Team Android Malware – Security Metrics

Azqa Nadeem – 4542606, Ioannis Schistos – 4633253,
Pouya Omid Khoda – 4625323, Maurits Mulders – 4204301

Introduction

The given dataset contains information regarding games and software that has been uploaded on Baidu [5] and 360 [4], two of the most popular android app stores in China. For every app, it contains information regarding its category, software version, size, software package, number of downloads, download URL, last update date, and download date.

Assumptions

We make the following assumptions for the remainder of the document:

- We assume that the difference between 'last update date' and 'download date' gives the number of days it has been since the last update was released.
- We assume that the apps which were last updated more than 50 days ago are malicious.
- The term 'infection' has been used to refer to a device being hijacked by a malware.
- If an actor is using a platform, it implicitly states that he/she is downloading apps from it.

Victim/Actor

We consider the victims to be children aged between 8-15 years who own an android device and play a lot of games.

Security issue

This dataset concerns with the danger of installing an infected app for a consumer/actor based on the maliciousness of the 3rd party android app platforms. In the context of the victim that we have chosen, the security issue is the danger that an android app platform exposes the children of a certain age group to, who play a lot of games. The danger level may vary based on the category of apps the victim is interested in and the platform he/she uses to download apps from.

Ideal metrics

In the context of the security issue, the following are some ideal metrics that we would like to have:

- *The risk for a consumer of a platform*
  This metric will be measured as the probability of that consumer getting infected multiplied by the impact for that consumer if he/she were infected. The probability and the impact themselves are ideal metrics. The probability will be measured as P(infection | category) which is the probability of getting infected given the set of categories of apps that a consumer is interested in. The impact will be different based on the type of infection (spyware, data exfiltration, ransomware, etc.) and who the victim is (business man, student, researcher, etc.).

- *The mitigation cost of a platform*
  The mitigation cost of an app is a lower level metric that will measure how much it would cost to recover from an infection caused by an app on a platform. For example, the cost of mitigating a ransomware for the actor under consideration might be none since there is no critical data on a child's phone, but the cost of spyware can be significant as it causes a privacy breach, not just for the child but for his/her parents as well. This metric can then be generalized for all the category of apps on a platform that a certain consumer is likely to download to arrive at the total mitigation cost of using a platform. For example, the actor under consideration might incur more mitigation cost on platform A because most of the games are infected by spyware, as compared to platform B where the apps are regularly patched.
- *Indicators or feature set of an app that accurately determine whether it is malicious or not*
- *The security trend of a platform based on past infections caused by apps hosted on a platform*
  This metric will measure, over a period of time, the increase/decrease in getting an infection from using a platform to download apps given a history of infections caused because of the use of that platform. It can be used to check the trend of the platform in terms of its security practices over a period, given all other variables remain constant. For example, a person gets infected 50% less than he got infected last year by downloading apps from Platform A shows that the security practices of platform A are improving.
- *Survival time of a malware*
  This metric would measure the average survival time of a malware in an app on a platform. It would be calculated as the difference between the time when the infection was first discovered and the time the developers patched their app. This would give the actor under consideration a time estimation of how long a certain category of apps can take to be safe enough to be downloaded.
- *Estimated number of affectees*
  This metric will estimate the number of infected consumers based on the distribution of app downloads over app updates. Essentially, it means that we would have the number of downloads of an app after every update of the app. Then, based on which update made the app vulnerable, we can calculate the number of consumers who were affected by it.

Metrics in Practice

Based on the security issue we mentioned earlier, there are metrics that exist and may be used to measure different aspects of the maliciousness of an app market.

- The risk [2] of an infection from an app is a common metric. More specifically, the risk that a user faces is calculated based on the probability of the user getting infected by a malicious app and the impact of that infection. The impact depends on the kind of infection.
- Another metric that is being used in practice is the response time in case of an infection. More specifically, it measures the response time of an app's developers to remove a vulnerability from the app. Such metric shows the time gap in which a user is vulnerable against a security threat. In practice, this metric can be decomposed in 3 different sub-metrics [3] that show the reaction chain within an app's development team. First is the mean time to alert, which describes the time between the incident and the actual alert. Then we can compute the mean time to investigation of the event to understand the reason behind the security incident. Finally, we compute the mean time to remediation. Based on all these metrics a general response time is calculated.
- Lockheed Martin also developed a similar metric. It includes the mean time to incident discovery, mean time to incident recovery, and the number of incidents. They are used to measure the

incident management of a team [6] [7]. These metrics, in particular, would be applicable on the app development team rather than the victim under consideration.

- On the other hand, metrics such as mean time to mitigate vulnerabilities and the number of known vulnerabilities are used to measure the vulnerability management [6] [7], which can be used to estimate the survival time of a danger to the app's users.

Methodology

In this section, we first describe the data cleaning and preprocessing steps we took and then describe the metric we have designed for our dataset.

Data cleaning and processing steps

Before starting any data analysis task, it is important to clean the data and to prepare it for processing because often times, the data is not in the format that is ideal for application. We had a similar issue. Therefore, we performed the following data cleaning steps.

- *Removed non-downloadable apps*

The apps that had a status of 'non-downloadable' in Download result were removed as they not only had some missing fields e.g. update date but it was also difficult to identify the reason why an app is non-downloadable, i.e. whether there was some problem with our internet connection or the app itself was flawed.

- *Translated category to English*

Since the dataset was created from two Chinese app markets, the app names and categories were in Chinese. In order to perform some meaningful analysis on the data, it was important for it to be in a language that was understandable to us. Therefore, we translated the categories to English. Some categories were similar but translated differently, e.g. in 360 games, one category was translated to "Chess world" and on Baidu games, a category was translated to "Chess tour". Considering the possibility that they might be referring to the same general category "Chess", we named both categories as "Chess" to allow comparisons between the two platforms.

- *Fixed data types*

The datetime string was changed to the corresponding data type in order to perform operations on dates.

- *Changed app sizes to a uniform metric*

The app sizes in the dataset were a mix of GBs, MBs, and kBs. We converted all the sizes into MBs by dividing kBs with 1024 and multiplying GBs with 1024.

Metric

- The danger for a consumer of a platform:

As the name suggests, this metric measures the 'danger of getting an infection' a victim is in for using a platform. This will be calculated by a formula composed of:

1) A set of categories the victim under consideration is interested in,
2) Percentage of malicious apps per category,
3) Popularity of that category (number of downloads of apps in that category/number of apps in that category)
4) Size of the platform (normalized by the total number of apps on that platform)

$$danger = \frac{\sum_{c \in C} M_c * P_c}{N}$$

Where: C is the set of categories of interest,

M_c is the fraction of malicious apps in c,

$P_c = \frac{\#\ downloads}{\#\ apps}$ measures the number of downloads controlled by the number of apps in each category,

N is the number of apps on a platform

- The resulting value of this metric will be a number.
- A scale will be generated per platform with minimum and maximum danger values. Minimum danger is when there is no malicious app on a platform. Hence, for min danger value, we set the fraction of malicious apps to 0. Similarly, for maximum danger value, the fraction of malicious apps is set to 1. Then, the scale will be divided into 3 equal regions, each showing low, medium, and high danger for the victim.

Evaluation

In this section, we will evaluate the metric described above in the context of the victim considered - a child

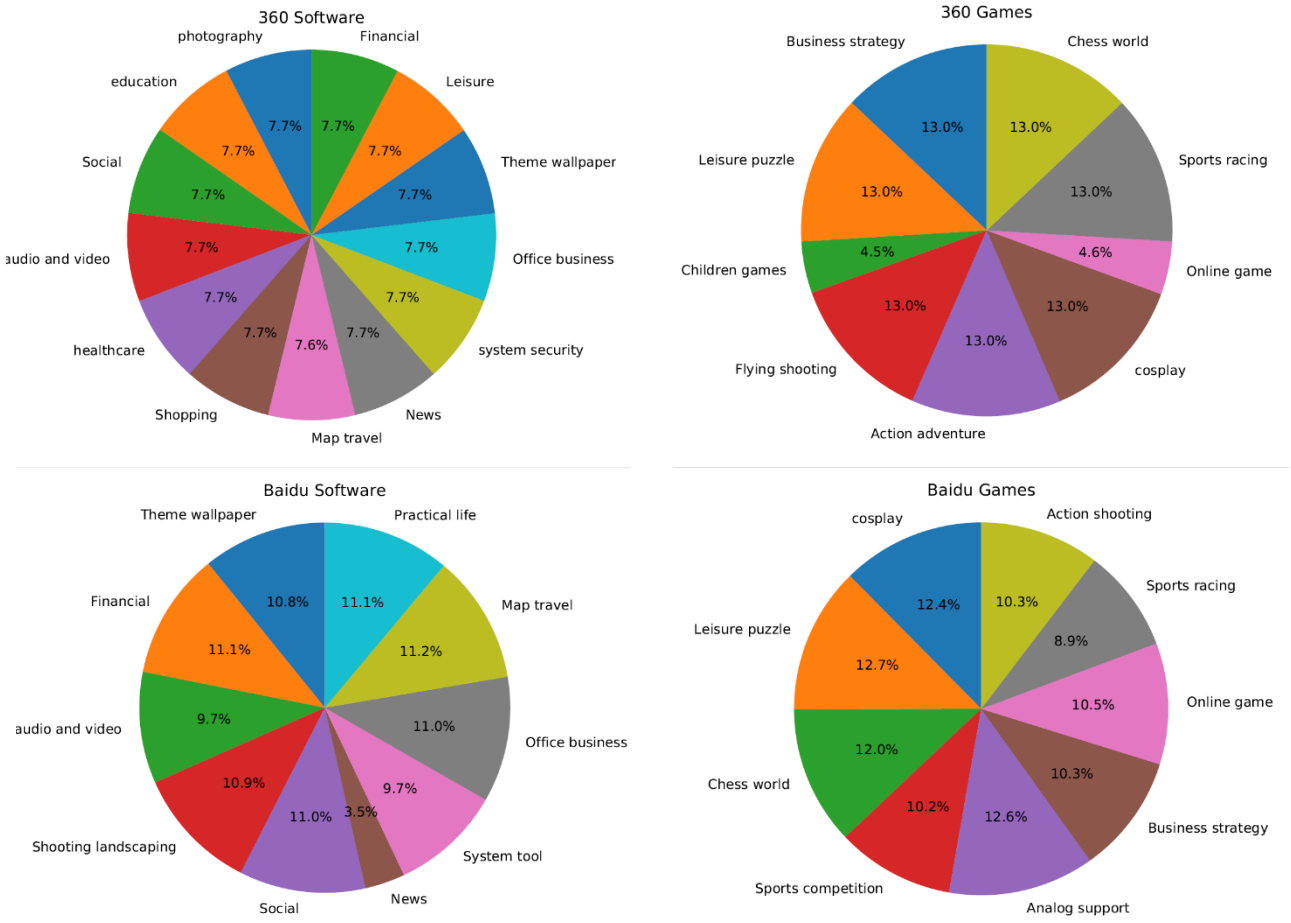


Figure 1: App distribution per category for Baidu and 360

aged between 8-15 years who plays a lot of games. We will evaluate the two app platforms (360 and Baidu) to see which one is more dangerous for that victim.

Figures 1 show the app distribution per category for each of the two platforms. We assume that the victim will be interested in games belonging to categories {sports racing, online games, cosplay, action adventure, flying shooting} on 360 platform and {sports racing, online games, sports competition, cosplay, action shooting} on Baidu platform.

Metric evaluation for 360

Table 1: Statistics for 360

Categories	# apps	# downloads	# malicious	Fraction of malicious apps	Popularity	Danger per category
Action adventure	2449	409564650	2301	0.9395671703	167237.5051	157130.8694
Sports racing	2443	525513456	2313	0.9467867376	215109.8878	203663.1889
Online games	872	273871749	756	0.8669724771	314073.1067	272292.7393
Cosplay	2447	576133612	2076	0.8483857785	235444.8762	199748.0846
Flying shooting	2449	469504514	2297	0.9379338506	191712.7456	179813.8737
					Total danger	**20.01677715**
					Max danger	**22.20949044**

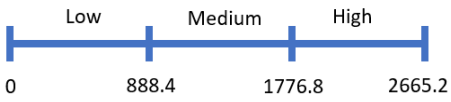
Table 1 shows the statistics and the data based on which the danger metric is calculated. As mentioned previously, each row refers to one category that the victim under consideration is interested in. The second column shows the count of apps in each of those categories. The third column shows the total number of downloads of the apps in that category. The fourth column shows the number of malicious apps in that category based on the assumption that apps that have not been updated for more than 50 days are malicious. Then, we calculate the fraction of apps that are malicious per category calculated by (number of malicious apps/number of apps). We also calculate the popularity of each category by (number of downloads/number of apps). As the danger formula suggests, we multiply the fraction of malicious apps per category with the popularity of that category and sum over all the categories of interest. Finally, we divide the result by the number of apps on the platform to control for the size of that platform. For 360 platform, we get the danger value of 20.01.
In order to put this number into perspective, we calculate the min and max value of the scale. The min danger value is when the fraction of malicious apps per category is 0. Hence, the overall min value on the scale would be 0. The max danger value is when the fraction of malicious apps per category is 1. Using the same formula, we get the value of 22.2. Using the scale in the figure below, we see that the victim under consideration is in *high* danger of getting an infection by using 360 platform to download games.

Metric evaluation for Baidu

Table 2: Statistics for Baidu

Categories	# apps	# downloads	# malicious	Fraction of malicious apps	Popularity	Danger per category
Sports racing	199	178781585	134	0.6733668342	898399.9246	604952.7131
Online games	239	930984597	190	0.7949790795	3895333.042	3096708.276
Cosplay	281	806580494	115	0.409252669	2870393.217	1174716.085
Sports competition	227	99758683	197	0.8678414097	439465.5639	381386.4145
Action shooting	232	1131281438	148	0.6379310345	4876213.095	3110687.664
					Total danger	**1718.367793**
					Max danger	**2665.257668**

Table 2 shows the same statistics for Baidu platform. The danger metric comes out to be 1718.4. The min danger value is 0 and the max danger value is 2665.2. Putting these values on a scale, as shown by figure below, we see that the victim using Baidu to download games is under *medium* danger of getting an infection.



Conclusion:
The victim under consideration is better off downloading apps from Baidu rather than 360 because of the relatively low danger of getting infections.

Limitations and Conclusion

This section discusses some limitations of our dataset, metric, and analysis. Finally, we give some concluding remarks about the work.

- We filtered out some data points with missing values. It might have been interesting to evaluate apps that had never been updated, or to include that data in some way.
- The dataset did not give us much to go on so we improvised and came up with scenarios ourselves. For example, some metadata was provided to us with information about how malicious an app was. Unfortunately, those apps were from a completely other platform, so we could not utilize that dataset at all. Finally, we improvised by assuming that apps that were last updated more than 50 days ago are malicious. In this way, we incorporated some security related information in the dataset.
- Our metric and the actor have been built on a set of assumptions which might be completely different from the real-world scenarios
- We have tried to control for the size of the platform in the calculation of the metric. However, other factors such as the number of downloads or the underlying infrastructure has not been

taken into account. Hence, two platforms which have the same number of apps, but one having higher number of downloads than the other will receive the same danger value, while in fact, the danger of using the more popular platform is higher than the other.

- The way the scale for min and max danger values has been developed is crude. We have tried to keep the evaluation simple for understanding purposes. For example, the division boundaries of low, medium and high danger can be based on smarter heuristics rather than just being divided equally.

In conclusion, the dataset assigned to us contained a number of apps collected from two of the biggest android app markets in China - 360 and Baidu. We assumed that a fraction of those apps was malicious. We then developed a metric that measured the level of danger of getting an infection a consumer is in for using each of the two platforms. These values can be compared and informed decisions about preferring one platform over the other can be made.

References

[1] Kikuchi, Y., Mori, H., Nakano, H., Yoshioka, K., Matsumoto, T., & van Eeten, M. (2016). Evaluating Malware Mitigation by Android Market Operators. In *CSET@ USENIX Security Symposium*.

[2] Holton, G. A. (2004). Defining risk. *Financial Analysts Journal*, *60*(6), 19-25.

[3] *Three Metrics That Measure Incident Response Performance ‹ Hexadite - Security Orchestration and Automation - Automated Incident Response*. (2017). *Hexadite.com*. Retrieved 24 September 2017, from https://www.hexadite.com/incident-response/three-metrics-measure-incident-response-performance/

[4] http://developer.360.cn/

[5] http://pcappstore.baidu.com/en/index.php

[6] Mateski, M., Trevino, C. M., Veitch, C. K., Michalski, J., Harris, J. M., Maruoka, S., & Frye, J. (2012). Cyber threat metrics. *Sandia National Laboratories*.

[7] https://www.nasa.gov/583318main_2011_Present_NASA_IT_Summit_Grandy_Serene_Implementing_Cyber_Security.pptx