Mobile Malware detection with permissions and intents analysis

Fauzia Idrees Abro, Muttukrishnan Rajarajana and Thomas Chen
City University of London, United Kingdom

***Abstract*—In this paper, we present a novel permissions and intents based framework for identifying Android malware apps. The proposed approach, when applied to 1,745 real world applications, provides 99.8% accuracy.**

I. Introduction

Android being the most widely used platform for smartphones is under constant attacks [1, 2, 3]. We propose a malware detection approach which classifies apps against certain combinations of permissions and intents which are unique to malware apps [4, 6]. We evaluate the efficacy of proposed approach by applying machine learning algorithms. We apply the ensemble methods to optimize the results. The fact that permissions and intents facilitate app collusion, our proposed solution is suitable for detection of colluding apps in addition to and unknown malware apps [7]

A. Contributions

The main contributions presented in this paper are:

- This work combines permissions and intents of applications to generate a distinguishing matrix that is used for efficient and accurate detection of malware and its associated families. Our method is capable of achieving 99% detection accuracy by combining permissions and intents.
- Classification of permissions and intents into four groups: Dangerous permissions, dangerous intents, normal permissions and normal intents.
- Generating a distinguishing matrix based on these features for efficient and accurate malware detection.
- We use ensemble methods to optimize the classification results.

B. Challenges

Existing anti-virus solutions are not capable of eliminating the exponentially increasing malware threats due to their reliance on signature-based detection [3]. Moreover, resource constrained smartphones are unsuited for continuous malware scanning [5]. Code obfuscation techniques used by malware are another challenge [1, 5].

II. METHODOLOGY

The proposed system is shown in Fig. 1. It consists of three main stages: Feature extraction, Pre-processing, and Classification. The feature extraction stage analyses the manifest file and extracts the permissions and intents. The pre-processor stage processes the extracted data to generate the vector dataset in an ARFF file format. The classifier stage takes each monitored vector as input and classifies the data set using trained classifier. Six machine learning classifiers: Nave Bayesian, Decision Tree, Decision Table, Random Forest, Sequential Minimal Optimization and Multilateral Perceptron (MLP) are used for classification. Finally, the reporter stage generates notifications for the user based on the classifier results.

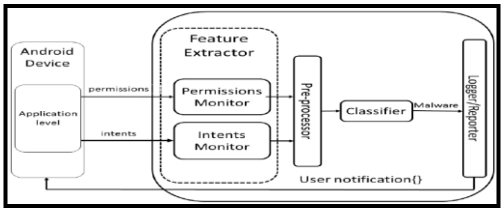

Fig.1. Diagram of proposed system

III. EVALUATIONS AND RESULTS

The proposed approach is evaluated on 1750 malware and benign apps collected from known sources [7]. The overall usage trend of dangerous permissions and dangerous intents in malware and benign apps is shown in Fig. 2 and 3 respectively. Number of permissions and intents used by different categories of apps are shown with arrows in Fig. 4 and 5 respectively.

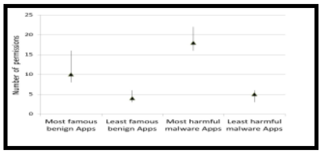

Fig. 2 Number of intents used by apps

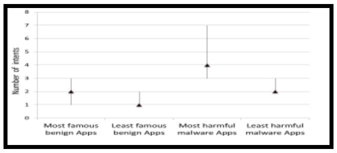

Fig.3 Number of permissions used by apps

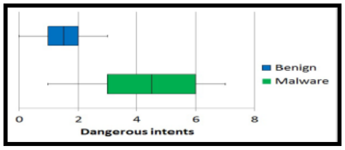

Fig. 4. Box plots for dangerous intents

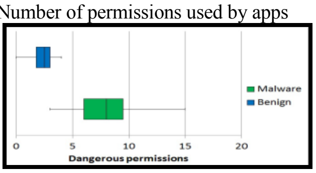

Fig.5. Box plots for dangerous permissions

Table 1 compares the performance of six classifiers against seven performance metrics. Results are further optimized with ensemble methods (Table 2). The product of probabilities method yields the best detection results.

TABLE I. COMPARISON OF CLASSIFIERS

Algorithm	***TPR***	***FPR***	***Precision***	***F1 score***	***Recall***	***AUC***	***Time***
MLP	0.993	0.006	0.995	0.995	0.995	0.996	1.18
Decision Table	0.993	0.006	0.995	0.996	0.996	0.996	0.23
Decision Tree	0.992	0.011	0.993	0.992	0.993	0.992	0.01
Naïve Bayesian	0.982	0.012	0.989	0.988	0.989	0.997	0.01
Random Forest	0.982	0.007	0.985	0.985	0.985	0.989	0.43
SMO	0.952	0.033	0.956	0.956	0.956	0.978	0.24

TABLE II. COMPARISON OF ENSEMBLE METHODS

Method	***TPR***	***FPR***	***Precision***	***F1 score***	***Recall***	***AUC***
Average Probability	0.972	0.012	0.975	0.975	0.975	0.982
Product Probability	0.998	0.011	0.998	0.997	0.997	0.998
Majority vote	0.982	0.021	0.986	0.986	0.986	0.989

IV. CONCLUSION

The proposed approach exploits the usage of permissions and intents by mobile apps to efficiently and accurately distinguish detect the mobile malware and their associated families.

References

[1] Adrienne Porter Felt et al., “Android permissions demystified.” Computer and communications security, pp 627-638, ACM, 2011.

[2] Fauzia Idrees et al., "Framework for distributed and self-healing hybrid intrusion detection and prevention system." ICTC, pp. 277-282. IEEE, 2013.

[3] Chao Yang et al., “Droidminer: Automated mining and characterization of ne-grained malicious behaviours in Android apps.” ESORICS, pp 163-182, Springer, 2014.

[4] Fauzia Idrees et al., "Investigating the android intents and permissions for malware detection." Wireless and Mobile Computing, pp. 354-358. IEEE, 2014.

[5] Daniel Arp et al., “DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket.” NDSS, 2014.

[6] Fauzia Idrees et al., "War against mobile malware with cloud computing and machine learning forces." CloudNet, pp. 278-280. IEEE, 2014.

[7] Fauzia Idrees et al., “PInDroid: Malware detection with ensemble learning methods.” Submitted to Elsevier.