chkrithika02@gmail.com.docx

by Chkrithika02@gmail.com.docx Last

Submission date: 20-Oct-2021 10:04AM (UTC+0530)
Submission ID: 1678822229
File name: chkrithika02_gmail.com.docx (77.77K)
Word count: 3265
Character count: 17719

Malscore: A Probability Scoring Method for Malware Classification

Mounvi Podapati
Vellore Institute of Technology, School of Computer Science and Engineering,
Vellore - 632014, Tamil Nadu, India

***Abstract*- A malicious software whose intent is to disrupt the computer operations and to demolish the system and network activities is known as Malware and its threat is increasing evidently in the past few years due to advancements of the ability to breach through a secured system. A number of techniques such as Machine learning models, mathematical models have been devised to accurately identify such existing malicious software's but still on lookout for the ability to detect zero-day malwares. The paper is essentially proposes a probability scoring method for detection and classification of malware. Malware that utilizes packing and other obfuscation techniques cannot be efficiently handled by the existing static analysis approaches which are usually quick. For this reason dynamic analysis are used to overcome such issues but they result in high classification costs. An improved and immune version of static analysis has been obtained as a result of Malscore's mix of static and dynamic analysis. The concept of using Image Recognition with CNN has proven to be the reason for the same. Different techniques such as static and dynamic analysis along with CNN and SPP layers are combined for efficiently detecting malwares using the grayscale images generated from the samples. Due to the continuous evolution of malwares every day, this malware analysis predominately constitutes to be one of the important research for cyber security researchers.**

I. INTRODUCTION

It is currently identified that a large source of unknown malware is being generated from known malware sources and with the advent of different automated technologies, it was found that the rate at which the malware morphs is more rapid than it was initially assumed. This has risen the need to find the similarities between different samples to track their source origin, their operational environment and ways to tackle them. It is noticed that the static malware is used for classification wherein the dynamic analysis can be availed to fully scrutinize the behavior of different classes of malware.

Malscore, a malware categorization method, has been suggested based on both probability grading and ML approaches. Static features are extracted from raw malware in the form of gray-scale images that are an effective reflectance of the malware's skeletal and static structure, whereas dynamic features are extracted by executing them in a sandboxed environment such as a cuckoo sandbox for performing behavioral malware analysis. These are now used to train both the Classifiers of types S and D which are based on the CNN with SPP and variable n-grams with n ranging from 2 to 4 with Machine Learning. These are currently used to train both S and D Classifiers and variable n-grams with n ranging from 2 to 4 and Machine Learning. A type of probability scoring named probability threshold was proposed to combine the S and D classifiers.

The resilience of the packed malware is considerably boosted, resulting in improved classification accuracy by Malscore. When training the model with numerous malware samples into the D-classifier, the cost of the dynamic analysis significantly increases. As a consequence, all of the malware's undesirable characteristics are filtered using probability scoring, yielding only trustworthy classification results for S-classifier. This technique greatly decreases the execution time of the dynamic analysis as well as the Malscore detection cost.

Using the grayscale pictures as static characteristics along with CNN and SPP layers to classify the various kinds of malware. This can aid in minimizing the loss with respect to the malware information by picture pre-processing. DF.IDF are used with modifiable n-grams as dynamic features to extract the useful features for classification. Malscore can thus, save as much as the semantic information of the virus by allowing only the feature association. Later both these analysis, namely static and dynamic are combined to present a probability scoring which not only minimizes the cost of classification of analysis type: dynamic but also enhances the accuracy of the malware classification. The Malscore resulted with a greater accuracy along with lower categorization cost on evaluating with multiple evaluation tests on a huge and genuine datasets.

II. PROPOSED METODOLOGY

I. System Framework:

The Malscore system structure is made of training and testing. The S and D-classifiers' feature learning is included in the training, and the features comprise grayscale pictures and native API call sequences. The testing phase includes phases 1 and 2. Grayscale images are utilized as static features, coupled with CNN and SPP layers, to identify various types of malware, while n-grams are used as dynamic features to extract relevant features for classification and are employed in the examination of native API call

-sequences as part of the second Phase. During the testing phase, a probability score is assigned with the goal of shortening the detection time.

The S-classifier from the first Phase generates a 1D pvv, and pk represents the likelihood of a malware belonging to a certain malware class k. The trustworthiness of the classification result may be easily assessed by comparing the maximum of pk with the probability threshold value. Phases 1 and 2 results are afterwards supplemented against each other, but Phase 2 simply attempts to analyze samples with poor credibility.

II. CNN and SPP analysis of the Grayscale Images

A. Grayscale Image Generations

Converting malware samples into grayscale pictures has been shown to be an efficient static analysis approach, and this conversion procedure consists of performing these respectful steps: initial executable file is thought to be a binary stream of bits divided into 8 bits or a single byte. These 8 bits are then transformed into a grayscale value of 256 levels (values between 0 and 255 ie. between black and white). These grey values are now orderely transformed into a 2D matrix.
Comparing with the size of the malware file, the dimensions of the matrix, i.e. height and width, are chosen for the grayscale images. Although grayscale pictures of various sizes are created, the CNN's fully connected layer requires them to be of the same size since the image must be pre-processed to unify the image size. The pictures are segmented using all of the known methods. This conversion procedure is time-consuming for pre-processing and reduces the association between picture blocks with useful information contained in them. To address this issue, this study offers an M-CNN-based CNN-based VGGNet and SPP model.

B. Construction of M-CNN

In deeper regions of the network, M-CNN employed smaller convolution filters. An input and output layer, multiple convolutional and pooling layers comprise the framework. The SPP layer generates a vector of k x b dimensions, k denotes number of filters and b denotes the number of bins in the final convolutional layer. Because of the dimensional vector that is fixed serves as the fully linked layer's input, images of varying sizes are permitted. As a consequence of the malscore's 3-layer pyramid pooling, vectors such as 1x1x512, 2x2x512, and 4x4x512 are created and linked to a 21 x 512 dimensional vector outputted to a fully connected layer. The PVV of each malware sample is calculated using the softmax algorithm. The probability of the softmax classification result is provided, which improves classification interpretability. Apply the following equation to estimate the probability of a sample belonging to a sample x:

$$P(y = k|x) = \frac{e^{z_k}}{\sum_{k=1}^{63} e^{z_k}}$$

zk is the softmax input vector in the above equation. For each malware sample x, a pvv (probability value vector) is generated: $pvv(x) = (P(y = 1|x), P(y = 2|x), \ldots, P(y = 63|x))$. pk represents the likelihood of a malware belonging to a certain malware class k. The trustworthiness of the classification result may be easily assessed by comparing the maximum $max(P(y=k/x))$ of pk with the probability threshold value.

III. Selection of variable n-gram features

For the D-Classifier, only the less common and distinct features of n-grams are filtered, which are obtained using the Document Frequency Inversed Document Frequency (DF.IDF) method, where DF selects features that are prevalent in other families and IDF selects features that are rare in other families. DF and IDF are calculated using the formulas:

$$DF(i, j, k) = \frac{\{j : i \in d(k)\}}{D(k)}$$

$$IDF(i, j, k) = log_2\left(\frac{D - D(k)}{\{j : i \in d \cap j : i \notin d(k)\} + 1}\right)$$

The dataset's total number of API call sequences is denoted by D(k), whereas (i, j, k) denotes the jth API call sequence of the ith n-gram in family k. The significance of each n-gram feature can be assessed using the product of DF with IDF.

$$DF \cdot IDF(i,j,k) = DF(i,j,k) \times IDF(i,j,k)$$

The number of times the n-gram feature i is repeated in the API call sequence j can be calculated using the TF (Term Frequency)

IV. Machine Learning Algorithms' Selection

D-classifier was utilized to obtain from the ML algorithms by utilizing the n-grams feature vectors with existing and well-known tags. This was used to detect the malware classes using unlabeled n-grams only. Five different ML algorithms namely: KNN, Naïve Baye Adaboost, SVM and RF have been taken into consideration for the same. These ML algorithms are then validated against their performance and the best is chosen for the D-Classifier.

- The SVM algorithm determines the classification amongst the others with respect to different boundaries of the classifications. Its core model is basically aimed to maximize the separation between items of the training set namely: the (+) and (-) items to find the optimal Seperating Hyperplane.
- Training samples for each tree using Random Sampling is generated by the RF with Replacement and picks various attributes at random across the nodes of the tree. The number of Decision tress can be modified to improve the classifier so that at each node, more features can be examined and RF can produce a decision tree with maximum depth. A voting is carried to combine the results of weak classifiers to form a stronger one.
- Based on the Classifier's error rate Adaboost algorithm aggregates the weighted prediction outputs and it initially starts with the weak classifiers. It employs repeated training on each classifier aimed to modify the distribution of probability.
- The Bayesian algorithm guarantees a greater and an accurate probability of an unknown malware belonging to a particular class using the independent features strengthening the NB.
- KNN is a machine learning approach that is both supervised and lazy. Based on the K nearest and frequent neighbors in the training set, the data points are classified under this approach.

III. RESULTS AND DISCUSSION

Malscore's categorization quality was evaluated by making use of Recall, Accuracy, Precision and F1score. The below are the four consideration for measuring the metrics:

- Precision = TP / TP + FP
- Recall = TP / TP + FN
- F1 score = 2 × Recall × Precision /Precision + Recall
- Accuracy = samples classified right / total samples

A stabilized accuracy was noticed only reaching about 20 features. Although multiple features were taken into consideration, the best accuracy was never more than 80%. Accuracy ranging from 41% to 79% was noticed for n-grams (n ranging between 2 and 4)-classifiers. Several ML algorithms were made use to compare the categorization performance of the different n gram classifiers on the datasets chosen. An accuracy of about 97% was achieved while classifying various classes of malwares in phase 2, which is a remarkable outperformance compared to all other techniques in all different aspects such as F1-score, recall and precision. 0.14% higher than classifier D was noticed against the Malscore's classification.

A steady accurate malscore was noted by keeping the PT ranging between 0.79 and 0.89. On increasing the samples from 17 thousand from 8 thousand, there was not a considerable rise in the accuracy. PT with value 0.94 was fixed as a system parameter upon malware classification and with this value the accuracy reached about 98.8%, with a significant reduction in the amount of samples in phase 2 as well.

The flaws of the static analysis was compensated by employing dynamic analysis, which is then utilized for classifying and detecting different samples. More than 99% of the time is accounted or consumed in language learning is spent on grayscale generations and n-gram feature extractions. The classifiers belonging to dynamic analysis take more time than compared the ones belonging to the static analysis.

IV. CRITICAL ANALYSIS OF RESULTS OBTAINED

This paper clearly examined the impact of variable n-grams (n ranging between 2 to 4) on classification outcomes against the fixed n-grams. Similarly, the impact of the five ML techniques on classification outcomes was clearly documented. Keeping the features below 12, the accuracy of the 3-gram classifier outperformed the others, while having a number greater than 12 produced greater accuracies for the n-gram classifiers. A consistent 97.5% accuracy which is nearly 2.3% to 5.2% greater than the other accuracies was noticed when the number of features was fixed to 16 with the n-gram classifiers. By aggregating the results obtained from the

first and second phase, the PT used both trained S –based form of the CNN algorithm as well as the learned classifier D –based RF method. Along with minimizing the time cost of Malscore, a proper PT also enhanced the classification accuracy considerably.

In terms of cost-accuracy tradeoff, Malscore's classification performance outperformed static analysis, dynamic analysis, and ensemble learning. As malware technology matures, more and more issues, such as crossing execution pathways, idea drift, and disguised system calls, will pose challenges to detection systems. In depth research is required with respect to Malscore and it has a huge scope of improvement to deal with multiple detecting issues.
Of about 0.95s pre-processing and 0.02ms testing time more than static analysis and 2.25s pre-processing and 0.05s testing time less than ensemble learning was noticed against the Malscores' average scores. Along with producing lower time cost, Malscore could also handle packaged malware. Owing to the higher cost of the dynamic analysis, the average time cost of Malscore is a greater value. Of about 75% of the samples had an execution time of 185ms when compared to 1.6s when it was run on the server.

A better classification performance and API call sequencing was observed with CNN and the RF algorithms when compared to other ML or DL algorithms while categorizing the grayscale images too. RF and CNN algorithms provided better results for Malscore while comparing the API call sequences and the black and white images compared to the other characteristics. It was noticed that in both feature selection and classification, the Malscore could provide with improved and enhanced classification performance.

V. ADVANTAGES OF THE PROPOSED METHODOLOGY

- With increasing n-gram features extracted, more detailed semantic structure of malware is obtained.
- The resilience of the packed malware is considerably boosted, resulting in improved classification accuracy by Malscore.
- All of the malware's undesirable characteristics are filtered using probability scoring, yielding only trustworthy classification results for S-classifier.
- It was noticed that in both feature selection and classification, the Malscore could provide with improved and enhanced classification performance.
- An accuracy of about 97% was achieved while classifying various classes of malwares in phase 2, which is a remarkable outperformance compared to all other techniques in all different aspects such as F1-score, recall and precision.
- This technique greatly decreases the execution time of the dynamic analysis as well as the Malscore detection cost. Using the grayscale pictures as static characteristics along with CNN and SPP layers to classify the various kinds of malware. This can aid in minimizing the loss with respect to the malware information by picture pre-processing. Malscore can thus, save as much as the semantic information of the virus by allowing only the feature association.

VI. LIMITATIONS

- Malscore uses dynamic analysis as an alternative in case of a packed malware instead of using it as a main detection method owing to its high cost. This has resulted in high accuracy and producing results with minimal costs but in the long run, the performance of such methods will degrade due to concept shift. 1
- Malscore does not account for concept drift, which doesn't offer any incremental learning techniques to strengthen Malscore's re-learning capacity.
- This method completely relies on the features extracted from the static and dynamic analysis and with increasing complexity of unwrapping such features in malwares that resist both types of analysis resulting with false analysis.
- The greyscale conversion procedure is time-consuming for pre-processing and reduces the association between picture blocks with useful information contained in them
- There exists a constraint on the training set which is quite limited and can generate sparse data which causes a breach among large sets. Larger value of n produces huge parameter spaces, resulting in dimension disaster and impossibility of implementation.

VI. ADOPTABLE ALTERNATE METHODOLOGIES

It is easily noticeable that the results obtained from the classification in Phase 2 are to be given more importance than those of the results obtained in Phase 1, irrespective of their equality as it's proved that the classification results obtained from Phase 1 are unreliable, owing to its low credibility. Rather than just adopting an entropy-based detection approach, we can employ a probability grading in Malscore directly, evaluating to choose whether or not execute dynamic analysis, which can save time and efforts needed for determining the classifications as the entropy-based technique can only identify the malware's packed nature and is unable to identify if malware employs dead code implantation, rearrangement of code, subroutine reordering, or various methods to escape static analysis.
Certain features like less visible static features and fewer packed/encryption units that really don't influence static feature distribution need not be taken into consideration hence relieving or filtering out the important features only, hence can speed up the process of classification.

Instead of relying on dynamic analysis for generating more precise results for classification and rather than splitting the samples into static and dynamic test samples, malscore may evaluate the dependability of malware in static analysis using probability scoring and determine whether to utilize dynamic analysis or not.

Concept drift being a cumbersome yet very useful feature of machine learning techniques, its effect can be reduced so that it has minimal influence by making use of the MaMaDoid and incremental learning techniques. This will be a relatively new concept with respect to this context but additional research and in depth implementation of this can be used to assess the influence the idea drift on Malscore. The stability of dynamic analysis should be strengthened not simply to augment static analysis, but also to play a significant role in malscore identification.

VII. CONCLUSION

Probability scoring along with Machine Learning techniques have been combined to produce a Malscore method which is used for malware classification. Malscore utilizes CNN with SPP to scrutinize the grayscale images which constitute the first phase's static features whereas different ML models along with n-gram are used to form an emphasis on the API call sequence, constituting the dynamic features in Phase 2. Came up with a scoring method based on probability to determine the validity of the categorization findings, while the second Phase continued to investigate lesser credibility malware. Out of 63 Malware families, an over about nearly 2 lakh experiments were carried out which showed that the Malscore has produced, with respect to the malware classification, an accuracy about 98.8%. A reduction of about 59.58 percent and 61.70 percent with respect to Malscore's pre-processing and test time was noticed contrasting the ensemble learning in testing. In an OS with 16GB RAM and 3.6GB CPU, the malscore could detect a packed malware in 0.04ms and unpacked in 0.001ms, indicating that it required very less time to produce the results. The resilience of the packed malware is considerably boosted, resulting in improved classification accuracy by Malscore.

REFERENCES

[1] Xue, D., Li, J., Lv, T., Wu, W., & Wang, J. (2019). Malware classification using probability scoring and machine learning. *IEEE Access*, *7*, 91641-91656.

chkrithika02@gmail.com.docx

ORIGINALITY REPORT

3%	2%	3%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Di Xue, Jingmei Li, Tu Lv, Weifei Wu, Jiaxiang Wang. "Malware Classification Using Probability Scoring and Machine Learning", IEEE Access, 2019 Publication	1%
2	V. Anandhi, P. Vinod, Varun G. Menon. "Malware visualization and detection using DenseNets", Personal and Ubiquitous Computing, 2021 Publication	1%
3	ijisrt.com Internet Source	<1%
4	Rajeshkannan Regunathan, Aramudhan Murugaiyan, K. Lavanya. "Neural Based QoS aware Mobile Cloud Service and Its Application to Preeminent Service Selection using Back Propagation", Procedia Computer Science, 2018 Publication	<1%
5	"Malware Analysis Using Artificial Intelligence and Deep Learning", Springer Science and	<1%

Business Media LLC, 2021
Publication

6	ijlemr.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		