ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 3, Issue 11, November 2016

Survey on Detecting Malicious Facebook Applications

[1] T.K.Pradeep Kumar [2] M. Sheela Devi [3] Bitan Paul [4] Neesu Dubey [5] Minish Dixit [6] Neha Jha
Department Of CSE
Sri Sairam College of Engineering, Bangalore

Abstract: **-- The popularity and addictiveness of Facebook is due to existence of the third-party apps as there are installations of nearly 20 million per day.Due to which, malware and spam are easy to spread since there is a potential use of these apps which has been identified by hackers. The problem is already weighty as we find that at least 13% of applications in our dataset are malevolent.There has been focus by the research community to detect malicious posts as well as campaigns.Our major contribution lies in developing FRAppE—Facebook's Rigorous Application Evaluator which is arguably the first tool focused on detecting malicious apps on Facebook. To develop FRAppE, we make use of information gathered by closely observing the posting behaviour of 111K Facebook apps seen across 2.2 million users on Facebook. First, we identify a set of features that helps us to differentiate between malign apps and kind apps. For example, we find that malicious apps often share names with other apps, and they commonly request fewer permissions than kind apps Second, leveraging these distinguishing features, we show that FRAppE can detect malicious apps with 99.5% accuracy, with no false positives and a high true positive rate (95.9%).Finally, we examine the ecosystem of malicious Facebook apps and recognize methods that these apps use to multiply. Interestingly ,we find that many apps conspire and support each other; in our dataset, we find 1584 apps enabling the viral multiplication of 3723 other apps through their posts. In long term measures, we identify FRAppE as a step towards creating an independent watchdog for app ranking & assessment, so as to make Facebook users aware before installing apps.**

I. INTRODUCTION

Background

We discuss how applications work on Facebook, and we outline the datasets that we use in this paper.

A. Facebook Apps

Facebook provides services toits users by means of Facebook applications. Not similar to conventional desktop and smartphone applications, installation of a Facebook application by a user does not involve the user downloading and executing an application binary. Instead, when a user adds a Facebook application to his/her profile, the user grants the application

Server performs following things:

1) Permissions to access a piece of information listed on the user's Facebook profile (e.g., e-mail address).

2) Permissions to execute particular actions on behalf of the user (e.g., ability to post on the user's wall).

These permissions to any application are granted by the Facebook for each user who installs the application by handing an O Auth 2.0 token to the application server. After that, the data can be accessed by the application and perform the actions on behalf

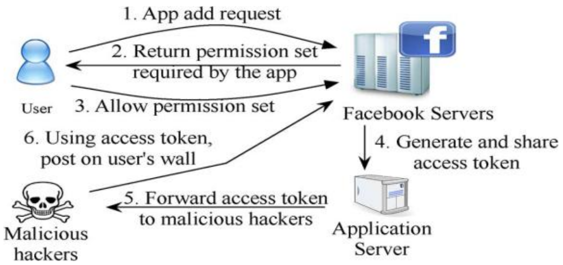

1. App add request
2. Return permission set required by the app
User
3. Allow permission set
Facebook Servers
4. Generate and share access token
6. Using access token, post on user's wall
5. Forward access token to malicious hackers
Malicious hackers
Application Server

Fig. 1. Steps involved in hackers using malicious applications to get access tokens to post malicious content on victims' walls of the user which are permitted to be done in an explicit manner. Fig. 1 depicts the steps involved in the installation and operation of a Facebook application.

Operation of Malicious Applications:

Facebook applications which are malicious commonly tend to operate as follows:

- Step 1: Users are persuaded, usually with some
- fake promises ,to install the app by hacker.
- Step 2: Once the app has been installed by the user, it redirects the user to a Web page where is request is made to the user to perform task such as completing a survey, again with the lure of fake rewards.
- Step 3: The personal information from the user's profile is then easily accessed by the application which in turn leads the hackers to make profit out of it in a potential manner.
- Step 4: The app makes malicious posts on behalf of the user to lure the user & convince his/her friends to install the same app.

The rotation continues with the app or conspiring app searching more and more users. Surveys or personal information can be sold to third parties to eventually profit the hackers.

B. Our Datasets

The base of our research is a dataset which is obtained from 2.2M active Facebook users, who are constantly monitored by My Page Keeper, our security application for Facebook. My Page Keeper evaluates every URL that it encounters on any user's wall or news feed to determine if that URL leads to social spam. My Page Keeper classifies a URL as social spam if it leads to a Web page that:

1) Spreads malware;
2) Attempts to "phish" for personal information;
3) Requests the user to carry out tasks (e.g., fill out surveys)that profit the owner of the Web site;
4) Promises false rewards;
5) Attempts to entice the user to artificially inflate the reputation of the page;

My Page Keeper evaluates each URL using a machine-learning-based classifier that leverages the social context attached with the URL. For any particular URL, the features used by the classifier are obtained by uniting information from all posts (seen across users) containing that URL.

Example features used by My Page Keeper's classifier include the likeness of text message across posts and the numbers of comments or Likes on those particular posts. My Page Keeper has false positive and false negative rates of 0.005% and 3%. Our dataset includes 91 million posts from 2.2 million walls constantly monitored by My Page Keeper over 9 months from June 2011 to Note that Facebook has deprecated the app directory in 2011, therefore there is no central directory available for the entire list of Facebook apps [19].

Table I
Summary of the dataset collected by mypagekeeper From june 2011 to march 2012

Dataset Name	# of apps	
	Benign	Malicious
D-Total	111,167	
D-Sample	6,273	6,273
D-Summary	6,067	2,528
D-Inst	2,257	491
D-ProfileFeed	6,063	3,227
D-Complete	2,255	487

Table II
Top malicious apps in d-sampe dataset

App ID	App name	Post count
235597333185870	What Does Your Name Mean?	1006
159474410806928	Free Phone Calls	793
233344430035859	The App	564
296128667112382	WhosStalking?	434
142293182524011	FarmVile	210

The D-Sample Dataset:
Finding Malicious Applications:

To recognize malicious Facebook applications in our dataset, we start with a simple strategy: If any post made by an application was flagged as malicious by MyPageKeeper, we mark the applicationas malicious. By applying this strategy, we identified 6350 malicious apps. Interestingly, we discover that many popular applications such as Facebook for Android were also marked as malicious in this process. This is in fact the outcome of hackers exploiting Facebook weaknesses. To avoid such misclassifications, we verify appsusing a white-list that is created by taking into consideration the most popular apps and important manual effort. After white-listing, we are left with 6273 malicious applications (D Sample datasetin Table I). Table II shows the top five malicious applications, in terms of number of posts per application.

Although we conclude the ground truth data about malicious applications from MyPageKeeper, it is possible that MyPageKeeper itself has potential inclination classifying malicious app's posts. For example, if a malevolent application is not so much popular and therefore does not appearin many users' walls or news feeds, MyPageKeeper mayfail to classify it as malicious (since it works on post level). However, as we show here later, our proposed system uses a different set of features than MyPageKeeper and can identify even very unpopular apps with high accuracy and low false positives and false negatives.

Fig. 2 shows the number of new malicious apps seen in every month of the D-Sample dataset. For every malicious app in the D-Sample dataset, we consider the time at which we observed the first post made by this app as the time at which the app was launched. We identify that hackers launch new malignant apps every month in Facebook, although September 2011, January 2012, and February 2012 see significantly higher new malicious app activity than other months. Out of the 798 malicious apps launched in September 2011, we find 355 apps all created with the name "The App" and 116 apps created with the name "Profile Viewing." Similarly, of the 3813 malicious apps created in February 2012, 985 and 589 apps have the name "Are You Ready" and "Pr0file Watcher," respectively. Other examples of

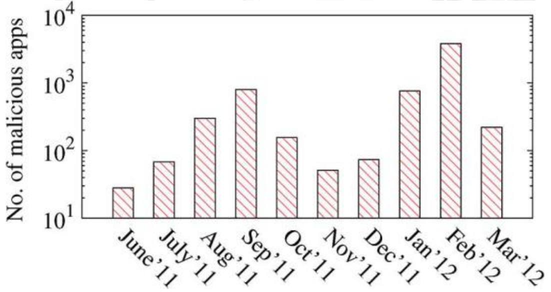


Fig. 2. Malicious apps launched per month in D-Sample dataset app names used often are:

"What does your name mean?," "FortuneTeller," "What is the sexiest thing about you?," and so on. D-Sample Dataset: Including Benign Applications:

To select an equal number of kind apps from the initial D-Total dataset,we use two criteria:

1) None of their posts were identified as malicious by MyPageKeeper
2) They are "vetted" by Social Bakers, which monitors the "social marketing success" ofapps.

This mechanism yields 5750 applications, 90% of which havea user rating of at least 3 out of 5 on Social Bakers. To match the number of malicious apps, we append the top 523 applications inD-Total (in terms of number of posts) and obtain a set of 6273 kind applications. The D-Sample dataset (Table I) is the unionof these 6273 benign applications with the 6273malicious applications obtained earlier. The most popular benign apps are FarmVille,Facebook for iPhone, Mobile, Facebook for Android,and Zoo World. For profiling apps, we collect the information for apps that is readily available through Facebook. We use a crawler basedon the Firefox browser instrumented with Selenium. From March to May 2012, we creep information for every application in our D-Sample dataset once every week. We collected app summaries and their permissions, which requires two different creeps.

D-Summary Dataset: Apps With App Summary:

We gather app summaries through the Facebook Open graph API, which is made available by Facebook at a URL of the form https://graph.facebook.com/App_ID. Facebook has a unique identifier foreach application. A summary of the app possesses several subsets of information such as application name, description, company name, profile link, and monthly active users. If any application has been removed from Facebook, the query results in an error. We were able to collect the summary for 6067 benign and 2528 malicious apps (D-Summary dataset in Table I). It is easy to understand why malicious apps were more often removed fromFacebook.

D-Inst Dataset: App Permissions:

We also wish tostudy the permissions that apps request at the installation time. For every application App_ID, we crawl

https://www.facebook.com/apps/application.php?id=App

_ID,which usually redirects to the application installation's URL.We were able to get the permission set for 487 maliciousand 2255 benign applications in our dataset. Automaticallycreeping the permissions for all apps is not trivial, asdifferent apps follow different redirection mechanisms, which areplanned for humans and not for creepers. As expected, here too thequeries for apps that are discarded from Facebook fail.

D-ProfileFeed Dataset: Posts on App Profiles:

Users canmake posts on the profile page of an app, which we can call the

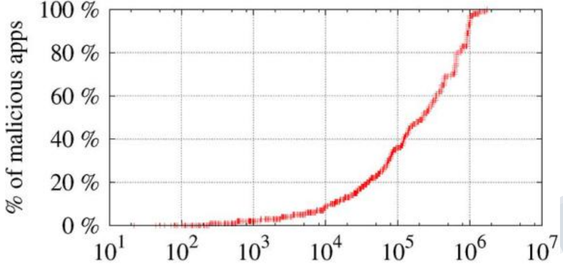


Fig. 3. Clicks received by bit.ly links posted by malicious apps.

profile feed of the app. We gather these posts using the Opengraph API from Facebook. The API returns posts appearing on the application's page, with several properties for eachpost, such as *message*, *link*, and *create time*. Of the apps inthe D-Sample dataset, we were able to get the posts for 6063benign and 3227 malicious apps. We construct the D-Complete dataset by taking the intersection of D-Summary, D-Inst, andD-ProfileFeed datasets.

II. PREVALENCE OF MALICIOUS APPS

The driving motivation for detecting malicious apps stems from the suspicion that a significant fraction of malicious post on Facebook are posted by apps.We find that 53% of malicious posts flagged byMyPageKeeper were posted by malicious apps.
We further quantify the prevalence of malicious apps in two different ways. 60% of malicious apps get at least a lakh clickson the URLs they post.

We quantify the reach of malicious appsby determining a lower bound on the number of clicks on thelinks included in malicious posts. For each malicious app in ourD-Sample dataset, we identify all *bit.ly* URLs in posts madeby that application.We focus on *bit.ly* URLs since bit.lyoffers an API for querying the number of clicks received byevery *bit.ly* link; thus, our estimate of the number of clicksreceived by every application is strictly a lower bound.

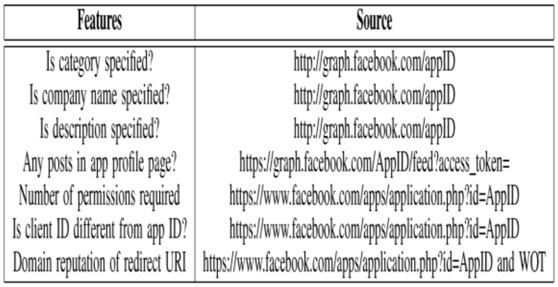
Across the posts made by the 6273 malicious apps in theD-Sample dataset, we found that 3805 of these apps had posted5700 *bit.ly* URLs in total.We queried *bit.ly* for the clickcount of each URL. Fig. 4 shows the distribution across maliciousapps of the total number of clicks received by *bit.ly*links that they had posted. We see that 60% of malicious appswere able to accumulate over 100K clicks each, with 20%receiving more than 1M clicks each. The application with thehighest number of *bit.ly* clicks in this experiment—the "What is the sexiest thing about you?" app—received 1 742 359 clicks. Although it would be interesting to find the *bit.ly*click-through rate per user and per post, we do not have datafor the number of users who saw these links. We can query*bit.ly*'sAPI only for the number of clicks received by a link. 40% of malicious apps have a median of at least 1000monthly active users.

Table III
Top five domains hosting malicious apps in d-inst dataset

Domains	Hosting # of malicious apps
thenamemeans3.com	34
fastfreeupdates.com	53
wikiworldmedia.com	82
technicalyard.com	96
thenamemeans2.com	138

We examine the reach of malicious appsby atleast 1000 users, while 60% of malicious applications achievedat least 1000 during the 3-month observation period. The topmalicious app here—"Future Teller"—had a maximum MAU of 260 000 and median of 20 000.

III. DETECTING MALICIOUS APPS



Features	Source
Is category specified?	http://graph.facebook.com/appID
Is company name specified?	http://graph.facebook.com/appID
Is description specified?	http://graph.facebook.com/appID
Any posts in app profile page?	https://graph.facebook.com/AppID/feed?access_token=
Number of permissions required	https://www.facebook.com/apps/application.php?id=AppID
Is client ID different from app ID?	https://www.facebook.com/apps/application.php?id=AppID
Domain reputation of redirect URI	https://www.facebook.com/apps/application.php?id=AppID and WOT

Now as we know the distinguishing characteristics inspecting the number of users that these applications had. To study this, we use the Monthly Active Users (MAU) metric provided by Facebook for every application. The number of Monthly Active Users is a measure of how many unique usersare engaged with the application over the last 30 days in activitiessuch as installing, posting, and liking the app. Fig. 4 plotsthe distribution of Monthly Active Users of the maliciousapps in our D-Summary dataset.For each app, the median and of both malicious and benign apps, we next use these facilities to develop proper classification techniques to recognize malevolent Facebook applications.We present two variants of our malicious app classifier— FRAppELite and FRAppE.

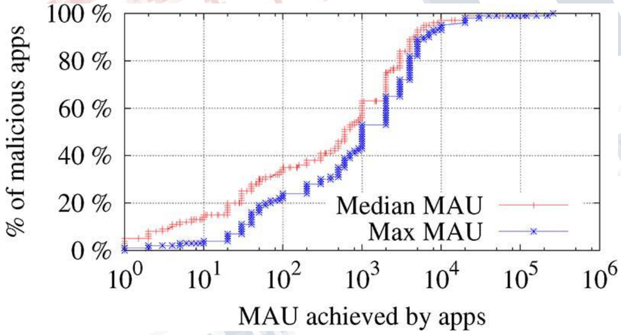


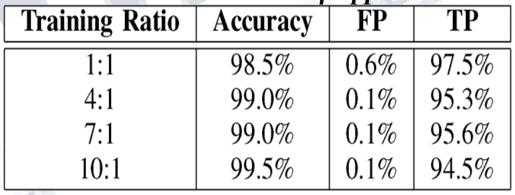
Fig. 4.Median and maximum MAU achieved by malicious apps maximum MAU values over the three months are shown. Wesee that 40% of malicious applications had a median MAU of

A. FRAppELite

FRAppELite is a lightweight version that makes use of only the application features available on demand. Given a specific app ID, FRAppELite creeps the on-demand features for that application and evaluates the application based on these featuresin real time. We visualize that FRAppELite can be incorporated, for example, into a browser extension that can evaluate any Facebook application at the time when a user is consideringinstalling it to his/her profile.

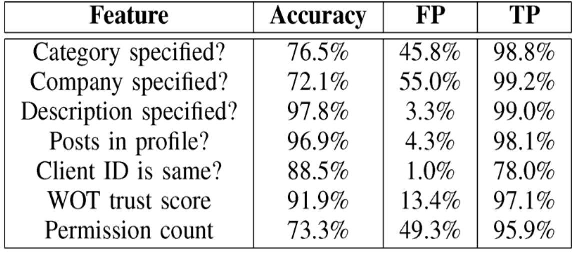
Table IV lists the features used as input to FRAppELite andthe source of each feature. All of these features are collectible on demand at the time of differentiation and do not require preknowledge about the app which is going to be evaluated.We use the Support Vector Machine (SVM) classifier for classifying malicious apps. SVM is widely used for binary classification in security and other disciplines. We usethe D-Complete dataset for training and testing the classifier. As shown earlier in Table I, the D-Complete dataset consists of 487malicious apps and 2255 benign apps. We use 5-fold cross validation on the D-Complete dataset fortraining and testing FRAppELite's classifier. In 5-fold cross validation, the dataset is randomly divided into five segments,and we test on each

Table V
Cross validation with frappe lite



Training Ratio	Accuracy	FP	TP
1:1	98.5%	0.6%	97.5%
4:1	99.0%	0.1%	95.3%
7:1	99.0%	0.1%	95.6%
10:1	99.5%	0.1%	94.5%

Table VI
Classification Accuracy With Individual Features



Feature	Accuracy	FP	TP
Category specified?	76.5%	45.8%	98.8%
Company specified?	72.1%	55.0%	99.2%
Description specified?	97.8%	3.3%	99.0%
Posts in profile?	96.9%	4.3%	98.1%
Client ID is same?	88.5%	1.0%	78.0%
WOT trust score	91.9%	13.4%	97.1%
Permission count	73.3%	49.3%	95.9%

Performance of the classifier. Accuracy is defined as the ratio of correctly identified apps (i.e., a benign/malicious app is appropriatelyidentified as benign/malicious) to the total number of apps. False positive rate is the fraction of benign apps incorrectly classifiedas malicious, and true positive rate is the fraction of benignand malicious apps correctly classified (i.e., as benign and malicious, respectively).

We conduct four separate experiments with the ratio of benignto malicious apps varied as 1:1, 4:1, 7:1, and 10:1. In each case,we sample apps at random from the D-Complete dataset and runa 5-fold cross validation. Table V shows that, irrespective of theratio of benign to malicious apps, the accuracy is above 98.5%.The higher the ratio of benign to malicious apps, the classifier gets trained to minimize false positives, rather than false negatives,in order to maximize accuracy. However, we note that the false positive rate is below 0.6% and true positive rate isabove 94.5% in all cases. The ratio of benign to malicious apps in our dataset is equal to 7:1; of the 111K apps seen in data of MyPageKeeper, 6273 apps were identified as malicious based on MyPageKeeper classification of posts, and an additional 8051apps are found to be malicious.

Henceforth, we can expect FRAppELite to offer roughly 99.0% accuracy with 0.1% false positives and 95.6% true positives in practice. To understand the contribution of each of FRAppELite's features toward its accuracy, we next perform 5-fold cross validationon the D-Complete dataset with only a single feature at atime.

Table VI shows that each of the features by themselves too result in reasonably high accuracy. The "Description" featureyields the highest accuracy (97.8%) with low false positives(3.3%) and a high true positive rate (99.0%). On the flip side,classification based solely on any one of the "Category," "Company,"or "Permission count" features results in a large number of false positives, whereas relying solely on client IDs yields a low true positive rate.

B. FRAppE

Next, we consider FRAppE—a malicious app detector that utilizes our aggregation-based features in addition to the on-demand features. Table VII shows the two features that FRAppEuses in addition to those used in FRAppELite. Since the aggregation-based features for an app require a cross-user and cross-app view over time, in contrast to FRAppELite, we visualize that FRAppE can be used by Facebook or by third-partysecurity applications that protect a large population of users. Here, we again conduct a 5-fold cross validation with the D-Complete dataset for various ratios of benign apps to maliciousapps. In this case, we find that, with a ratio of 7:1 in benign to malicious apps, FRAppE's additional features improvize the accuracy to 99.5% (true positive rate 95.1% and true negative rate100%), as compared to 99.0% with FRAppELite. Furthermore,the true positive rate gets increased from 95.6% to 95.9%, and we don't have a single false positive.

Table VII
Additional Features Used In Frappe

Feature	Description
App name similarity	Is app's name identical to a known malicious app?
External link to post ratio	Fraction of app's posts that contain links to domains outside Facebook

Table VIII
Validation Of Apps Flagged By Frappe

Criteria	# of apps validated	Cumulative
Deleted from FB graph	6,591(81%)	6,591 (81%)
App name similarity	6,055(74%)	7,869 (97%)
Post similarity	1,664 (20%)	7,907(97%)
Typosquatting of apps	5(0.1%)	7,912(97%)
Manual validation	147 (1.8%)	8051 (98.5%)
Total validated	-	8051(98.5%)
Unknown	-	93 (1.5%)

C. Identifying New Malicious Apps

We next train FRAppE's classifier on the entire D-Sample dataset and use this classifier to identify new malicious apps. To do so,we apply FRAppE to all the apps in ourD-Total dataset that aren't in the D-Sample dataset; for these apps, welack information as to whether they are malicious or benign. Of the 98 609 apps that we test in this experiment, 8144 apps were flagged as malicious by FRAppE.

Validation:

Since we lack ground truth information for theseapps flagged as malicious, we apply a host of complementary mechanisms to validate FRAppE's classification. Next we describe these validation techniques; as shown in Table VIII, wewere able to validate 98.5% of the apps flagged by FRAppE.

Deleted From Facebook Graph:

Facebook itself monitors itsplatform for malevolent activities, and it disables and deletes from the Facebook graph malicious apps that it identifies. If the

Facebook API (https://graph.facebook.com/appID) returnsfalse for a particular app ID, this indicates that the app no longer exists on Facebook; we consider this to be indicative of black listingby Facebook. This technique validates 81% of the maliciousapps identified by FRAppE. Note that Facebook's measuresfor detecting malicious apps are however insufficient; ofthe 1464 malicious apps identified by FRAppE but are still active on Facebook,35% have been active on Facebook since over 4 monthswith 10% dating back to over 8 months.

App Name Similarity:

If an application's name exactlymatches that of multiple malicious apps in the D-Sampledataset, that app is likely to be part of the same campaign and therefore malicious as well. On the other hand, we found several malicious apps using version numbers in their name. Thus, in addition, if an app name contains aversion number at the end and the rest of its name is identicalto multiple known malicious apps that similarly use version numbers, this too is indicative of the app likely being malicious.

Posted Link Similarity:

If a URL posted by an app matches the URL posted by a previously known malicious app, thenthese apps are likely part of the same spam campaign, thus validating the former as malicious.

***URL hijacking(Typo-Squatting) ofPopular App*:**

If an app's name is "Typo-Squatting"that of a popular app, we consider it malicious. For example, we found five apps named "FarmVile," which areseeking to leverage the popularity of "FarmVille." Note thatwe used "typosquatting" criteria only to validate apps that werealready classified as malevolent by FRAppE. We did not use thisfeature as standalone criteria for classifying malicious apps ingeneral. Moreover, it could only validate 0.5% of apps in ourexperiment as shown in Table VIII.

Manual Verification:

For the remaining 232 apps unverified by the above techniques, we cluster them based on name similarity among themselves and verify one app from each clusterwith cluster size greater than 4.
For example, we find 83 appsnamed "Past Life." This enabled us to validate an additional147 apps marked as malicious by FRAppE. D. Representativeness of Ground Truth for Benign Apps

We demonstrate the representativeness of benign apps usedin our ground truth data set in the following ways. First, we selected 6000 apps randomly from 91 000 apps in our dataset andcompared the median MAU to that of 6000 benign apps in our ground truth dataset. As shown in Fig. 14, benign apps have median MAUs distributed across a wide range similar to theMAUs

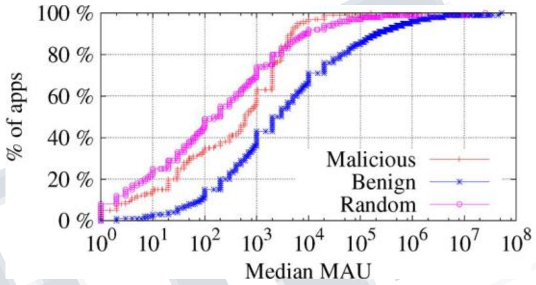


Fig. 5.MAU comparison among malicious, benign, and randomly selectedapps.

Of randomly selected apps. Second, we tested FRAppE on two different sets of benign apps (1125 apps in each set), where oneset had significantly more popular apps (median MAU 20 000)than the other (medianMAU 500).We repeated 5-fold cross validation on each set independently and found that the false positive rate showed only a marginal increase from 0% in the case of popular apps to 0.18% for unpopular apps. Thus, FRAppE's accuracy is not biased by the popularity of apps in our dataset of benign apps.

REFERENCES

[1] SazzadurRehman, Ting-Kai Huang, Harsha V. Madhyastha, and MichalisFaloutsos on "Detecting Malicious Facebook Applications", 2015.

[2] "MyPageKeeper," [Online]. Available: https://www.facebook.com/ apps/application.php?id=167087893342260

[3] "Wiki: Facebook platform," 2014 [Online]. Available: http://en.wikipedia.o

[4] Harsha V. Madhyastha University of California, Riverside, CA, USA