http..."</code>
11	<code>5eba1cdf6c50a33ea26b7eb7</code>			<code>"google.com/url?q=https://..."</code>
12	<code>5eba1ce16c50a33ea26b93f2</code>			<code>"mirrativ.page.link/?link=https..."</code>
13	<code>5eba1ce16c50a33ea26b95f6</code>			<code>"fkdeals.app/?l=https://sto..."</code>
14	<code>5eba1ce36c50a33ea26bac3a</code>			<code>"mirrativ.page.link/?link=https..."</code>
15	<code>5eba1ce66c50a33ea26bc9af</code>			<code>"mkj8v.app.goo.gl/?afl=https%3A..."</code>
16	<code>5eba1ce66c50a33ea26bcac8</code>			<code>"b.hatena.ne.jp/entry?url=http..."</code>

Figure 3.2: Results of running the query on MongoDB

Total Tweets	Tweets containing URL
9,100,980	2,318,337

Table 3.1: Count of total tweets versus tweets containing URLs

On sanitizing the data and removing the duplicate entries of URLs, the field obtained with unique URLs was then kept for further studies.

The same steps were performed with the data received from March 15, 2020, to May 21, 2020. Out of three million tweets received, 72,314 URLs extracted from the tweets' field, were found to be unique. The graph in Figure 3.3 shows the top 10 links/applications which were most tweeted in the time frame of Mid March 2020 to May 2020. As can be seen, `images.app.goo.gl` or "Google Photos" was found to receive the maximum mentions, followed by "Apple apps".

3.2 Analysis of AndroZoo data

In referring to Figure 2.7, we can conclude that 67.40% of the applications on AndroZoo were scanned on and before 2018. Many of the applications labeled as malicious here do not

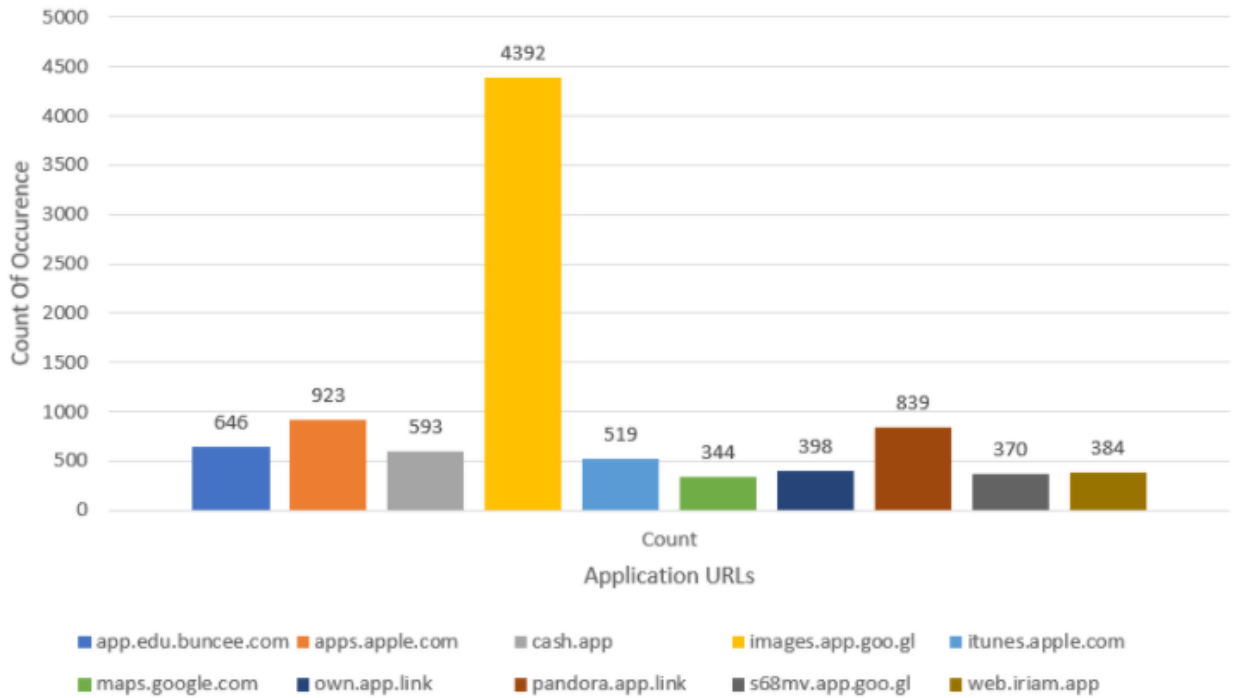


Figure 3.3: *Most frequently tweeted application links*

exist in the markets at the present date, meaning they are no longer available to download and generally not mentioned on Twitter, or other social media platforms, anymore. This is the main reason behind the low percentage of match between the URLs extracted from Twitter and the data from AndroZoo.

The data on AndroZoo was divided into two categories:

- Malware: VT Detection count is greater than or equal to 1.
- Benign: VT Detection count is 0.

Figures 3.4 and 3.5 show the queries used to get the count of each category.

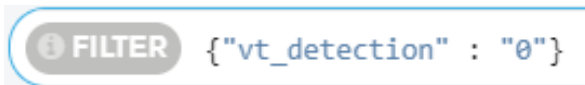


Figure 3.4: *MongoDB Query to get the results for VT Detection count equals to zero*

On running the above queries, we received the count of each malicious and benign applications on AndroZoo.



Figure 3.5: *Mongo Query to get the results for VT Detection count greater than Zero*

- Number of Benign Applications: 9,079,451
- Number of Malicious Applications: 2,518,852

Following were the results obtained on comparing these two files from AndroZoo with that of URL file received from Twitter:

- Count of Benign Applications = 21200
- Count of Malicious Applications = 9750

The application names were unique. Fields with the duplicate values were removed before performing the compare operations.

The applications found were then labeled accordingly. In conclusion, the percentage of applications found, both malware and benign, on AndroZoo represent one percent of the total apps, as depicted in Figure 3.6.

3.3 Analysis of Google PlayStore data

The applications labeled as “malicious” or “benign” were also compared with the database of Google PlayStore and the following results were obtained;

- Count of Benign Applications = 1971
- Count of Malicious Applications = 19

In conclusion, the percentage of applications found on Google Playstore with that of labeled apps derived from Twitter = 6.5%.

Figure 3.7 below shows the total count of applications present in AndroZoo, labeled as malware or benign versus count of applications found and labeled in Twitter after matching

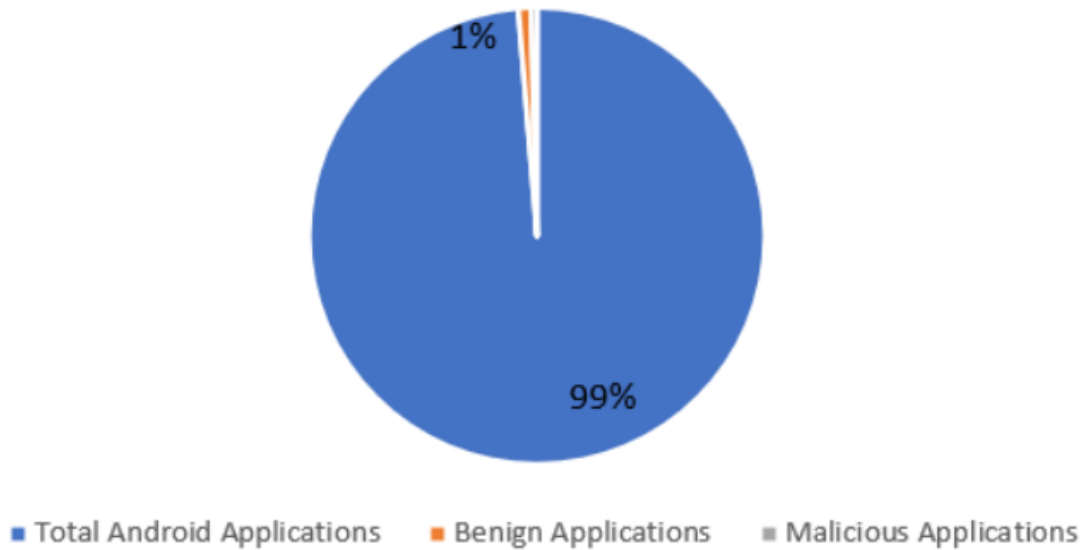


Figure 3.6: *Number of malicious and benign applications labeled vs total application links on Twitter*

with AndroZoo data. Finally, the count of applications received after comparing the tweets data with that of AndroZoo was compared with Google PlayStore data, and the count of applications found there is shown.

The malicious apps found in Google PlayStore had a VT detection count of less than 2. Few of the genuine apps such as Instagram and Whatsapp were also given the VT detection count of 1. This might be because of the few spam pages these apps contain, which are designed by the users of these platforms.

3.4 Analysis of user behavior

I further studied the user data whose tweets contain the mention of the Android applications, which are labeled as “malware”. This was done to draw some conclusions about the user behavior and anomalies occurring if any. To serve the purpose, the following fields from the Twitter data user objects, are taken into consideration: *protected, verified, followers_count, friends_count, favourites_count, statuses_count, geo_enabled and*

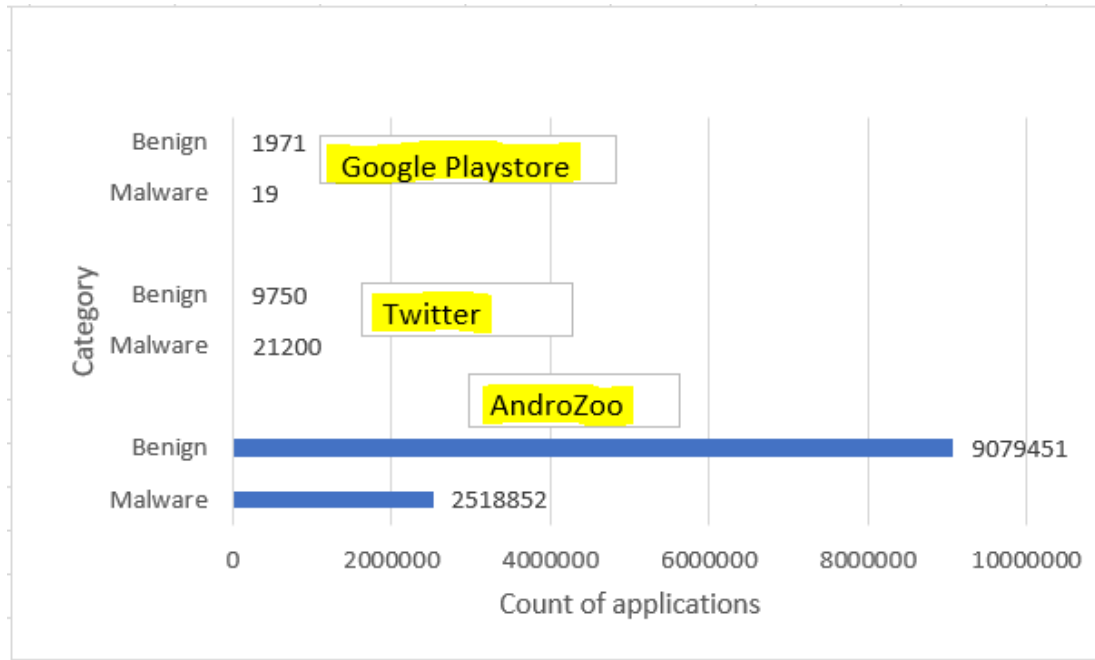


Figure 3.7: Total Number of apps matched on each platform

default_profile. Details about each of the fields are described in Table 2.1.

These user attributes were compared with each other and the results are shown in 3.2. To facilitate the comparison, users have been categorized as below:

- Category “A”: Users whose tweets contain the applications labeled as Malware.
- Category “B”: Users whose tweets contain the applications labeled as Benign.

Despite some differences in terms of user attributes, between users whose tweets contain malware apps and users whose tweets contain benign apps, no strong conclusions can be inferred regarding the characteristics of the malware-posting users. Intuitively, it makes sense that the malware-posting users try to look as similar as possible to the benign-posting users, so that evade detection. However, it may still possible to detect them with more sophisticated models that identify relationships between attributes, using machine learning. This will be studied in future work.

fieldName	Category “A”	Category “B”
protected	Out of all 9750 unique users, none of them was found to be “protected”, which means their tweets are visible to everyone irrespective of whether they have a Twitter account or not	Only 1% of the users had set their tweets to be protected
verified	91.2% of the user’s accounts were not verified	76.7% of the user’s accounts were not verified
followers_count	As per the data, 707 is the average number of followers an account has on Twitter ⁹ . 14.8% of the users in our database have followers more than 500	About 28% of these users had this count greater than 500. This data is compared in figure 3.8
friends_count	Around 46.3% of these users had friends more than the 350	46.1% of the users had friends more than 350
favourites_count	On average, a regular Twitter user likes 20,000 tweets over a period time ⁹ . From the data collected by running queries on MongoDB, it was concluded that about 69.2% of the users’ favorite count, lies in the range	67.1% of these users had a favorite count greater than 20,000
statuses_count	Around 70.4% of these users’ status count comes to be less than 50	54.5% of the users had a status count less than 50
default_profile	About 47.2% of the users had kept their default profile as “TRUE”	67% of these users had their default profile “TRUE”
geo_enabled	52.5% of the users in the database had their geological location enabled	32% of the users had enabled their geological location
lang	English: 78.1% Spanish: 4.2% Arabic: 2% Japanese: 2%	English: 55.3% Spanish: 5% Japanese: 3.2% Arabic: 1%

Table 3.2: *User behavior comparison based on field names*

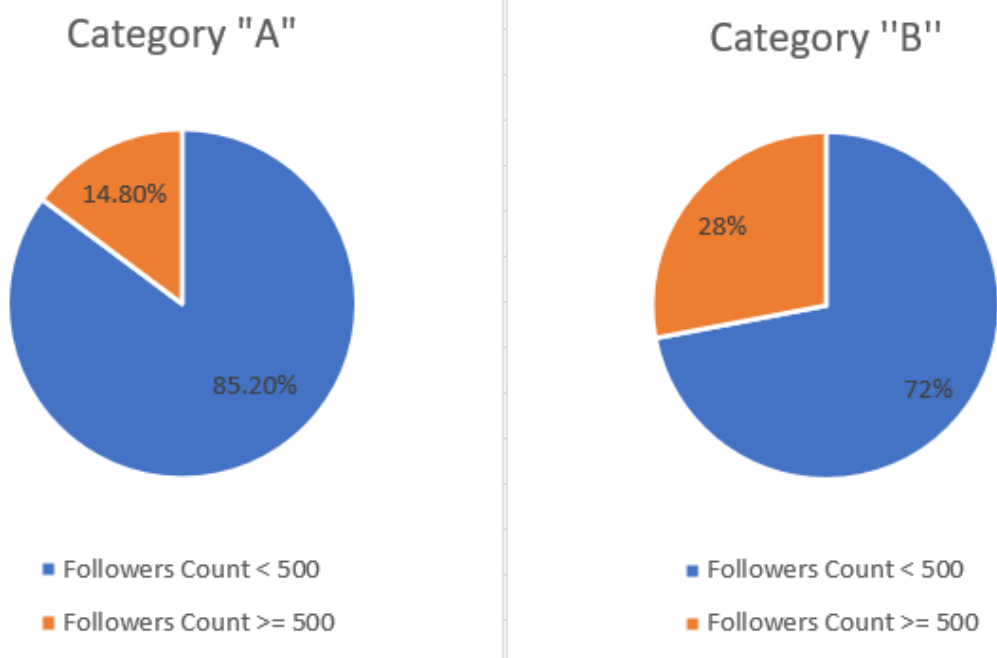


Figure 3.8: *Count of followers of an account whose tweets contain malware app links.*

Chapter 4

Conclusions and future work

In this thesis, I matched the Android applications' URLs from the data available at the AndroZoo and Google PlayStore. Those applications which were found in either of the platforms were then compared and labeled based on the AndroZoo VT detection count. AndroZoo is a trusted platform used to deduce the accurate label of the application. Hence the results from it were taken as ground truth in the analysis. A total of 30,950 application links matched with the AndroZoo database. Among these, 9,750 applications were labeled as malicious and the remaining 21,200 were labeled as benign. Furthermore, I filtered the tweets which contained the mention of malicious Android applications and studied their user's metadata. This approach was taken to study the user's characteristics and to form a pattern based on anomalies. It was found that the majority of the users were neither protected nor verified on Twitter. However, similar data was found for the users who posted links about benign applications. Hence, we can conclude that single features extracted from the user object, can not alone predict the nature of a user on Twitter. This pattern in later stages can be combined with the Machine Learning classifiers to classify Twitter users as either spam, bot, or genuine.

There is still a large number of applications that are left unlabeled. These applications are either new to the market or have not caught the attention and are left undiagnosed. These could be submitted to AndroZoo or VirusTotal or other trusted antivirus vendors to

depict the count of detection.

Python scripts used in the project can be found at [**https://github.com/grajawat/Twitter**](https://github.com/grajawat/Twitter)

Bibliography

- [1] Tianyi Gu. Newzoo’s global mobile market report: Insights into the world’s 3.2 billion smartphone users, the devices they use and the mobile games they play. 2019. URL <https://newzoo.com/insights/articles/newzoos-global-mobile-market-report-insights-into-the-worlds-3-2-billion-smartphone->
- [2] J. Clement. Number of available applications in the google play store from december 2009 to september 2020. 2020. URL <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>.
- [3] Ying Lin. 10 twitter statistics every marketer should know in 2020 [infographic]. 2020. URL <https://www.oberlo.com/blog/twitter-statistics#:~:text=Here's%20a%20summary%20of%20the,are%20between%2035%20and%2065.>
- [4] Joshua Roesslein. Hello tweepy. 2020. URL http://docs.tweepy.org/en/v3.9.0/getting_started.html#hello-tweepy.
- [5] Jikai Tang. Trendsmap a real-time us trends map for twitter. Master’s thesis, Brown University, 2016.
- [6] Twitter. Data dictionary. 2020. URL <https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview/user-object/>.
- [7] W3Techs. Androzoo, 2016. URL <https://androzoo.uni.lu/>. Last accessed 16 September 2017.
- [8] Ian Blair. Mobile app download and usage statistics (2020), 2020. URL <https://buildfire.com/app-statistics/#:~:text=There%20are%202.8%20million%20apps,on%20the%20Google%20Play%20Store.>

[9] Kit Smith. 60 incredible and interesting twitter stats and statistics, 2020.

URL <https://www.brandwatch.com/blog/twitter-stats-and-statistics/#:~:text=As%20of%20Q1%202019%2C%2068m,have%20no%20followers%20at%20all.>