Improving the Automatic Identification of Malicious Android Apps in Unofficial Stores through Logo Analysis

L. Vollero[1], D. Biondo[2], R. Setola[1], G. Bocci[2], R. Mammoliti[2] and A. Toma[2]
[1]*Università Campus Bio-Medico di Roma, Rome, Italy*
[2]*Sistemi Informativi, Sicurezza Informatica, Incident Prevention and Management, Poste Italiane, Rome, Italy*
{*l.vollero, r.setola*}*@unicampus.it*, {*d.biondo, boccigi2, mammoliti.rocco, tomaales*}*@posteitaliane.it*

Keywords: Security, Logo Analysis, Image Processing, Classification.

Abstract: The wide diffusion of mobile devices and the ability of users to customize their experience through applications (Apps) is opening to new problems related to privacy, security and data integrity for the mobile ecosystem. Smartphones, in general, and Android devices, in particular, are rapidly becoming emerging threat vectors of cybercrime activities. Unofficial Android markets, especially those with weak controls on published Apps, are the places where frauds may easily start and spread. Hence, the ability to identify and quickly shut down deceptive Apps is of paramount importance in the protection of users, services and infrastructures. Traditional approaches that aim at mitigating the presence of malicious Apps in unofficial markets, are based on crawlers for scanning stores and checking the words used in Apps' description. These methods works very well when the App's title, keywords and description match specific patterns that identify services to protect and the application owner or App's signature do not match expected ones. Unluckily, the performance of such methods reduce sharply when the store adopts a language that is not supported by the recognition system or the App publisher uses misleading words in the App's description. Nevertheless, App publishers always use a logo which is familiar to the user in order to highlight the application and increase the probability that the users install it. In this paper we presents a system that overcomes the limitation of traditional approaches including logo analysis in the process of App recognition. Our contribution is the definition and evaluation of a logo-based complementary system to be used in conjunction with traditional approaches based on word lists checking. The system and the performance of the proposed solution are presented and analyzed in the paper.

1 INTRODUCTION

The wide diffusion of mobile devices and the ability of users to customize their mobile experience through applications (Apps) is impacting privacy, security and data integrity in the mobile ecosystem. Indeed, applications downloaded from official and unofficial markets are not always safe due to both low quality practices in their design and development or, worse, to the malicious intents of their developers. The profile of a typical end-user, who has a very poor knowledge of the hardware and software of mobile devices, and even less of their vulnerabilities (Bertino, 2016; Seo et al., 2014), worsen this security problem making it even more problematic. The ability to install applications with just "one click", and with few or without any check on permissions, reputation of developers and application rank scratches just the surface of users wrong habits.

Two main approaches can be used to solve security flaws: (i) reactive and (ii) proactive. Under a reactive approach, as soon as a threat is detected a "patch" is released to recover the original security level of the system. The main problem with this approach is that the "patch" does not protect the user from data already disclosed and it is not always possible to apply seamlessly the patch. An inexpert user could be incapable of applying the patch or even being unaware of needing it.

Proactive approaches try to thwart security flaws before they act against users. Under proactive approaches, developers and content distributors try to have a safe environment for their user and the systems they are in charge to protect. One of the most important proactive activity is the identification of dangerous applications and their removal before users install and run them. The goal is to avoid or at least reduce the exposition of users to threats. The process of identification of dangerous Apps is today performed periodically by means of automatic tools. In partic-

Vollero, L., Biondo, D., Setola, R., Bocci, G., Mammoliti, R. and Toma, A.
Improving the Automatic Identification of Malicious Android Apps in Unofficial Stores through Logo Analysis.
DOI: 10.5220/0006270305670572
In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*, pages 567-572
ISBN: 978-989-758-209-7

ular, market of interest are explored by special software components, crawlers, that collect information related to Apps, analyze them and generate alerts to inform office delegated of Apps removal from stores. The main limitation of such an approach is that market exploration and alerts production are driven by keywords and this may have limitations when, for instance, the language of the market is not supported or the malicious developer does not use expected patterns of keywords in the description of Apps.

In this paper we propose to extend the traditional proactive approach with a logo recognition component. The idea is that even when the keywords are completely unrelated to the distributed App, at least the logo resembles that of the application the developer wants to use in its fraudulent intents. Hence, a logo recognition system can fruitfully enhance the traditional approach based on keywords.

The paper is organized as follows. In Section 2 we describe the typical management of Mobile Apps security with the description of a real case scenario. The proposed approach is defined and analyzed in Sections 3 and 4, whereas in Sectio 5 a performance assessment is presented. Section 6 ends the paper with conclusions and final remarks.

2 SECURITY MANAGEMENT

The Forum of Incident Response and Security Teams (FIRST, 2016) is an international organization, founded in 1990, which hosts the Computer Emergency Response Teams (CERT) coordination center (CERT, 2016). There are more than 350 CERTs world-wide. Their purpose is to collect and process data about cyber security threats as well as defining countermeasures and guidelines for the various stakeholders. A CERT carries out different tasks, being incident management one of the most important. The goal of the incident management process is to continuously increase the preparation and protection levels. To this aim, incident management goes through the phases depicted in Figure 1.

The process is constantly fed by data about ongoing security events. For instance, obtained through monitoring some resources, requests from citizens and other entities, reports of known authorities and more. As a result of the process, the CERT produces a number of artefacts including security reports and countermeasures.

Internally, the incident management process relies on three tasks.

- **Detection**, which is carried out to precisely identify and characterize the security event and its features, e.g., to distinguish between system vulnerabilities and malware reports.

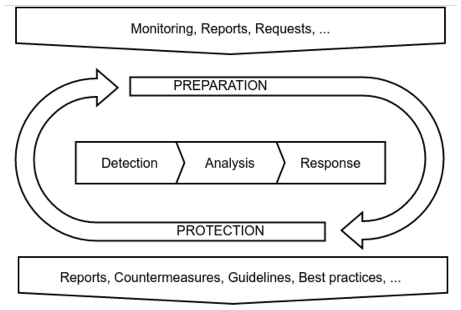
Monitoring, Reports, Requests, ...
PREPARATION
Detection
Analysis
Response
PROTECTION
Reports, Countermeasures, Guidelines, Best practices, ...

Figure 1: Incident Management.

- **Analysis**, which specifically targets a security event with dedicated techniques to effectively assess it and to understand its scope and implications.
- **Response**, which generates the appropriate security artefacts for the event, e.g., countermeasures for attacks and best practices to avoid a vulnerability.

Clearly, efficiency and effectiveness are fundamental for minimizing the impact of such security events. These requirements are typically not met by manual processes and automation is often necessary. In this paper we use as model Poste Italiane, which is the largest infrastructure in Italy in the area of mail delivery, logistics, financial and insurance services. Poste Italiane offers a wide array of services to citizens, businesses and to the Public Administration and some of them are based on Apps.

Poste Italiane (PI) hosts one of the active CERTs in Italy (PI CERT, 2016). For the reasons stated above, the PI CERT has a strong interest in developing and acquiring new technologies for improving its responsiveness and effectiveness. About the mobile applications security, on which Poste Italiane is investing more and more as their digital service delivery tool, the CERT has made a security methodology (as shown in Figure 2) that allows to target monitoring and analysis of the entire mobile application asset.

The methodology targets all the existing Apps that might negatively affect the assets and reputation of Poste Italiane. For instance, tampered versions of official Apps released on blackmarkets might carry malicious instructions. Also, third party Apps interacting with the digital services infrastructure of Poste Italiane might misuse or compromise some sensitive data of the users. Clearly, the complexity of the monitoring activity largely depends on the total number of applications and markets to be controlled. Basically, Poste Italiane has 19 official Android application released on Google Play and 12 iOS applica-

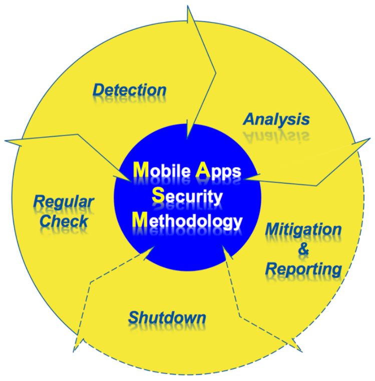

Figure 2: Mobile Application Security Methodology.

tions released on Apple Store. Nevertheless, approximately 65 instances exist on approximately 70 different markets currently known and monitored. Moreover, other 19 third party, unofficial applications have been discovered to interact with the digital services of Poste Italiane. In total there are approximately 150 instances of these Apps appearing in the 70 monitored markets.

The security methodology and monitoring is based on 5-steps. The App Detection phase relies on the search functionality provided by every market to discover new instances of applications referring to Poste Italiane in their name or description. Web crawlers can be also applied to carry out a similar operation. If an application refers somehow to Poste Italiane it does not immediately imply they actually carry out some interaction. Once the application is recognized as relevant, it undergoes the Analysis phase that allow to bring out any security issue (violation of consumer privacy, presence of malware, clone apps, etc ...). The Apps analysis is based on the three-tier approach summarized below:

(i) The static technical analysis of mobile Apps is performed without actually executing them. The analysis is performed by examining the application "as-is", and mainly through code inspection. So, it is based on a set of activities that have no interactions with the mobile Apps and therefore do not consider elements which could influence the execution; in fact, these characteristic features may be closely connected to the environment in which the App is executed and derived from the exchange of messages/signals with external entities (eg. other Apps).

(ii) The dynamic technical analysis of mobile Apps is performed at run-time, during their execution in a controlled environment (i.e. sandbox). The analysis is performed with the aim to evaluate interactions between mobile phone and other entities (e.g. web servers, local/remote files, other Apps, etc...).

(iii) In order to have a big-picture of the security levels related to analyzed Apps, it is performed a legal analysis aimed at identifying violations of contractual obligations and current laws. The results of the legal analysis are shared with the relevant corporate functions within the Directorate of Legal Affairs.

The Mitigation and Reporting phase draws up detailed report accompanied by objective evidence obtained from the analysis and defines techniques for the mitigation/contrast of the issues found during the analysis phase. A shutdown phase performs severe action against malicious/unauthorized mobile Apps, like remove request towards the market owner to delete the suspicious or obsoletes apps from the "net". Finally the Regular Check phase performs continuous monitoring of Apps detected and analyzed in the previous phases.

The approach presented above works very well when the search engine and/or the crawler for the specific market works well and the applications have meaningful name, keywords and description. However, this may not be the case when the App's publisher wants to hide its application from crawlers or the language of the store is not supported by the crawler itself. In such cases a new paradigm is needed that may complement the standard way of dealing with the identification of potentially malicious Apps. In next Section we introduce this methodology, first describing a model for logo analysis, then discussing its implementation and preliminary results obtained from a set of markets.

3 THE LOGO APPROACH

The approach proposed in this paper is based on logo analysis and can be applied in scenarios where the App's developer uses the logo as the main tool to fraud the users. Indeed, it is a common habit for users to thrust familiar images and this may represent a vulnerability easy to exploit. Several strategies for image recognition can be used in practice. The choice of the classification approach mainly resides in the principal characteristics taken into account for the images to be identified. Due to the high variability in logo shape, color, orientation, texture complexity, we opted for the Bag-Of-Visual-Words (BoVW) approach (Csurka et al., 2004), a method that deals with the problem of generic visual categorization.

BoVW is a method that mimics to computer vision from the semantic identification of text in natural language processing. In natural language processing the identification of text can be based on relative incidences of different words: e.g. the prevalence of scientific terms indicates with high probability a scientific text (Harris, 1954).

Since images are not composed of discrete words, the BoVW method requires the construction of a mapping from images to a "vocabulary". Moreover, this vocabulary has to be constructed to properly express images. Several strategies can be employed in vocabulary construction, and we chose SURF (Speeded Up Robust Features) features (Bay et al., 2008) as words of our vocabulary.

3.1 Vocabulary Construction

Vocabulary construction is performed on a set of training images, composed of samples from all the classes that the final classifier will be able to detect. In performing this task, SURF starts finding points of interest in images at different scales. In practice, points of interest are image's locations that exhibit a very high values of the Hessian determinant. In particular, given an image point $\mathbf{p} = (n, m)$ and being $I(n, m)$ its intensity[1], the Hessian matrix $\mathrm{H}(\mathbf{p}, \sigma)$ at point $\mathbf{p}$ and scale σ is:

$$\mathrm{H}(\mathbf{p}, \sigma) = \begin{pmatrix} \mathrm{L}_{r,r}(\mathbf{p}, \sigma) & \mathrm{L}_{r,c}(\mathbf{p}, \sigma) \\ \mathrm{L}_{c,r}(\mathbf{p}, \sigma) & \mathrm{L}_{c,c}(\mathbf{p}, \sigma) \end{pmatrix}$$

where $\mathrm{L}_{s,t}(\mathbf{p}, \sigma)$ is the derivative with respect to s and t, where s and t are either rows (r) or columns (c), of the smoothed image. The smoothing controls the scale at which the features are detected through the σ parameter which is the standard deviation of the gaussian smoother used before the derivative operator that computes $\mathrm{L}_{s,t}(\mathbf{p}, \sigma)$. In order to control noise, a non-maximum suppression logic is used. In our implementation we used a $(3,3,3)$ box for non-maximum suppression, where the three values represents respectively the distances in rows, columns and scale.

Given the points of interest present in the image, the vocabulary is then constructed. For any point, a local description vector of the image activity is extracted, i.e. a 3D neighborhood of the point which is vectorized. These descriptors are then processed by a vector quantization algorithm. In our implementation we used k-means clustering and centroids to represent each cluster. The final centroids represent the words of our vocabulary.

[1]We consider grayscale images, but all the reasoning can be adapted easily to color images.

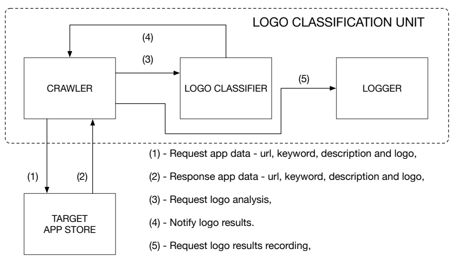
LOGO CLASSIFICATION UNIT
(4)
(3)
CRAWLER
LOGO CLASSIFIER
(5)
LOGGER
(1)
(2)
TARGET APP STORE
(1) - Request app data - url, keyword, description and logo,
(2) - Response app data - url, keyword, description and logo,
(3) - Request logo analysis,
(4) - Notify logo results.
(5) - Request logo results recording,

Figure 3: Mobile Application Security Methodology.

3.2 The Logo Classifier

The construction of the classification process requires a training phase, where the characteristics of each class are encoded into the classification logic. This process is performed by using the vocabulary constructed as described above in Section 3.1. In particular, all the images belonging to the training set are mapped to features and, then, by means of a nearest neighbor approximation to histograms of visual words occurrences.

The representation of images as histograms is the starting point for the training of the chosen classifier. In this work we use a multiclass linear Support Vector Machine (SVM) classifier (Hsu and Lin, 2002), fed with histograms and providing as output the estimated class of the image.

4 THE CLASSIFICATION UNIT

The logo classification unit is a software component that is designed to integrate a crawler and a logo classifier. The structure of this unit is depicted in Figure 3, where we assume that the unit is operating on a specific App store. The replication of this unit, or of the crawler inside it, adapts the system to work in the analysis of several stores.

The crawler visits the target store and grabs the logo of each App of interest. Other information are grabbed together with logos, such as App titles, urls, keywords and descriptions. These information are used for logging purposes or for other processing tasks complementary to logo analysis (keyword search, semantic analysis, etc ...). When limited to the recognition of logos, as in the model of Figure 3, the crawler sends each received logo to the classification module and it receives back a classification outcome. The classification outcome reports classification results and fidelity scores for each results. These information are used to flag suspicious Apps and to drive the decisions on store exploration.

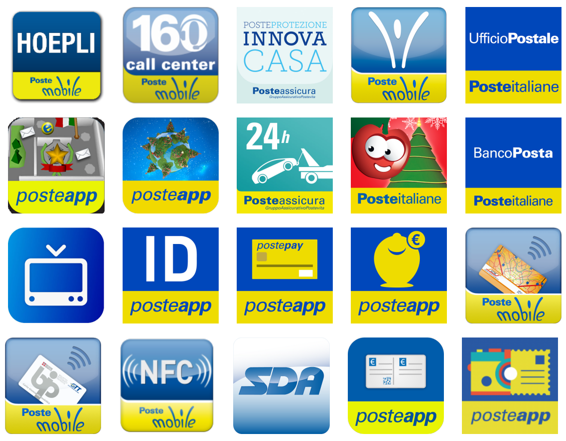
Figure 4: The original logos of PI Apps.

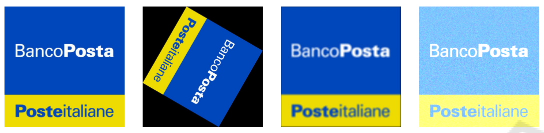
Figure 5: A PI logo subject to different masquerading transformations. From left to right: compression, rotation, blurring and noise.

5 PERFORMANCE

In this section we discuss the performance attainable by the classification process. The aim is to quantify the goodness of the approach in two different tasks:

(i) identification of an official logo even under masquerading,

(ii) identification of a logo image that resembles that of a reference App or a group of reference Apps.

When we refer to an official application we consider a logo of Poste Italiana as reference and, in particular, we consider the logos shown in Figure 4. The database used in the following tests is composed of about 1500 images, with 479 images belonging to general Apps extracted randomly from different stores, 13 images marked as suspect by visual inspection and 20 groups of 48 images of PI logos. Each one of the latter groups represents a specific PI App and the number of images is related to a set of masquerading transformations applied to them. Database information and the type of transformations used to masquerade PI logos are summarized in Tables 1 and 2. An example of masquerading transformations is presented in Figure 5. The App Markets used to obtain logos of generic Apps are those listed in Table 3 and their are the tail results on keywords for PI Apps.

To test the goodness in task (i), we performed two tests. In both we trained 20 different classifiers, one

Table 1: Database Description.

Group	Number	Description
Positive	20	PI logos
Suspect	13	Logos similar to PI ones
Negative	479	Generic App logos

Table 2: Transformation applied to PI logos.

Type	Description
Compression	Raw to JPEG
Noise	Gaussian noise
Rotation	Rotation for different angles
Resize	Different resolution
Blurring	blurring with several smoothing factors

for each PI logo. In the former test we used two classes, one represented by the logo to recognize and the other by the union of all the other logos. In the latter test, we used three classes, one for the logo to recognize, one for logos marked as Suspect and the other PI logos, and one for all the other logos. In both tests, the training was performed on a subset (random selection of half of images in each class). Obtained average results on the test sets, composed of all images that were not included in the training set, are summarized in Table 4 and 5.

(i) When analyzing the first test, we observe that results show the effectiveness of the approach in recognizing the logos of Apps to protect (last row of Table 4). Indeed, the fraction of False Negative is always negligible and this represent a very useful characteristic of the approach: we are almost sure to detect the malicious use of a real logo of interest. Conversely, the fraction of False Positives is close to 0.1. However, this does not represent a problem in the application we are considering where False Negatives have an impact which is more important than False Positives. Moreover, when inspecting results, most of the False Positives are PI logos included in the Negative class before the training of the classifier, and this represents an element of robustness of the approach.

(ii) Analogous considerations can be stated for the scenario with three classes. In this case, however, we can observe the very important role that the Suspect class assumes. Again the True Positive rate ensures a very robust identification of PI logos, with a very marginal uncertainty with other classes. As stated in case (i) this is a very positive aspect of the approach from a security point of view. The behavior of the classifier with respect to the Negative class is comparable to what observed in case (i), with the exception that the Suspect class replace the Positive class in attracting Negative samples. Samples of the Suspect class, however, are not well recognized and a lot of False

Table 3: Unofficial Markets.

Market	URL
Apkdownloader	`https://apk-dl.com`
Aptoide	`http://www.aptoide.com`
AppBrain	`http://www.appbrain.com`

Table 4: Results for Single PI logo vs Extended Negative.

	Predicted	
Known	E-Negative	Positive
E-Negative	0.92 ± 0.02	0.08 ± 0.02
Positive	0.00 ± 0.00	1.00 ± 0.01

Negative are present. This is obviously a very negative aspect. This behavior can be explained by analyzing the composition of the Suspect class, which is not as well characterized as the Positive class of logos. Indeed the Suspect class is composed of images not homogeneous if we exclude the PI logos not object of identification in the specific test. The behavior is, hence, as that observed on the Negative class, but with a reduced ability to attract samples, due to a less consistent set of images in this class.

To investigate further the False Positive effect of the specific classification model used in the paper, we conducted a test based on three global classes: Positive, including all the PI logos in their original shape and under masquerading, Suspect logos and Negative logos. The performance have been investigated by means of the leave-one-out strategy. Again we found a perfect identification of the Positive class, with a True Positive rate equal to 1. The results for the Negative class are reported in Figure 6, confirming the performance already observed in case (ii) above.

6 CONCLUSIONS

In this paper we present a computer vision system for the automatic identification of potentially malicious Apps by means of logos analysis. The idea behind the proposed system is to improve the performance of proactive security approaches, increasing the element analyzed in the identification of Apps distributed through official and unofficial stores. The proposed system is designed to analyze the companion logos of Apps and to check whether they are similar to those of groups of Apps to protect. The proposed system, hence, complements traditional approaches based on words checking and makes possible the identification of malicious Apps in a wider set of scenarios. A possible implementation of the classification unit devoted to logo recognition and preliminary results are presented in the paper. Obtained results show that the method is effective also when trained on a very small dataset and encourage for further investigations.

Table 5: Results for Single PI logo vs Suspect vs Negative.

	Predicted		
Known	Negative	Suspect	Positive
Negative	0.81 ± 0.01	0.18 ± 0.02	0.01 ± 0.01
Suspect	0.52 ± 0.03	0.47 ± 0.04	0.01 ± 0.01
Positive	0.00 ± 0.00	0.01 ± 0.01	0.99 ± 0.01

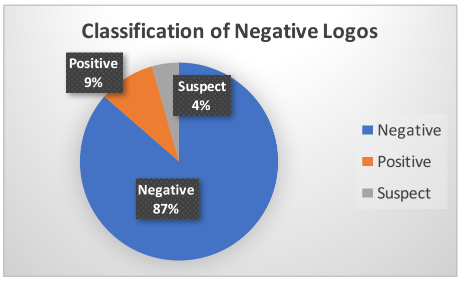

Figure 6: Results of the global three class classification.

REFERENCES

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359. Similarity Matching in Computer Vision and Multimedia.

Bertino, E. (2016). Securing mobile applications. *Computer*, 49(2):9–9.

CERT (2016). http://cert.org/.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

FIRST (2016). https://www.first.org/.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425.

PI CERT (2016). http://www.picert.it.

Seo, S.-H., Gupta, A., Sallam, A. M., Bertino, E., and Yim, K. (2014). Detecting mobile malware threats to homeland security through static analysis. *Journal of Network and Computer Applications*, 38:43 – 53.