Android Malware Identification Based on Traffic Analysis

Rong Chen[1,2], Yangyang Li[1(✉)], and Weiwei Fang[2]

[1] Innovation Center and Mobile Internet Development and Research Center, China Academy of Electronics and Information Technology, Beijing 100041, China
yli@csdslab.net
[2] Beijing Jiaotong University, Beijing 100044, China
{17127054,fangww}@bjtu.edu.cn

Abstract. As numerous new techniques for Android malware attacks have growingly emerged and evolved, Android malware identification is extremely crucial to prevent mobile applications from being hacked. Machine learning techniques have shown extraordinary capabilities in various fields. A common problem with existing research of malware traffic identification based on machine learning approaches is the need to design a set of features that accurately reflect network traffic characteristics. Obtaining a high accuracy for identifying Android malware traffic is also a challenging problem. This paper analyses the Android malware traffic and extract 15 features which is a combination of time-related network flow feature and packets feature. We then use three supervised machine learning methods to identify Android malware traffic. Experimental results show that the feature set we proposed can accurately characterize the traffic and all three classifiers achieve high accuracy.

Keywords: Malware traffic · Traffic analysis · Traffic classification · Machine learning

1 Introduction

With the development and popularization of smart phones, smart phones have become a very important part of people's life. It offers a wide variety of applications to meet people's daily needs [1], and more and more users store their private information in their smart phones. According to statistics from data Internet statistics company Statista, in 2016, there were 2.1 billion smart phone users worldwide, and the number is expected to grow to 2.87 billion in 2020 [2]. The widespread deployment of WIFI networks and the large number of applications available in the application market [3], compared with traditional network

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grants 2017JBM021 and 2016JBZ006, and CETC Joint Fund under Grant 6141B08020101.

X. Sun et al. (Eds.): ICAIS 2019, LNCS 11632, pp. 293–303, 2019.
https://doi.org/10.1007/978-3-030-24274-9_26

traffic, have enabled mobile devices not only to involve traditional communication activities (such as making voice calls and sending short messages), but also to apply to more advanced scenarios such as finance, online games and e-shopping. As mobile devices tend to store owners' private data [4] (such as contacts, photos, videos and GPS locations), more and more attackers and traffic analysts are targeting the network traffic they generate in an attempt to mine useful information. Google's Android has become the most popular mobile platform overtaking other operating systems. Such increasing popularity of Android smart phones has attracted malicious app developers as well. According to the statistics given in [5], among all malwares targeting mobile devices, the share of Android malwares is higher than 46%. Another recent report also alerts that Android malwares have grown around 400% since summer 2010 [6]. New techniques for Android malware attacks are emerging. Given this significant growth of Android malware, there is a pressing need to effectively mitigate or defense against them. In this paper, we focus on malware traffic and we extracted 15 features from raw network traffic. We propose a machine leaning model using three supervised machine learning methods for android malware traffic identification. Organisation of paper is as follows. Section 2 overviews related work. Section 3 demonstrates methodology. Section 4 is about experimental study and results. Conclusion and Future Work are depicted in Sect. 5.

2 Related Work

Existing technology proposed for android malware classification falls into three categories [7]: port-based identification [8], deep packet inspection (DPI) [9] identification and the machine learning (ML) identification [12]. Port-based identification was used in the past to associate applications with network connections, but the accuracy of this method is decreasing with the increased use of dynamic ports [10] and applications evading firewalls. Despite the decreased accuracy, port numbers are often utilized as one of the packet features. Furthermore, port-based identification is still quite often used to establish a ground truth for traffic identification experiments. Finsterbusch [11] summarized current main DPI-based traffic identification methods. DPI technology is influenced by network traffic encryption measures. At present, the mainstream research mainly uses machine learning methods [13]. Machine learning approach of traffic identification attracts a lot of research in academia [14–16], and related work mainly focused on how to choose a better dataset. Dhote et al. [17] provided a survey on feature selection techniques of internet traffic identification. There are generally two kinds of traffic features that are mainly used in machine learning methods [18]. One is flow features, that is, the communication of the two sides of all the data packets reflected. The other is packet features, which are the features of each packet. Wang [19], Mauro [20] and Cheng [21] apply flow features to research P2P traffic, WebRTC, Coull et al. [20] use packet features to research iMessage traffic. Korczynski [22] and Koch [23] apply packet features to identify encrypted traffic. The corresponding classifiers are C4.5 decision tree; Naive Bayes and random forest respectively.

Aghaei et al. [24] proposed a identification method with flow features and C4.5 decision tree classifier on proxy traffic. Xu [25] proposed a identification method with only time related flow features on both regular encrypted traffic and protocol encapsulated traffic. A few researchers, such as, Du. [26] and Alshammari. [27] use combination of flow and packets features to identify [18] encrypted traffic. Most of the researches employ supervised machine learning Methods [31].

3 Methodology

3.1 Dataset

Arash [28] published CICAndMal dataset which includes four types of Android malware traffic. In this paper, we select three types of Android malware traffic from CICAndMal dataset. Each type of Android malware traffic includes 10 malware families and we randomly choose one pcap file from each malware family. Therefore, every malware traffic consists of 10 pcap files. And the benign traffic data also comes from Arash [29]. It consists of 173 pcap files which are generated by 1,500 benign Android applications from google play. Table 1 shows the detailed malware family of three types of malware traffic in our dataset. The size of our Android malware dataset is 3.2 GB, and the format is pcap.

Table 1. The details of Android malware families.

Traffic type	Malware families	
Adware	Dowgin family	Ewind family
	Feiwo family	Gooligan family
	Kemoge family	Koodous family
	Mobidash family	Selfmite family
	Shuanet family	Youmi family
Ransomware	Charger family	Jisut family
	Koler family	LockerPin family
	Simplocker family	Pletor family
	PornDroid family	RansomBO family
	Svpeng family	WannaLocker family
Scareware	AndroidDefender 17	AndroidSpy.277 family
	AV for Android family	AVpass family
	FakeApp family	FakeApp.AL family
	FakeAV family	FakeJobOffer family
	FakeTaoBao family	Penetho family

3.2 Model

In this paper, we propose an Android malware traffic identification model using a machine learning (ML) architecture. Figure 1 shows the overview of our proposed Android malware traffic identification method. Generally, this model consists of flow seperation, feature extraction and training machine learning classifiers.

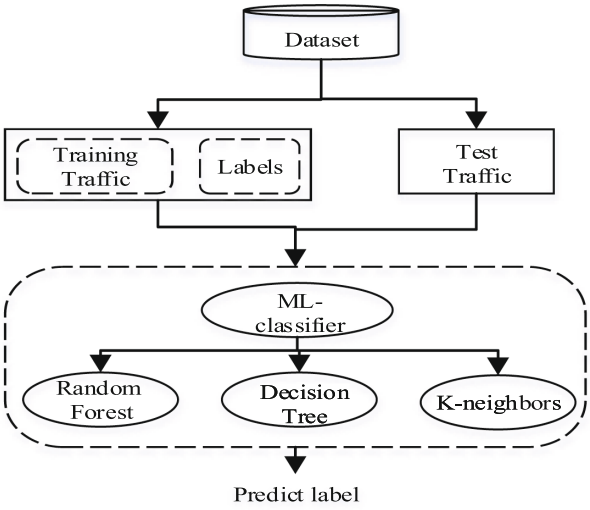

Fig. 1. The architecture of machine learning model.

A. Flow Seperation
Machine learning based traffic identification approach need to split continuous traffic to discrete units based on certain granularity at first [30]. The flow seperation module, We use flow and session which were also used by most researchers. Firstly, we apply five-tuples (source IP, source port, destination IP, destination port, transport layer protocol) to divide continuous network traffic into discrete flows. A session is a bi-directional flow that its source IP and destination IP can swap. For each flow, three packet series are considered: incoming packets only, outgoing packets only, and bi-directional traffic (i.e. both incoming and outgoing packets). A program written by Java languages are used to split continuous traffic to discrete flows. Before flows are passed on to the next stage, they are discarded if they contain any TCP retransmissions or other errors.

B. Feature Extraction
During the feature extraction, all traffic files (.pcap) are processed automatically generate feature sets (.csv). Feature extraction involves deriving 15 features from each flow. We extract simple packet features (e.g. packet length) and time-related

flow feature from packets header portion of every session. These statistical features were computed using the Python pandas libraries. Subsequently, those features are labeled and then fed into supervised machine learning algorithms for classifying benign and malware traffic. The 15 selected features are showed in Table 2.

Table 2. Feature extracted from traffic flows.

ID	Feature	Description
1	Flow duration	Duration of the flow in Microsecond
2	flowBytesPerSecond	Number of flow bytes per second
3	total_opackets	Total packets in the outgoing direction
4	total_ipackets	Total packets in the incoming direction
5	min_opktl	Minimum size of packet in outgoing direction
6	max_opktl	Maximum size of packet in outgoing direction
7	mean_opktl	Mean size of packet in outgoing direction
8	std_opktl	Standard deviation size of packets in outgoing direction
9	min_ipktl	Minimum size of packet in incoming direction
10	max_ipktl	Maximum size of packet in incoming direction
11	mean_ipktl	Mean size of packet in incoming direction
12	std_ipktl	Standard deviation size of packets in incoming direction
13	min_flowpktl	Minimum length of a flow
14	max_flowpktl	Maximum length of a flow
15	mean_flowpktl	Mean length of a flow

C. Training Classifier

Machine learning can be used to automatically discover the rules by analyzing the data, and then the rules can be used to predict unknown data. Three classfiers were chosen because they are particularly suited for predicting classes (in our case, network traffic) when trained with the features that we extracted from network flows.

Random Forest algorithms achieve best performance. A Random Forest classifier is an ensemble method that uses multiple weaker learners to build a stronger learner. This classifier constructs multiple decision trees during training and then chooses the mode of the classes output by the individual trees. The construction process of random forest can be described as follows.

Algorithm 1. Random Forest Algorithm.

Require:
Training sample set T, Sample to be classified x

Ensure:
Sample label y

1: Random sampling of rows: assuming that the number of training set samples is N, the training set of a decision tree is constituted by random sampling N times in the way of putting back;
2: Column random sampling: in the attribute set of the training set (assuming that there are M attributes), randomly select the subset containing M (m l M) attributes;
3: Decision tree generation: select an optimal attribute in the sub-set of m attributes for complete splitting to construct the decision tree. No pruning is needed in the splitting process to maximize the growth of each decision tree;
4: Generate random forest: repeat steps 1-3 to grow multiple decision trees to generate forest;

K-Neighbors is one of the simplest and most well-known classification algorithms. It relies on the assumption that nearby data sets have the same label with high probability. The algorithm implementation is described as follows.

Algorithm 2. K-Neighbors Algorithm.

Require:
Training sample set T, Sample to be classified x, Number of neighbors k

Ensure:
Sample label y

1: first initialize the distance as the maximum distance;
2: calculate the distance dist between unknown samples and each training sample;
3: obtain the maximum distance max_dist in the current k closest samples;
4: If dist is less than max_dist, the training sample is taken as the k-nearest neighbor sample;
5: repeat steps 2, 3 and 4 until the distance between the unknown sample and all training samples was calculated;
6: count the occurrence times of each category in k nearest neighbor samples;
7: select the category with the highest occurrence frequency as the category of the unknown sample;

Decision tree is a prediction model, which represents a mapping between object attributes and object values. Each node in the tree represents the judgment condition of object attributes, and its branches represent the objects that meet the node conditions. The leaf nodes of the tree represent the predicted results to which the object belongs. The generation process of a decision tree is mainly divided into the following three parts: feature selection, constructing decision tree, decision tree pruning. The first is feature selection. Selecting an

appropriate feature as the judgment node can quickly classify and reduce the depth of the decision tree. The goal of decision tree is to classify data sets according to corresponding class labels. In this paper, we use the gini coefficient ratio to select features. In the classification problem, assuming that there are K categories and the probability of the K^{th} category is P_k, the gini coefficient can be expressed as:

$$Gini(p) = \sum_{1}^{k} P_k(1 - P_k) = 1 - \sum_{1}^{k} (P_k)^2 \tag{1}$$

According to the formula, the higher the degree of data mixing in the data set, the higher the gini index. When dataset D has only one category, the gini index has a minimum value of 0. If the selected attribute is A, then the calculation formula for the gini index of the data set D after splitting is as follows:

$$Gini_A(D) = 1 - \sum_{1}^{k} \frac{D_j}{D} Gini(D_j) \tag{2}$$

Since the algorithm of decision tree is very easy to overfit, it must be pruned for the generated decision tree. There are many algorithms for pruning, and there is room for optimization in the pruning method of decision tree. There are two main ideas. One is pre-pruning, that is, when the decision tree is generated, the pruning is decided. The other is post-pruning, that is, construct the decision tree firstly, and then pruning through cross-validation. In our experiment, We chose the latter and got a good classification result.

4 Experiment and Results

Our experiments included two types of classification tasks, namely, binary classification and multi classification. Specifically, the binary classification task included benign and malware. The multi classification task included four types of classes, i.e., scareware, ransomware, adware and benign. Among all of the two experiments, the proportion of the training and test set is 8:2. Table 3 presents the distribution of traffic records in our dataset. We evaluate the performance of the three machine learning. In this section, we briefly introduce evaluation metrics for the performance analysis. Finally, we discuss experimental results of two experiments.

Table 3. Distribution of traffic samples in our dataset.

Type	Malware traffic			Benign traffic
	Adware	Ransomware	Scareware	Benign
Total Samples	34882	38159	34656	137105
Train Samples	27905	30527	2722	109684
Test Samples	6977	7632	6932	27421

4.1 Evaluation Metrics

In general, we use the confusion matrix to evaluate the performance of the machine learning algorithm. The confusion matrix contains three metrics, i.e., precision (P), recall (R), F-measure (F). These confusion metrics are made up of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Specifically, TP and TN are the number of instances predicted correctly as malware or benign, respectively. Accordingly, FP and FN are the number of instances incorrectly predicted as malware or benign.

Precision (P): Precision presents the percentage of all samples predicted as malware traffic that are truly malware.

$$P = \frac{TP}{TP + FP} \tag{3}$$

Recall (R): Recall presents the percentage of all malware traffic samples that are predicted to be truly malware.

$$R = \frac{TP}{TP + FN} \tag{4}$$

F1-score ($F1$): $F1$ value is the harmonic mean of precision and recall which can be better to evaluate the performance.

$$F1 = \frac{2PR}{P + R} \tag{5}$$

4.2 Experimental Result

Binary Classification: In this experiment, three types of malware traffic will be labeled as malware. Therefore, there are two classes benign and malware. All three machine learning algorithms are conducted to verify the combined features for malware identification. The results of binary classification are presented in Fig. 2. We found that the values of all the evaluation metrics of all three algorithms achieved were over 85%. These results implicate that the combined feature can be used to effectively classify Android malware and benign traffic. RandomForest is the best performer, with precision of 95%, recall of 95%, F1-value of 95%. And Decision Tree performs slightly worse, with precision of 93%, recall of 92%, F1-value of 92%. K-Neighbors performs the worst, with precision of 85%, recall of 86%, F1-value of 84%.

Multi Classification: On the basics of binary classification, we use the same algorithm to identify specific Android malware traffic, i.e., adware, ransomware, scareware. Experiments are conducted to evaluate the performance of RandomForest, Decision Tree, K-Neighbors. As observed from Fig. 3, the precision of three methods is higher than 80%. The evaluation metrics of RandomForest is the best with precision of 86%, recall of 85% and F1-value of 86%. The Decision Tree performs almost the same as RandomForest, with precision of 84%, recall of 84% and F1-value of 84%. The worst performance is from K-Neighbors algorithm with precision of 81%, recall of 81% and F1-value of 81%.

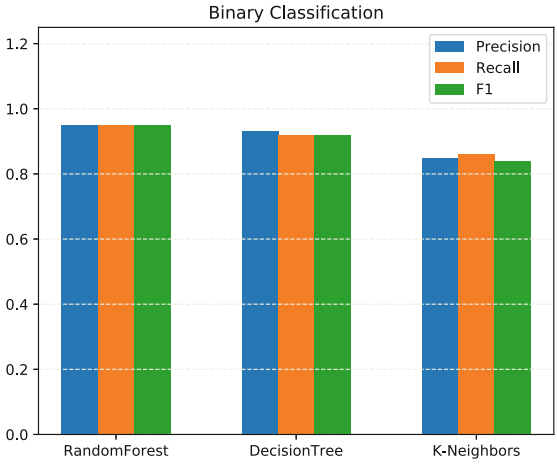

Fig. 2. The result of binary classification.

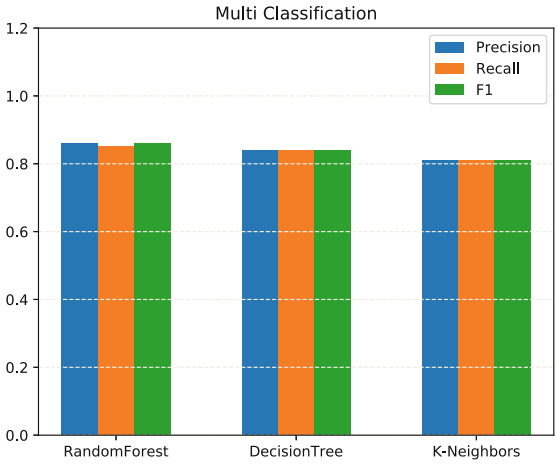

Fig. 3. The result of multi classification.

5 Conclusion and Future Work

In this paper, we have studied the time-related flow feature and packet feature to address the challenging problem of characterization of android malware traffic and identification of Android malware traffic. we proposed a set of feature and Android malware identification model which contains three common machine learning algorithms. The experimental results show that the proposed model performs well when only classifying malware and benign traffic, with an average accuracy of 95%. When identifying specific malware and benign traffic, the

performance is comparatively worse. We notice that the RandomForest classifier performs the best among all of the two experiments. As Android malware application is a fast variant, the numerous derived features contained in existing datasets may not be practical in the future. Therefore, we plan to study the methods of deep learning which can automatically learn features from traffic.

References

1. Taylor, V.F., Spolaor, R., Conti, M., et al.: AppScanner: automatic fingerprinting of smartphone apps from encrypted network traffic. In: IEEE European Symposium on Security & Privacy. IEEE (2016)
2. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/
3. Shen, M., Wei, M., Zhu, L., et al.: Certificate-aware encrypted traffic classification using second-order Markov chain. In: 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS). ACM (2016)
4. Cui, J., Zhang, Y., Cai, Z., Liu, A., Li, Y.: Securing display path for security-sensitive applications on mobile devices. CMC Comput. Mater. Continua **55**(1), 017–035 (2018)
5. Malicious mobile threats report 2011/2012. http://apo.org.au/node/29815
6. Gao, C.-X., Wu, Y.-B., Cong, W., et al.: Encrypted traffic classification based on packet length distribution of sampling sequence. J. Commun. **36**(9), 65–75 (2015)
7. Biersack, E., Callegari, C., Matijasevic, M. (eds.): Data Traffic Monitoring and Analysis. LNCS, vol. 7754. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36784-7
8. Conti, M., Mancini, L.V., Spolaor, R., et al.: Analyzing Android encrypted network traffic to identify user actions. IEEE Trans. Inf. Forensics Secur. **11**(1), 114–125 (2016)
9. Bujlow, T., Carela-Espaiol, V., Barlet-Ros, P.: Independent comparison of popular DPI tools for traffic classification. Comput. Netw. **76**, 75–89 (2015)
10. Madhukar, A., Williamson, C.: A longitudinal study of P2P traffic classification. In: IEEE International Symposium on Modeling. IEEE (2006)
11. Finsterbusch, M., Richter, C., Rocha, E., et al.: A survey of payload-based traffic classification approaches. IEEE Commun. Surv. Tutor. **16**(2), 1135–1156 (2014)
12. Feizollah, A., Anuar, N.B., Salleh, R.: Evaluation of network traffic analysis using fuzzy C-means clustering algorithm in mobile malware detection. Adv. Sci. Lett. **24**(2), 929–932 (2018)
13. Okada, Y., Ata, S., Nakamura, N., et al.: Application identification from encrypted traffic based on characteristic changes by encryption. In: 2011 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR). IEEE (2011)
14. Gui, X., Liu, J., Chi, M., et al.: Analysis of malware application based on massive network traffic. China Commun. **13**(8), 209–221 (2016)
15. Zuquete, A., Rocha, M.: Identification of source applications for enhanced traffic analysis and anomaly detection. In: IEEE International Conference on Communications (2012)
16. Kohout, J., Komrek, T., Tech, P., et al.: Learning communication patterns for malware discovery in HTTPs data. Exp. Syst. Appl. **101**, 129–142 (2018)

17. Velan, P., Permk, M., Teleda, P., et al.: A survey of methods for encrypted traffic classification and analysis. Int. J. Netw. Manag. **25**(5), 355–374 (2015)
18. Alshammari, R., Zincir-Heywood, A.N.: An investigation on the identification of VoIP traffic: case study on Gtalk and Skype. In: International Conference on Network & Service Management (2010)
19. Wang, D., Zhang, L., Yuan, Z., et al.: Characterizing application behaviors for classifying P2P traffic. In: International Conference on Computing (2014)
20. Coull, S.E., Dyer, K.P.: Traffic analysis of encrypted messaging services: Apple iMessage and beyond. ACM SIGCOMM Comput. Commun. Rev. **44**(5), 5–11 (2014)
21. Mauro, M.D., Longo, M.: Revealing encrypted WebRTC traffic via machine learning tools. In: International Joint Conference on E-business & Telecommunications. IEEE (2016)
22. Korczynski, M., Duda, A.: Markov chain fingerprinting to classify encrypted traffic. In: Infocom. IEEE (2014
23. Koch, R., Rodosek, G.D.: Command evaluation in encrypted remote sessions. In: International Conference on Network & System Security. IEEE Computer Society (2010)
24. Aghaei-Foroushani, V., Zincir-Heywood, A.: A proxy identifier based on patterns in traffic flows. In: HASE, January 2015
25. Cheng, J., Ruomeng, X., Tang, X., Sheng, V.S., Cai, C.: An abnormal network flow feature sequence prediction approach for DDoS attacks detection in big data environment. CMC Comput. Mater. Continua **55**(1), 095–119 (2018)
26. Du, Y., Zhang, R.: Design of a method for encrypted P2P traffic identification using K-means algorithm. Telecommun. Syst. **53**(1), 163–168 (2013)
27. Alshammari, R., Zincir-Heywood, A.N.: Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? Comput. Netw. **55**(6), 1326–1350 (2011)
28. Lashkari, A.H., Kadir, A.F.A., Taheri, L., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark android malware datasets and classification. In: Proceedings of the 52nd IEEE International Carnahan Conference on Security Technology (ICCST), Montreal, Quebec, Canada (2018)
29. Lashkari, A.H., Kadir, A.F.A., Taheri, L., Ghorbani, A.A.: Towards a network-based framework for android malware detection and characterization. In: Proceeding of the 15th International Conference on Privacy, Security and Trust, PST, Calgary, Canada (2017)
30. Xiao, B., Wang, Z., Liu, Q., Liu, X.: SMK-means: an improved mini batch K-means algorithm based on MapReduce with big data. CMC Comput. Mater. Continua **56**(3), 365–379 (2018)
31. Dhote, Y., Agrawal, S.: A survey on feature selection techniques for internet traffic classification. In: Computational Intelligence and Communication Networks, Jabalpur, pp. 1375–1380 (2015)