Muñoz, A., Martín, I., Guzmán, A. & Hernández, J. A. (2015). Android malware detection from Google Play meta-data: Selection of important features. *2015 IEEE Conference on Communications and Network Security (CNS)*.

DOI: 10.1109/CNS.2015.7346893

Android malware detection from Google Play meta-data: Selection of important features

Alfonso Muñoz, Ignacio Martín, Antonio Guzmán, José Alberto Hernández

Android malware has emerged in the last decade as a consequence of the increasing popularity of smartphones and tablets. While most previous work focuses on inherent characteristics of Android apps to detect malware, this study analyses indirect features to identify patterns often observed in malware applications. We show that modern Machine Learning techniques applied to collected metadata from Google Play can provide a first approach towards the detection of malware applications, and we further identify which features have the highest predictive power among the total.

Smartphone security; Android malware; Google Play; Machine Learning.

I. Introduction and Motivation

According to latest estimates, the number of smartphone users has reached 1.75 billion at the beginning of 2014, and is expected to grow up to 2.50 billion to 2017. Android has positioned itself as the leading operating system in the smartphone industry, accounting for more than 75% of devices by the end of 2014. Unfortunately, such popularity has made Android become one of the most valuable targets for malware developers. It is estimated that around 3-4% of total applications available in Google Play are malware (around 60k apks from a total of 1.5M+).

Previous work on Android malware detection has traditionally focused on exploring intrinsic features of the Android application itself, mainly permissions, API calls, CPU usage, system calls and/or process or memory information (see [1], [2], [3], [4]).

However, there is a lot of information available at Google Play that can be used to identify patterns in malware. Such information along with modern machine learning tools can provide a first step towards malware detection, as a complement to traditional application sandboxing. In the next sections, we show that certain features available in Google Play, especially those related with the developer and certificate issuer, are extremely powerful in discriminating malware from goodware. Other features, such as those concerning the sentiment analysis of the comments written by the users, which have been also considered in the past (see [5]) are not considered in this study.

II. Dataset and Methodology

The dataset under study comprises around 25k applications randomly obtained from Google Play Store, collected between May 2014 and March 2015. Such dataset has been collected used using the Tacyt cyber-intelligence tool developed internally at Eleven Paths, Telefónica[1]. For each application, we have collected around 48 features including:

1) Intrinsic application features: including its title, size (in bytes), code version, number of Android permissions used, number of images and files used, and the date of creation, upload and update at Google Play.

2) Application Category: including game, education, entertainment, lifestyle, business, etc.

3) Developer-related features: including the user contact name, email and webpage. There exist around 7569 different developer names in this dataset. This information allows to create white and blacklists regarding developers' reputation, and see if those developers associated with malware applications are recurrent in developing malware.

4) Certificate-related features: including the relevant dates and issuer information. In particular, each application certificate contains the expedition and expiration date, issuer and subject name and the country where the certificate is expedited. Again, white and black lists may be created regarding certificate issuers.

5) Social-related features: involving relevant feedback collected from users and available at the market. As Google Play is strongly connected with the social network G+, features like number of total votes or average star rating are provided in this set of features.

Once downloaded, all applications have been checked for malware detection using the VirusTotal web service[2]. VirusTotal tests each application against a number of malware engines (McAfee, AVG, VIPRE, TrendMicro, etc.), thus producing a binary result (malware/goodware). In our 25k apk dataset, 386 apks (1.57%) have been tagged as malware by any AV engine.

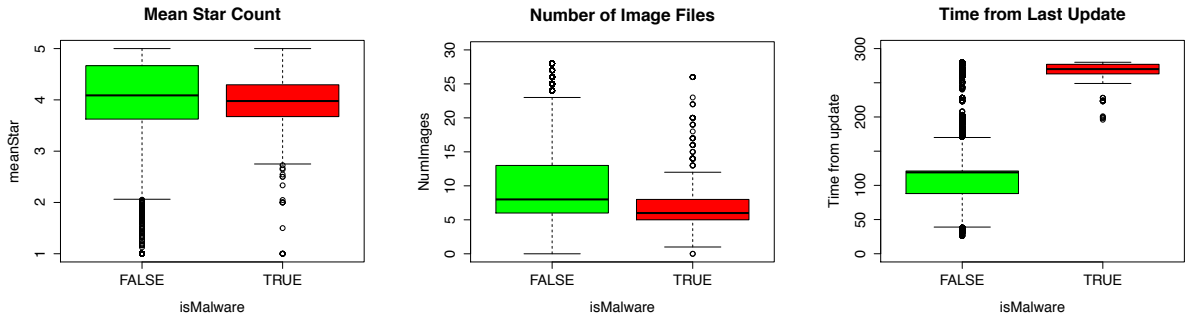
Fig. 1 shows three boxplots regarding three particular features: the average number of stars obtained by an application (left), the number of image files per application (center) and the time from last version update (right). As shown, the first feature has little predictive power (same behaviour for malware and goodware), whereas the last one clearly reveals a pattern to identify malware: most malware applications are not regularly updated in Google Play (more than 250 days median). In conclusion, some features seem very promising while others are expected to have little or no predictive power.

In the next section, we analyse the predictive power of the 48 features collected from Google Play using the well-

A. Muñoz and A. Guzmán are with ElevenPaths, Grupo Telefónica, Spain (email: {Alfonso.Munoz,Antonio.Guzman}@11paths.com)
I. Martín and J. A. Hernández are with Univ. Carlos III de Madrid, Spain (email: {ignmarti,jahgutie}@uc3m.es)
The authors would like to acknowledge the support of the project BigDatAAM (grant no. FIS2013-47532-C3-3-P) funded by the Spanish MINECO.

[1] See https://www.elevenpaths.com/technology/tacyt/index.html

[2] Virustotal- Free Online Virus, Malware and URL Scanner, available at: https://www.virustotal.com/



(a) MeanStar (b) NumImages (c) LastUpdate

Fig. 1. Goodware/Malware boxplot comparison for three features: Average stars, Number of image files and Number of days since last update

known Logistic Regression model along with the Step Akaike Information Criterion for feature selection.

III. Experiments and conclusions

Table I shows the values of Accuracy, Precision, Recall and F1-score metrics for both training and test datasets, for each group separately and altogether, both before and after Step-AIC is applied. Small values of the decision threshold η produce a situation where many applications are detected as malware to avoid FN, at the expense of producing a high value of FP. Second, it may be observed that social features yield really bad scores, thus showing little predictive power. On the contrary, certificate and developer information as well as intrinsic application information are the most promising candidate groups for malware detection, along with the application category.

Feat. group	p	Accuracy	Precision	Recall	F1-score
Before Step-AIC (train/test)					
Intrinsic	8	0.89/0.89	0.95/0.96	0.12/0.12	0.22/0.21
Developer	3	0.98/0.97	0.86/0.36	0.52/0.31	0.65/0.34
Certificate	3	0.98/0.97	0.84/0.42	0.44/0.27	0.58/0.33
Social	7	0.98/0.98	0.01/0.006	0.08/0.05	0.01/-
Categories	16	0.85/0.84	0.37/0.31	0.04/0.03	0.07/0.05
Total	48	0.98/0.97	0.96/0.53	0.45/0.31	0.61/0.39
After Step-AIC (train/test)					
Intrinsic	4	0.89/0.89	0.95/0.95	0.12/0.12	0.22/0.22
Developer	2	0.98/0.97	0.86/0.36	0.52/0.31	0.65/0.33
Certificate	3	0.98/0.97	0.84/0.42	0.44/0.27	0.58/0.33
Social	5	0.98/0.98	0.009/0.003	0.08/0.03	0.01/-
Categories	11	0.82/0.82	0.42/0.42	0.03/0.03	0.07/0.06
Total AIC	8	0.97/0.96	0.95/0.56	0.42/0.30	0.59/0.39

TABLE I

Ten-Fold validation results (Logistic threshold: $\eta = 0.035$).

The use of Step-AIC is observed to reduce the number of features to the eight most relevant ones, yielding simple models without compromising performance. Such 8 most relevant features are shown in Table II, sorted by p-value. Remark that the smaller the p-value, the more critical this parameter is in the model.

As shown, features related with the developer's (developerRep) and certificate-issuer's reputation (issuerRep) are the most important ones in discriminating malware from goodware, along with the number of days elapsed since the application was created and first uploaded (createDate and uploadDate) at Google Play. In addition, the number of permissions required by the application (numPerm) is also a good indicator of malware (this is consistent with the literature) and the number of one-star votes too (oneStarRatingCont). Finally, LIFESTYLE application category (cat.LIFESTYLE) is also a good indicator since we have empirically observed that the percentage of malware under the category LIFESTYLE is substantially larger than it is for goodware applications.

Features	Estimate	z-score	p-value
developerRep	7.340675	9.265851	$1.935270\text{x}10^{-20}$
createDate	$-4.487129\text{x}10^{-02}$	-7.937822	$2.057620\text{x}10^{-15}$
(Intercept)	$7.172184\text{x}10^{02}$	7.878447	$3.314755\text{x}10^{-15}$
numPerm	$1.141816\text{x}10^{-01}$	4.289095	$1.794028\text{x}10^{-05}$
issuerRep	3.084339	3.689452	$2.247377\text{x}10^{-04}$
cat.LIFESTYLE	$-8.352139\text{x}10^{-01}$	-2.684080	$7.272962\text{x}10^{-03}$
uploadDate	$5.408091\text{x}10^{-04}$	1.673462	$9.423637\text{x}10^{-02}$
oneStarRatingCont	$3.718558\text{x}10^{-05}$	1.659646	$9.698561\text{x}10^{-02}$
cat.OTHER	$-4.552049\text{x}10^{-01}$	-1.599757	$1.096526\text{x}10^{-01}$

TABLE II

Eight-variable final model after Step-AIC is applied

Future work shall evaluate complex non-linear models such as Support Vector Machines and Random Forests to the dataset in order to boost the model's performance.

References

[1] J. Sahs and L. Khan, "A machine learning approach to android malware detection," in *Intelligence and Security Informatics Conference (EISIC), 2012 European*, Aug 2012, pp. 141–147.

[2] N. Peiravian and X. Zhu, "Machine learning for android malware detection using permission and api calls," in *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, Nov 2013, pp. 300–305.

[3] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. Bringas, and G. lvarez, "PUMA: Permission usage to detect malware in Android," in *Proc. Int. Conference CISIS'12-ICEUTE'12-SOCO'12*, ser. Advances in Intelligent Systems and Computing, 2013, vol. 189, pp. 289–298.

[4] Z. Aung and W. Zaw, "Permission-based Android malware detection," *Int. J. Scientific and Technology Research*, vol. 2, no. 3, pp. 228–234, 2013.

[5] S. N. Hanumanthegowda, "Automated machine learning-based detection of malicious Android applications using Google Play Metadata," Master's thesis, Northeastern University, Illinois, USA, 2013.