Understanding the Security Management of Global Third-Party Android Marketplaces

Yuta Ishii[†], Takuya Watanabe[‡], Fumihiro Kanei[‡], Yuta Takata[‡], Eitaro Shioji[‡]
Mitsuaki Akiyama[‡], Takeshi Yagi[‡], Bo Sun[†], Tatsuya Mori[†]
[†]Waseda University, Tokyo, Japan, [‡]NTT Secure Platform Laboratories, Tokyo, Japan
{yuta,sunshine,mori}@nsl.cs.waseda.ac.jp,lastname.firstname@lab.ntt.co.jp

ABSTRACT

As an open platform, Android enables the introduction of a variety of third-party marketplaces in which developers can provide mobile apps that are *not* provided in the official marketplace. Since the initial release of Android OS in 2008, many third-party app marketplaces have been launched all over the world. The diversity of which leads us to the following research question: *are these third-party marketplaces securely managed?* This work aims to answer this question through a large-scale empirical study. We collected more than 4.7 million Android apps from 27 third-party marketplaces, including ones that had not previously been studied in the research community, and analyzed them to study their security measures. Based on the results, we also attempt to quantify the *security index* of these marketplaces.

CCS CONCEPTS

• **Security and privacy → Software security engineering**;

KEYWORDS

Mobile Application Market; Security Measures

ACM Reference Format:
Yuta Ishii, Takuya Watanabe, Fumihiro Kanei, Yuta Takata, Eitaro Shioji, Mitsuaki Akiyama, Takeshi Yagi, Bo Sun, and Tatsuya Mori. 2017. Understanding the Security Management of Global Third-Party Android Marketplaces. In *Proceedings of 2nd International Workshop on App Market Analytics, Paderborn, Germany, September 5, 2017 (WAMA'17),* 7 pages.
https://doi.org/10.1145/3121264.3121267

1 INTRODUCTION

Android is one of the most popular mobile device platforms in the world, with the number of Android apps available on Google play exceeding 2.9 million as of May 2017 [3]. The large popularity of Android also has attracted attackers as the target of malicious activities such as the uploading of malware or illegal piracy [5]. One of the reasons there are many malicious/cloned apps on Android is the openness of the Android ecosystem, which has enabled various third-party marketplaces to operate independently from the official Android marketplace – Google Play. We also note that in some areas, such as China and Cuba, access to Google Play has some restrictions [9]. In such areas, Android users need to rely on the third-party markets as their default gateway to acquire apps. As we will show later, third-party markets are operated by various organizations, including web service providers or hardware vendors.

The diversity of the third-party marketplaces operating globally has made it difficult to assess their security. Although there have been a large number of studies that have analyzed apps collected from third-party marketplaces, their overview has been limited to a few popular marketplaces. As we shall show later, there exists several marketplaces that have *not* received attention in the research community.

On the basis of this background, we aim to answer the following research question:
RQ: *Are these third-party marketplaces securely managed?*
To answer the question, we first collected over 4.7 million of Android apps from 27 third-party marketplaces. We also collected 15 K of paid apps from the official market, Google Play. We then analyzed them to assess their security measures, i.e., uploaded/scanned malware samples and duplicated apps among marketplaces. We defined a *security index* that aims to quantify the security measures of a market and then visualized the relationship between the security index and the popularity of the marketplace. Through this analysis, we derived the following findings. Apps published in some marketplaces, which have not been studied in the research communities, have not been exposed to the "dragnet" of online virus checkers. There are some intrinsic patterns of releasing/cloning apps among multiple marketplaces, and, finally, there are several marketplaces that, despite their high popularity, are insecure.

2 DATA

In the process of compiling a list of third-party marketplaces operated in various countries, we found that English searching is not always the best solution. We, therefore, recruited several people who are conversant in languages, including Chinese, Italian, German, Spanish, Turkish, Russian, Arabic, Vietnamese, Thai, Persian, etc. We asked the participants to report on the third-party marketplaces widely used in their countries/areas. In total, we collected a report involving marketplaces using 15 different languages. From the list of reported third-party marketplaces, we selected the 13 marketplaces that appeared to be important.

WAMA'17, September 5, 2017, Paderborn, Germany

ACM ISBN 978-1-4503-5158-4/17/09…$15.00
https://doi.org/10.1145/3121264.3121267

Table 1: List of marketplaces: 13 marketplaces at the top of the table are the ones we crawled and the 14 marketplaces/dataset at the bottom are from Androzoo dataset.

Market	# of APK	Language	Operator	Notes
(1) *Alandroid*	9,620	Arabic	Unknown	A market for Middle Eastern countries.
(2) *Appvn*	34,415	Vietnamese	Unknown	A market that provides mobile apps for multi-platform.
(3) *Aptoide*	138,421	multi-language	A company in Portugal	Users can establish their own markets.
(4) *Baidu*	13,020	Chinese	Baidu, a big enterprise developing portal websites	A region restricted to access Google Play.
(5) *Blackmart*	100,127	English	Unknown	Users download apps via its market client app.
(6) *Cafebazaar*	54,034	Persian / English	A company in Iran	A market in Iran. A region restricted to access Google Play.
(7) *Entumovil*	235	Portuguese	A company in Cuba	A market in Cuba. A region restricted to access Google Play.
(8) *Getjar*	38,180	English	A company in Lithuania	–
(9) *Mobogenie*	31,547	multi-language	A company in the U.S.A.	Users download apps via its client app for smartphones or for PC.
(10) *Mobomarket*	10,392	Indonesian / Thai / English	A company acquired by Baidu	Users download apps via its client app for smartphones or for PC.
(11) *Uptodown*	59,428	Spanish / multi-language	A company in Spain	A market that provides mobile apps for multi-platforms. Users download apps via its market client app.
(12) *Yandex*	22,964	Russian	Yandex, a big enterprise developing portal websites	Users download apps via its market client app.
(13) *Zhushou360*	204,417	Chinese	360, a security vendor in China	A region restricted to access Google Play.
Google Play	3,608,379	multi-language	Google	The official market.
Anzhi	736,517	Chinese	A company in China	A region restricted to access Google Play.
Appchina	593,128	Chinese	A company in China	A region restricted to access Google Play.
Mi.com	104,029	Chinese	Xiaomi, a big smartphone vendor enterprise	A region restricted to access Google Play.
1mobile	57,525	multi-language	Unknown	–
Angeeks	55,815	Chinese	A company in China	A region restricted to access Google Play.
Slideme	52,448	English / French	A company in the U.S.A.	–
Torrents	5,294	–	–	The data collected by BitTorrent.
Freewarelovers	4,145	English	A company in Germany	–
Proandroid	3,683	Russian	Unknown	–
Hiapk	2,510	Chinese	A company acquired by Baidu	A region restricted to access Google Play.
Fdroid	2,023	English	A company in England	The market that provides only open source software.
Genome	1,247	–	–	Not a marketplace. The malware dataset collected by Zhou et al.[21]
Apk_bang	363	Unknown	Unknown	Closed.

(1) https://www.alandroidnet.com/, (2) http://appvn.com/android, (3) https://www.aptoide.com/, (4) http://shouji.baidu.com/, (5) http://www.blackmart.us/, (6) https://cafebazaar.ir/, (7) http://www.entumovil.cu/downloads/apps, (8) http://www.getjar.com/, (9) http://www.mobogenie.com/, (10) http://www.mobomarket.net/, (11) https://www.uptodown.com/android/, (12) https://m.store.yandex.com/, and (13) http://zhushou.360.cn/.

The 13 markets we chose are listed in Table 1. The market names used on this table are chosen for convenience and can differ from their actual names. We collected data from June to September of 2016. The format of how apps are released varies by market (website or client application).

As the application protocol interface (API) also differs by market, we conducted reverse engineering for each market using selected tools [4, 7, 8] and developed crawlers. Additionally, we parsed metadata such as app documentation or download counts if they were available. To obtain these data, we primarily crawled apps listed in each market's rankings. It is unclear whether we were able to crawl the entirety of each market, and we will describe this problem in greater detail in Section 6. We collected about 800 thousand apps in total, including duplicate apps that were released to multiple markets.

In addition to the 13 marketplaces we described above, we also made use of the Androzoo dataset [11], which is a dataset used by researchers of Android apps from 14 different markets/sources that includes Google Play, the *Genome* malware dataset [21], and torrent files.

We used the Androzoo list of July 10th, 2016, which included about 4.3 million apps. Although this dataset does not include metadata for markets, it does includes the detection count of VirusTotal. Further details on the markets contained in this dataset can be found in the original paper [11].

In total, the size of the dataset we collected was about 45 TB. A breakdown of the dataset is summarized in Table 1. We computed the MD5 hash value for each Android application package (APK) and found a unique APK number of 4,761,283 and a unique package name of 2,827,578.

3 ANALYSIS OF MOBILE APP SECURITY

3.1 Dragnet Analysis

Our "dragnet analysis" provides a unique metric to app secureness by measuring the degree in which an app is exposed to the eyes of scrutinization; intuitively, it is based on the assumption that the more apps are seen and investigated by researchers, the safer they should be. In this work, we attempted to measure such a property by whether or not an app has already been scanned by online app scanning services in the past. Nowadays, the use of such services is quite prevalent as they are used not only in many research activities but also by general users via web interfaces. Furthermore, the results are often made public and are available through search engines, and uploaded files are shared across some organizations. We believe these points justify our choice.

Table 2: Fractions of unscanned apps.

Market	not-scanned fraction (%)
Cafebazaar	75.6
Yandex	20.6
Mobomarket	20.5
Baidu	19.0
Getjar	15.5
Appvn	14.1
Zhushou360	13.1
Alandroid	6.9
Aptoide	4.6
Mobogenie	2.2
Entumovil	1.7
Blackmart	1.7
Uptodown	1.2

The online scanner service we used is VirusTotal [10], which is an online anti-virus service that comprises more than 60 different commercial anti-virus checkers. We queried the hash value of each APK we crawled to VirusTotal in order to obtain a scan report for each. For each market, we calculated the percentage of apps that had been not scanned by VirusTotal at that time. Because the data from Androzoo have already been scanned, we ignored them in this analysis. The results of this analysis are shown in Table 2.

We found that several marketplaces such as *Cafebazaar*, which is a marketplace operated in Iran, had fewer fractions of apps that had been scanned with the online virus checkers. This observation indicates that many of apps in such marketplaces potentially have unfixed vulnerabilities.

3.2 Breakdown of Benign / Malicious Apps

Figure 1 shows the fractions of apps detected as malware for each market. For this analysis, we analyzed the apps, which had not been scanned with the VirusTotal before. We note that apps that were too large to be scanned are omitted. To determine if an app can be labeled as "malware", we use a conservative threshold; i.e., we consider any app detected 10 times or more is considered to be malware. We also note that we consider any app detected 10 times or more and with at least one adware label to be adware Because the data from Androzoo only include detection count (not with label), we cannot distinguish between adware and malware for this market, and we consider all of the Androzoo's apps shown here to be malware. Although we could obtain scan reports for such apps ourselves, we had to skip it due to the limitation of our resource and time.

As *Genome*, shown in the top part of the figure, is the dataset of malware, it is not surprising that the fraction of malware is 100%. This test shows us that the virus checkers we used worked correctly for this dataset. Overall, Chinese marketplaces, such as *Appchina*, *Anzhi*, and *Baidu*, have high fractions of malicious apps. In contrast, Google Play had the high fraction of benign apps. It is interesting that *Cafebazaar*, whose not-scanned rate was the highest, had the highest benign rate. This result indicates that a market may have a low malicious rate even if researchers have not previously examined it. Or, this could indicate that virus checkers used for VirusTotal missed the detection of malicious apps in the marketplace; we leave the study for our future work. We also see that *Fdroid*, an open source community, also had a high benign rate. This observation seems to be natural given the nature of the marketplace.

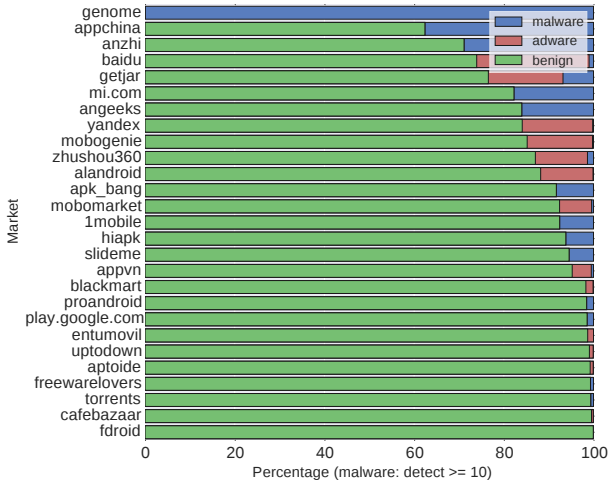

Figure 1: Fractions of malicious apps for each market.

4 CIRCULATION OF MOBILE APPS ON THE GLOBAL MARKETPLACES

Some apps are distributed to multiple markets; we call these *multi-released apps*. In some cases, an app is released to multiple marketplaces by its developer, seeking for more customers. Or, an outsider can also release a repackaged app to other marketplaces to benefit from pirated version of the app or even to disseminate malware, which looks like a popular benign app. In this work, we are interested in the latter case. Since it is not straightforward to distinguish between the two cases shown above, we do not attempt to directly distinguish them. However, we will study the characteristics of multi-released apps from the viewpoints of security measures; i.e., inclusions of malware and the use of license verification library for protecting apps from the illegal piracy.

To detect multi-released apps, we can use two approaches, the package name match and the MD5 hash value match. While the package name match can incorporate the variation of versions, it could suffer from the app renaming because there is no centralized system that governs the naming of apps among the third-party marketplaces. MD5 hash value match strictly guarantees that apps with same hash value are identical. However, they cannot incorporate the variation of versions. Since it is not feasible to collect all the versions for the millions of apps, we use these two approaches to compensate for each weak point.

Given these backgrounds in mind, we aim to understand how app multi-releasing occurs in the wild.

4.1 Statistics of Multi-released Markets

Figure 2 shows the cumulative percentages of numbers of multi-released markets for each app. From the package name result, we see that around 10 – 20% of apps are released to at least two markets.

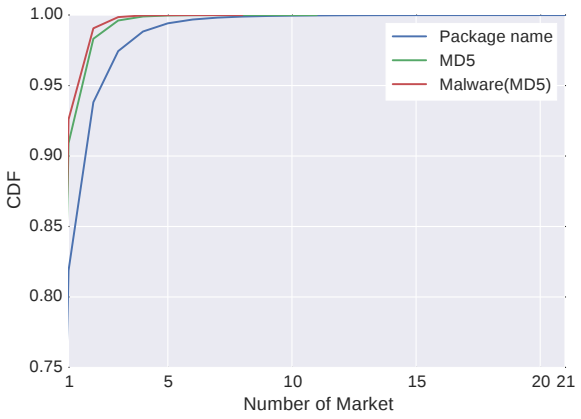

Figure 2: CDF of the number of multi-released markets for each app.

We can also see that there apps distributed to a large number of marketplaces; the top one app has been multi-released to 21 markets (com.estrongs.android.pop). com.google.zxing.client.android, com.shazam.android, and xcxin.filexpert have been released to 19 markets. All these apps are legitimate and popular apps in Google Play. We see that an MD5 match is a more strict condition than a package name match; therefore the number of multi-released marketplaces tend to be small. We also present the result for the multi-released malware, which we will describe later. While majority of malware samples are published at a single market, there are a few malware samples that are distributed to multiple markets.

4.2 Co-occurrence among Markets

Figure 3 is a heat map that shows how many apps published in a market are also published in other markets. Here, we use the package name match. The color of each cell represents a score computed with the following eq:

$$S(B|A) = \frac{|\,D(M_A) \cap D(M_B)\,|}{|\,D(M_A)\,|},$$

where $D(X)$ is a set of apps in a market X, M_A and M_B are the marketplaces A and B, respectively. A market in vertical axis and horizontal axis are represented as A and B, respectively. If the computed score is high, it means that most of the apps in market A are contained in market B.

There are many markets having common apps with Google Play. Unless they have special reasons not to, it is natural for developers to release their apps to the official market. On the other hand, the Iranian market *Cafebazaar* and Chinese markets, such as *Mi.com*, and *Anzhi*, do not have many apps in common with Google Play. As we mentioned in Section 5, we see some individuality in the Iranian marke. We speculate that the result for the Chinese markets reflects the fact that Google Play is unavailable in China. We also see that several Chinese markets, such as *Mi.com*, *Anzhi*, *Hiapk*, *Appchina*, and *Zhushou360*, have high numbers of shared apps among them, indicating that many apps are multi-released in those markets.

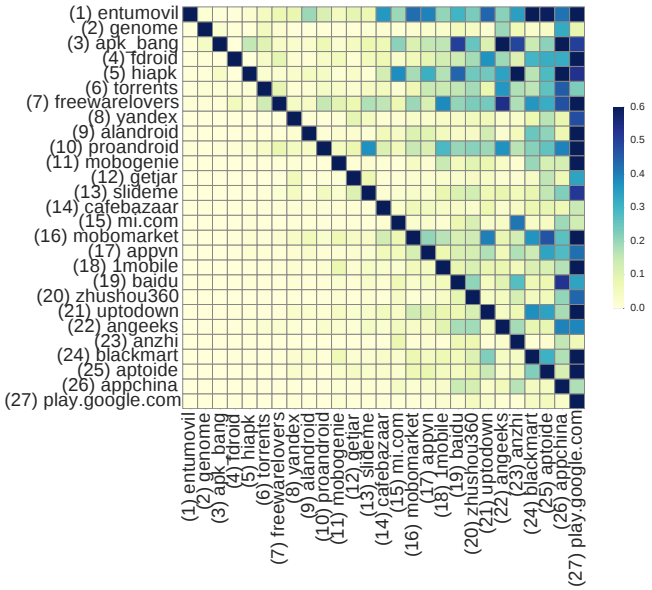

Figure 3: Heat map of multi-released apps. Color of each cell represents score $S(B|A)$, where A and B are the marketplaces shown in vertical axis and horizontal axis, respectively.

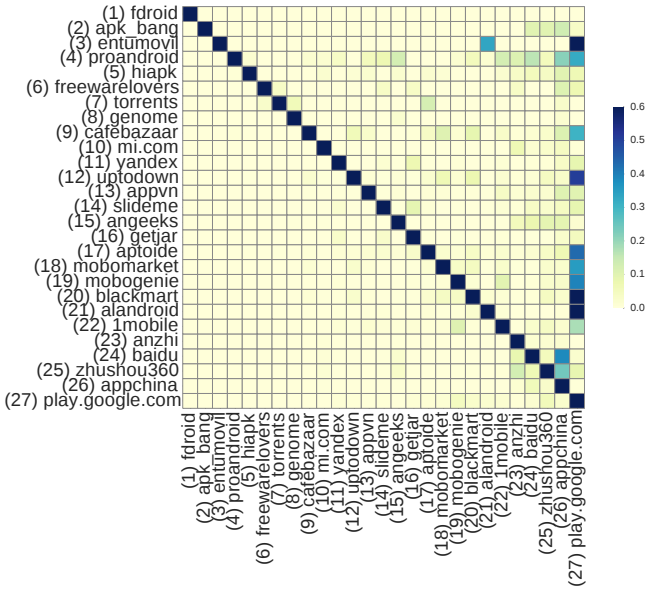

Figure 4: Heat map of multi-released malware. For visibility, the maximum value in this figure is set to 0.6.

4.3 Multi-released Malware

Figure 4 shows a heat map of multi-released malware generated in the same manner as Figure 3. This time, we used MD5 for matching because we aimed to ensure that we analyze only malware apps, but not apps that are benign and repackaged into malware. In this analysis, we make no distinction between adware and malware. We first see that there were released malware apps even in the

Table 3: Numbers of paid apps matched to the free apps dataset.

matching signature	# of match	match rate(%)
Package name	4,251	28.5
MD5	1,907	12.8

Table 4: Characteristics of multi-released paid apps. The numbers are average values. As of May 2017, 1 USD ≈ 115 JPY.

	Price (JPY)	download count
All paid apps	334.6	10,553.7
package name match	335.7	33,761.3
MD5 match	293.3	27,973.0

official market, at least at the time of crawling. In many cases, they are eliminated from the market within several weeks. There are several markets that contain malware in common with Google Play. We also see that several Chinese markets have intrinsic correlations, e.g., apk_bang, angeeks, and appchina. As these marketplaces have common malware samples with high probabilities, it is likely malware authors targeted these marketplaces at once.

4.4 Multi-released Paid Apps

In addition to the free apps we collected for this research, we had previously collected 14,906 paid apps from Google Play [17]. We checked the matching between this paid apps dataset and the free apps dataset. Table 3 shows the results. For package name match, we observe roughly 30% of paid apps we collected matched to those in free apps dataset. As we mentioned earlier this number include the cases where popular paid apps are repackaged/renamed in the app with the same package name and released to the third-party marketplaces. For MD5 matching, however, it is not possible to repackage/rename the apps without changing the hash values. Therefore, it is likely that some paid apps published in Google Play are illegally exported to other third-party marketplaces and published as free apps. Of course, there are also cases in which apps are available for free for a limited time or in which developers intentionally release to third-party markets for free.

Next, we look into the markets in which paid apps are multi-released for free at a high frequency; the top 4 markets matching this criteria were *Appvn*, *Appchina*, *Aptoide*, and *Anzhi*. Although we could not determine why these markets have so many paid multi-released apps, we conjecture that these marketplaces imported these paid apps so that it can attract customers. We note that in many cases, the downloaded paid apps from these multi-releasing marketplaces do not work due to the license verification restriction.

Characteristics of multi-released paid apps: Table 4 shows the statistics of the multi-released paid apps. There seems to be little relationship between multi-released apps and price. However, we see a trend that the download count of multi-released apps is about three times higher than that of not multi-released apps, suggesting that the top-ranked apps tend to be multi-released.

Table 5: Multi-releasing and LVL adoption.

	# of apps	# of LVL adoption (%)
not multi-released	10,655	1,803 (17%)
multi-released	4,251	1,739 (41%)

LVL adoption: We checked the adoption of License Verification Library (LVL) [1] for multi-released paid apps. LVL is the licensing system that validates whether a user has purchased an app legitimately. An app correctly implementing LVL queries the license server about the purchase status of the user; if the user bought the app legitimately, the app continues to work normally, otherwise, the app can stop working or perform other behaviors.

As apps with LVL need to declare the permission `CHECK_LICENSE`, we can automatically detect apps that aim to use it[1]. Although checking the permission necessarily means that the app actually has code of LVL and uses it, we assume that the derived statistics well represent the ground truth. Table 5 shows the results. Interestingly, the fraction of LVL adoption in the multi-released paid apps is higher than that of others, suggesting that more protected apps are multi-released. We also see that the 59% of multi-released apps do not adopt LVL. These apps can be launched without any limitations unless they implement their own license verification schemes.

An attacker can remove LVL code from an app using the repackaging techniques. Using the package name matching, we checked whether LVL had been removed for the multi-released apps with LVL. We found that 153 of 1,739 multi-released paid apps with LVL had their LVL permissions removed. Although this fraction is not significantly high, it signals a threat to app developers that an app's LVL can be removed and the app can be released to other marketplaces. We finally note that there are cases in which crackers remove/modify LVL code without removing the permission.

4.5 Certificate of Multi-released Apps

Finally, we studied the certificates of multi-released apps. The certificates were extracted using the `openssl` tools. If the certificate in one app is the same as in another, there is a high probability that the two apps were developed by the same developer. Although we cannot conclude if an app has been released to multiple markets by its original developer, we can say that apps sharing a common certificated have not been repackaged by outsiders. On the other hand, if an app with the same package name has been signed with keys in different certificates, there is a high probability that they have been repackaged.

We found that 478,653 out of 512,577 multi-released apps for which we could extract certificates using package name matching consistent certificates across each copy. However, the remaining 33,924 package names had at least two certificates that differed from each other. Although some developers may change a certificate across markets, we conjecture that such a case is not common. We note that the package name matching may have brought the

[1]We note that the `CHECK_LICENSE` permission is sometimes used for APK Expansion Files instead of LVL implementation.

Table 6: Factors determining market quality

factor	weight	explanation
Rate of benign apps	w_1	the rate of benign apps shown in Figure 1 (0.0 – 1.0)
Review system	w_2	1 if there is a user review system
Explanation of app permission	w_3	1 if there are explanations of the permissions the app requests
Report system	w_4	1 if there is a system to report an app as inappropriate by users
Safety badge	w_5	1 if there is a badge to show a app is safe by virus-checkers
HTTPS	w_6	1 if the market communicates with HTTPS

overlook of the repackaged multi-release apps with changed package names.

We found several legitimate apps that were multi-released to many markets (over 17), but were signed with a single key; namely, com.evernote, com.estrongs.android.taskmanager, wp.wattpad, and com.dropbox.android. These legitimate apps are popular in many of global marketplaces. We also found that multi-released apps signed with different certificates had higher malware detection rate, suggesting that such multi-release apps are repackaged into malware.

5 SECURITY INDEX OF MARKETPLACES

Several factors determine the security index of a market. For instance, even if a market has many malware apps, if efforts are taken by the market to prevent users from downloading them, we can regard such markets to be safe to some extent. To calculate the security index of a market, we examined the factors described in Table 6. The rate of benign apps takes a continuous value ($[0, 1]$), while the other factors are binary ($\{0, 1\}$). Note that each factor is evaluated not by app but by market.

We defined a security index calculated as the sum of each factor multiplied by its weight; i.e., $SI = \sum_i w_i f_i$, where w_i and f_i are the weight and value of the i-th factor, respectively. We explored possible parameter space and empirically derived the weight value as $w_1 = 5.0, w_2 = w_3 = w_4 = w_5 = w_6 = 0.5$, which was determined so that (1) it refelects the fact that the rate of benign apps should contribute to the security index a lot and (2) it clearly visualizes the differences among the marketplaces. An index takes a value within a range between 0.0 and 7.5. To visualize the relationship between a market's security index and its influence, we used the Alexa[2], which provides the ranking of websites. We plot the results in Figure 5. We note that *Genome* and *Torrent* are removed because they are not markets. We also removed *Apk_bang* because the market was unavailable the time of the study.

We notice several markets have low security index despite their high Alexa ranks, e.g., *Baidu*, *Yandex*, and *Mi.com*. As these marketplaces have high impact in terms of the size of users, they may need to improve their security index. As the low indices are mostly caused by high malware rates, they may want to introduce a mechanism to quickly eliminate apps detected as malware. Google Play's index is high, but it is not the best because it does not provide safety badge. *Cafebazaar* has an index close to that of Google Play. As we described in Section 2, the market is the most prominent market in

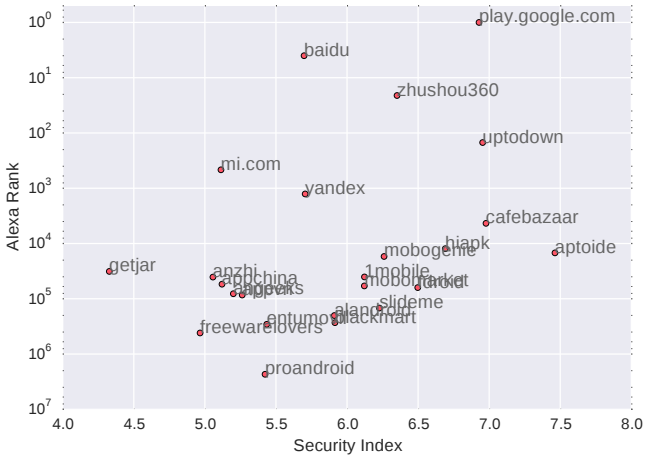

Figure 5: Relationship between security index and Alexa ranking

Iran and has an own billing system. Because the access to Google Play is restricted in Iran, many users rely on the market. Therefore, it is good that the market exhibits a good security index.

6 DISCUSSION

In this section, we discuss several limitations of our work, i.e., app collection, the methodology of analyzing apps, and collection of metadata.

6.1 App Collection

Although it is not feasible to cover all marketplaces, we did attempt cover a wide range of countries. We did not collect all apps in every market. Instead, we sampled top-ranked apps in each market. Another approach we could take is to apply other sampling schemes such as random sampling. We were also limited by not being able to track the updating or deletion of apps because we did not crawl the markets continuously, like the related work [13]. We leave these issues for our future study.

6.2 The Methodology of Analyzing Apps

First, to cope with a high volume of data, we did not conduct detailed code level analysis. For example, we would compare the codes of multi-released apps with the corresponding codes of originals. Second, we simply matched multi-released apps by package name. As most markets do not allow the submission of apps with existing package names, when an outsider attempts to multi-release an app to such markets, they must change the package name. For repackaged apps in particular, attackers often randomize or append characters to the original string. Therefore, our approach may miss such multi-released apps. To address this issue in the future, we will need to use methods such as APPraiser [12] to detect repackaged apps.

In Section 5, we discuss the measurement of the security index. However, measuring such index is not an easy task. For instance, the requirements for developers when they submit an app differs

among marketplaces. The strictness of reviewing apps by marketplaces are also different among marketplaces. As it is often difficult to quantify such parameters objectively, in this work, we chose some parameters that are intuitively easy to understand from the viewpoints of end-users.

6.3 Collection of Metadata

Androzoo, which we used to obtain Google Play data, does not have market metadata. If we had had access to metadata, we could have observed the transition of multi-released apps by submission date or compared developer information among markets. We leave this issue for our future work.

7 RELATED WORKS

App repackaging is a great concern in Android security. Repackaging means disassembling the APK – the Android executable file – inserting malicious codes or advertising modules, and then rebuilding the app [6]. Previously, several methods to detect repackaged apps have been proposed [14, 18–20].

In the following, we present studies that analyzed third-party markets themselves. In 2011, Vidas et al. [15] collected 41,057 apps from 194 markets and analyzed repackaged apps. They then proposed an app verification protocol called AppIntegrity to counter repackaging. AppIntegrity verifies an app to its developer's server using the domain name contained its package name. In 2013, Lindorfer et al. [13] developed the Andradar framework after a pre-survey involving crawling eight markets. Andradar scans and tracks malware in 16 markets in real time. It is of interest that, in developing the framework, they observed the processes of multi-releasing and deletion of apps. Unlike our work, Andradar focuses only on malware and does not use a large-scaled dataset crawled from each market. Viennot et al. [16] developed a system called PlayDrone, which efficiently crawls the official Google Play Store. Using roughly 1 million apps collected with PlayDrone, they performed various analyses of Android apps. In this study, in addition to Google Play, we also collected from third-party markets. We also discuss security measures in these markets.

8 SUMMARY

In this paper, we aimed to answer the simple research question: *are Android third-party marketplaces secure?* To answer this question, we collected more than 4.7 million Android apps from 27 of third-party marketplaces, including markets that had not previously been studied by the research community, and analyzed them in order to study their security measures. Based on our results, we attempted to quantify the security indices of marketplaces. Our key findings are summarized as follows: Apps published in some marketplaces, which have not been studied in the research communities, have not been exposed to the "dragnet" of online virus checkers. There are some intrinsic patterns of releasing/cloning apps among multiple marketplaces, and, finally, there are several marketplaces that, despite their high popularity, are insecure. From these findings, we propose the following suggestions to the stakeholders of Android ecosystem:

- **End-users**: Examine various indicators on the market to measure the safety of apps.
- **App developers**: Check apps that you have developed to see if they have been multi-released without your knowledge. Implement countermeasures against manipulation. For paid apps, implement LVL in addition to taking the steps above. Appropriate LVL implementation can prevent an app from launching even if it has been multi-released.
- **Marketplaces**: Improve the security indice we provided in this work so that end users can measure the safety of the market in a visible way. Sharing the information about the detected malicious apps or illegally pirated apps among marketplaces would be beneficial.

ACKNOWLEDGMENT

A part of this work was supported by JSPS Grant-in-Aid for Scientific Research B,Grant Number JP16H02832.

REFERENCES

[1] Adding licensing to your app | android developers. https://developer.android.com/google/play/licensing/adding-licensing.html.
[2] Alexa - actionable analytics for the web. http://www.alexa.com/.
[3] Android operating system statistics - appbrain. http://www.appbrain.com/stats/.
[4] apktool. https://code.google.com/p/android-apktool/.
[5] F-Secure: Android accounted for 97% of all mobile malware in 2013, but only 0.1% of those were on Google Play. http://thenextweb.com/google/2014/03/04/f-secure-android-accounted-97-mobile-malware-2013-0-1-google-play/.
[6] Fake apps: Feigning legitimacy. http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-fake-apps.pdf.
[7] jadx. https://github.com/skylot/jadx.
[8] smali. https://code.google.com/p/smali/.
[9] Supported locations for distribution to google play users. https://support.google.com/googleplay/android-developer/table/3541286.
[10] Virustotal. https://www.virustotal.com/.
[11] K. Allix, T. F. Bissyandé, J. Klein, and Y. Le Traon. Androzoo: collecting millions of android apps for the research community. In *Proceedings of the 13th International Workshop on Mining Software Repositories*, pages 468–471. ACM, 2016.
[12] Y. Ishii, T. Watanabe, M. Akiyama, and T. Mori. Clone or relative?: Understanding the origins of similar android apps. In *Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics*, pages 25–32. ACM, 2016.
[13] M. Lindorfer, S. Volanis, A. Sisto, M. Neugschwandtner, E. Athanasopoulos, F. Maggi, C. Platzer, S. Zanero, and S. Ioannidis. Andradar: fast discovery of android applications in alternative markets. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 51–71. Springer, 2014.
[14] Y. Shao, X. Luo, C. Qian, P. Zhu, and L. Zhang. Towards a scalable resource-driven approach for detecting repackaged android applications. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 56–65. ACM, 2014.
[15] T. Vidas and N. Christin. Sweetening android lemon markets: measuring and combating malware in application marketplaces. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 197–208. ACM, 2013.
[16] N. Viennot, E. Garcia, and J. Nieh. A measurement study of google play. Proc. of ACM SIGMETRICS 2014, June 2014.
[17] T. Watanabe, M. Akiyama, F. Kanei, E. Shioji, Y. Takata, B. Sun, Y. Ishii, T. Shibahara, T. Yagi, and T. Mori. Understanding the Origins of Mobile App Vulnerabilities: A Large-scale Measurement Study of Free and Paid Apps. In *Proceedings of IEEE/ACM 14th International Conference on Mining Software Repositories (MSR 2017)*, July 2017.
[18] Zhauniarovich, Yury, Gadyatskaya, Olga, Crispo, Bruno, L. Spina, Francesco, Moser, and Ermanno. Fsquadra: Fast detection of repackaged applications. Proc. of IFIP DBSec '14, pages 131–146, 2014.
[19] W. Zhou, Y. Zhou, M. Grace, X. Jiang, and S. Zou. Fast, scalable detection of "piggybacked" mobile applications. In *Proc. of the third ACM CODASPY 2013*, pages 185–196.
[20] W. Zhou, Y. Zhou, X. Jiang, and P. Ning. Detecting repackaged smartphone applications in third-party android marketplaces. In *Proc. of the second ACM CODASPY 2012*, pages 317–326.
[21] Y. Zhou and X. Jiang. Dissecting android malware: Characterization and evolution. In *2012 IEEE Symposium on Security and Privacy*, pages 95–109. IEEE, 2012.