Malicious Code Detection based on Image Processing Using Deep Learning.

Conference Paper · November 2018

DOI: 10.1145/3194452.3194459

Rajesh Kumar
University of Electronic Science and Technology of China
17 PUBLICATIONS **0** CITATIONS

Malicious Code Detection based on Image Processing Using Deep Learning

Rajesh Kumar
University of Electronic Science and Technology of China
0086-15520777096
rajakumarlohano@gmail.com

Zhang Xiaosong
University of Electronic Science and Technology of China
008618982067786
s_x_zhang@163.com

Riaz Ullah Khan
University of Electronic Science and Technology of China
0086-15520763595
rerukhan@gmail.com

Ijaz Ahad
University of Electronic Science and Technology of China
ijazahad1@gmail.com

Jay Kumar
Quaid-e-Azam University Islamabad, Pakistan
00923332836704
jay_tharwani1992@yahoo.com

ABSTRACT

In this study, we have used the Image Similarity technique to detect the unknown or new type of malware using CNN approach. CNN was investigated and tested with three types of datasets i.e. one from Vision Research Lab, which contains 9458 gray-scale images that have been extracted from the same number of malware samples that come from 25 different malware families, and second was benign dataset which contained 3000 different kinds of benign software. Benign dataset and dataset vision research lab were initially executable files which were converted in to binary code and then converted in to image files. We obtained a testing accuracy of 98% on Vision Research dataset.

CCS Concepts

• Security and privacy → Malware and its mitigation

Keywords

Malware Detection, Convolutional Neural Network, Malware Classification, Deep Learning

1. INTRODUCTION

1.1 Background

One of the major challenges in the realm of security threats is malicious software which is also referred as malware. The main focus of malware is, to gather the personal information without the attention of users and to disturb the computer operations which makes problems for users. There are many kinds of malware i.e. Virus, Worm, Trojan-horse, Rootkit, Backdoor, Spyware, Adware etc [2]-[4]. Annual reports from antivirus companies show that thousands of new malware are created every single day. This new malware become more sophisticated that they could no longer be detected by the traditional detection techniques such as signature-based detection, heuristic detection or behavior- based detection.

Signature-based detection searches for specified bytes sequences into an object so that it can identify exceptionally a particular type of a malware. Its drawback is that it cannot detect zero-day or new malware since these malware signatures are not supposed to be listed into the signature database [5]. Heuristic-based detection was developed to basically overcome the limitation of the signature detection technique, in the way that it scans the system's behavior in order to identify the activities which seems to be not normal, instead of searching for the malware signature. Heuristic-based detection method can be applied to newly created malware whose signature has not yet been known. The limitation of this technique is that it affects the system's performance and requires more space. Behavior-based detection technique is more about the behavior of the program when it is executing. If a program executes normally, then it is marked as benign, otherwise it is marked as a malware. By analyzing this definition of the behavior-based detection, we can directly conclude that the drawback of this technique is the production of many false positives and false negatives, considering the fact that a benign program can crashed and be marked as a virus or virus can execute as if it was a normal program and simply be marked as benign.

1.2 Motivations

Malware is growing in the huge volume every day, we used image processing technique in order to improve accuracy and performance. Image processing technique analyzes malware binaries as gray-scale images. The previous research [10] proposed a new method for visualization to classify malware using image processing technique. Some of mature image processing techniques are widely used for object recognition

e.g. taobao is popular shopping website in china which find's the product using image recognition technique. This method performs high accuracy in practice. In this study, we converted binary code to images for recognizing malware which preserve the similarities variant images. We observed that the image recognition method is helpful to achieve better performance and accuracy.

1.3 Our approach

Malware classified in different families has multiple characteristics or features. Many authors used machine learning models such as Regression, K-nearest-neighbor, Random Forest etc. Main disadvantage of using machine learning is, features extraction is manual. Gavrilut et al. [8] gave an

ICCAI 2018, March 12–14, 2018, Chengdu, China
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6419-5/18/03$15.00
https://doi.org/10.1145/3194452.3194459

overview of different machine learning techniques that were previously proposed for malware detection. Unlike Machine Learning, Deep learning skips the manual steps of extract- ing features. For instance, we can feed directly images and videos to the deep learning algorithm, which can predict the object. In this way deep learning model is more intelligent rather than machine learning model. We used convolutional neural networks because it is reliable and it can be applied to the entire image at a time and then we can assume they are best to use for feature extraction. Recently Constitutional Neural Networks [6] is the new approach to detect malware by using image based similarity technique. Its automated image comparison helps analysts to visually identify com- mon code portions or specific instruction blocks within a sample. In this work we used three different datasets and compared the accuracy. Secondly we used different tech- niques to prepare datasets for training and testing purposes. we trained and tested the CNN model for better under- standing of the malware behavior. Overall, we show that our proposed approach constitutes a valuable asset in the fight against malware. Figure 1 gives a brief overview to the different stages i.e. from data preparation to malware detection.

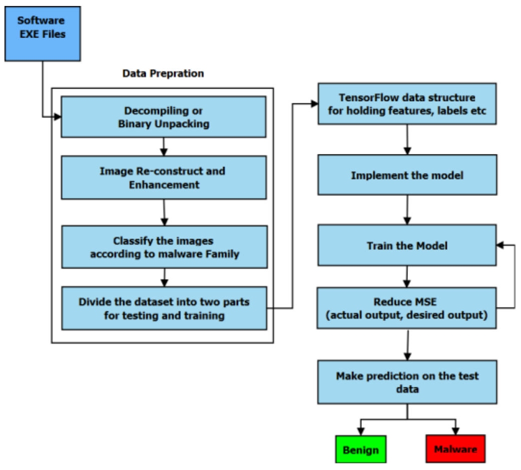

Software EXE Files
Data Prepration
Decompiling or Binary Unpacking
Image Re-construct and Enhancement
Classify the images according to malware Family
Divide the dataset into two parts for testing and training
TensorFlow data structure for holding features, labels etc
Implement the model
Train the Model
Reduce MSE (actual output, desired output)
Make prediction on the test data
Benign
Malware

Figure 1. Workflow diagram; from data preparation to malware detection.

1.4 Contributions

The main contributions of the paper are summarized as follows:

We used the Convolutional Neural Networks for detection of malware, based on image similarity which is further described in Section 3

We successfully analyzed and detected unknown or new type of malware

We achieved better results in terms of training / testing accuracy and speed of detection which is further described in section 3

We achieved 98% of accuracy on Vision Research Lab's Dataset.

1.5 Structure of paper

The Section 1 discusses the background, motivation, ap- proach used in this study and main contributions of this work. Section 2 gives a brief overview to the methodology that how to convert executable files in to images and also setup the python libraries. Section 3 proposes a malware detection technique, discusses optimized CNN model, de- scribes the implementation and experiment results in terms of accuracy. Finally, Section 4 concludes the paper.

2. DATA PREPARATION AND ENVIRON- MENT SETUP

This section is divided into two parts. The first part is, to collect malware and benign datasets from different sources and second part describes the techniques of preparation of the dataset. In second part we used a technique to prepare dataset which is described in Section 2.2.

2.1 Collection of Dataset

We have collected three datasets from different sources. Two of them are malicious datasets from two different sources i.e. from Vision Research Lab and from Microsoft Malware Clas- sification Challenge. We also collected 3000 benign file from different sources. All three datasets are discussed briefly in the following discussions.

2.1.1 Vision Research Lab Dataset

First dataset is collected from Vision Research Lab and this dataset is called Malimg Dataset [10]. The dataset comprises 25 malware families while the number of variants is different in each family. Dataset is shown in Table 1 along with class name, family name and number of samples.

Table 1. Malimg Dataset from Vision Research Lab Dataset

No	Class	Family Name	No of Samples
1	Worm	Allaple.L	1591
2	Worm	Allaple.A	2949
3	Worm	Yuner.A	800
4	PWS	Lolyda.A A 1	231
5	PWS	Lolyda.A A 2	184
6	PWS	Lolyda.A A 3	123
7	Trojan	C2Lop.P	146
8	Trojan	C2Lop.gen !G	200
9	Dialer	Instantacc ess	431
10	Trojan Downloader	Swizzor.ge n!l	132
11	Trojan Downloader	Swizzor.ge n!E	128
12	Worm	VB.AT	408
13	Rogue	Fakerean	381
14	Trojan	Aluron.ge n!J	198
15	Trojan	Malex.gen !J	136
16	PWS	Lolyda.AT	159
17	Dialer	Adialer.C	125
18	Trojan Downloader	Wintrim.B X	97
19	Dialer	Dialplatfor m.B	177
20	Trojan Downloader	Dontovo.A	162
21	Trojan Downloader	Obfuscator .AD	142
22	Backdoor	Agent.FYI	116
23	Worm:AutoIT	Autorun.K	106
24	Backdoor	Rbot!gen	158
25	Trojan	Trojan	Trojan

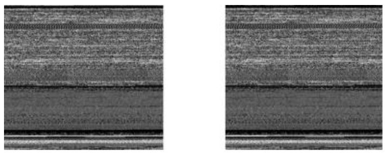


Figure 2. Images extracted from malware.

Malimg Dataset consists 9,458 gray-scale images of 25 malware families. Ratio of 90-10 was used for model performance evaluation. 90% of the total data was used for train- ing and 10% was used for testing. The real malware binaries of this dataset was available in [1],

As Gavrilut et al. [7] explained that a binary code of a given malware can be read as a vector of 8 bits un-signed integers and organized into 2-dimensional array which can be visualized as a gray-scale image in the range of [0,255], where 0 represent black and 255 for white. The size of the image is different depending on their families. We observed in Figure 2, that images which belong to the same family are looking very similar to one another.

2.1.2 *Benign files*

We collect 3000 benign files from different sources.

2.2 Data Preparation Techniques

This paper proposes the following two techniques to process the data.

2.2.1 *Direct Convert Assembly to Image*

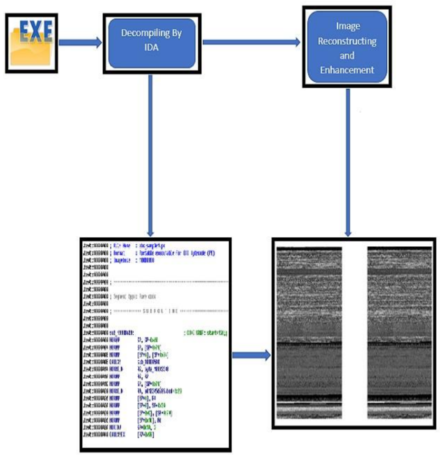


Figure 3. Overview architecture of Preparation Dataset.

Decompiling: we used the following algorithm to de- compile the exe file to binary and assembly.

Convert Assembly Code to Image: The process of con- verting assembly to images is shown in Figure 3.

2.3 Environment Setup

Centos system with 64bit with 8 GB RAM environment is used to perform tests. We used Python programming lan- guage to perform the experiments. Python packages and libraries such as Tensor Flow, Docker Server, Anaconda are used which helped to detect the malware. The Tensor Flow Library is used for training the model which uses the con- volutional natural network (CNN).

3. IMPLEMENTATION AND PERFORMANCE EVALUATION OF THE PROPOSED MODEL

3.1 Proposed Model

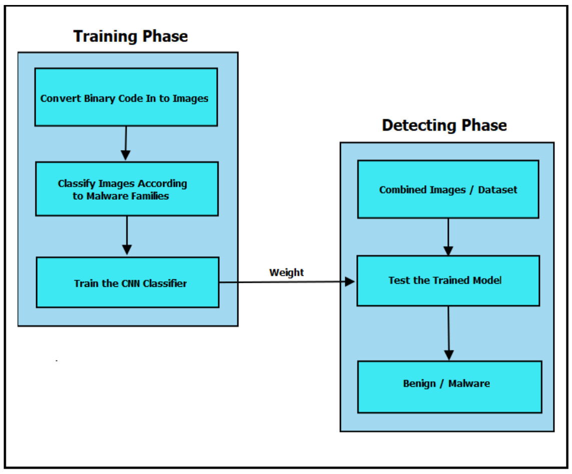
In this design we divided model in two phases i) Training phase and ii) Detection phase. For the training and the de- tection of malware we used CNN model, as shown in Figure

4. We prepare the dataset using different techniques shown in data preparation section. The output of the data prepa- ration section is “image files”. Images have binary labels

i.e. either benign or malware. we used supervised learn- ing model in which the features are extracted automatically. The detection phase is shown in Figure 4. The same exe file convert in image and trained classifier detect the malicious code.

3.2 Training Convolutional Neural Networks Structure

We have used convolutional neural networks because it is re- liable and it can be applied to the entire image at a time and then we can assume they are best to use for feature extrac- tion. convolutional neural network is a feed-forward neural network where the connectivity pattern between neurons is inspired by the structure of an animal visual cortex and that has proven great value in the analysis of visual imagery.



Training Phase
Convert Binary Code In to Images
Classify Images According to Malware Families
Train the CNN Classifier
Weight
Detecting Phase
Combined Images / Dataset
Test the Trained Model
Benign / Malware

Figure 4. Architecture of proposed Method.

All the images are reshaped into a size of 128 X 128 pixels. Since all the models of deep learning accept data in form of numbers, we have used image library from PIL package of Python to generate vectors of images and further processing are done on these vectors.

We have then designed a three layers deep Constitutional Neural Network for the detection task, which has the fol- lowing properties: On the Rectified Linear Units (ReLU)

layers, we first apply a two dimensional convolutional layer and after each layer, we applied a nonlinear later also known as activation layer. In convolutional layer, we have opera- tions like element-wise multiplication and summations. The ReLU adds non-linearity to the system. We have used the ReLU instead of non-linearity function because it is faster than tanh or sigmoid and help in vanishing gradient problem which arises in lower layers of the network.

We have also used max pooling layer instead of other layers. It takes a filter and a stride of the same length then applies it to the input volume and outputs the maximum number in sub region that the filter involves around. The intuition behind this was the fact that our malware image is a gray scale and the layers like average max pooling may not help much because there are a lot of dark space in the image and they don't contribute much in the model.

The output that we want is a single class in which the given malware belongs to. After applying all the layers, we have a three-dimensional vector of arrays. To convert this vector into a class probability, we convert these vectors into a sin- gle layer of one dimension, known as fully connected layer. Down-sampling all the vectors to a one-dimensional vector may lead to loss of data. For that reason, we have used two fully connected layers.

Cross entropy loss function that is commonly used for multi class classification was used for this work as well as Adam optimizer for optimization task. The overall architecture of the model is show in Figure 5.

Initially, all the images were of different sizes and had to be converted into 128 X 128 pixels before they are used as input to the model.

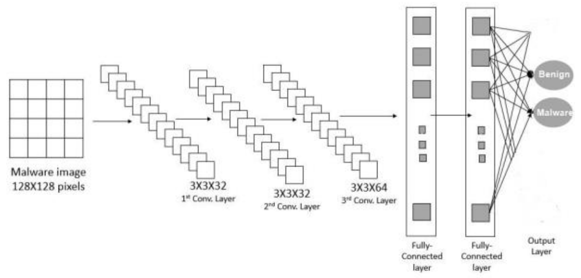


Figure 5. Overview architecture of CNN proposed Method.

3.3 Implementation

The following tools and techniques are required for experi- mental setup. For preparation of dataset we used method shown in section 2, tools and algorithm which were used to arrange the dataset for achieving better results is also written in data preparation section. For detecting malware we use supervised learning to train the model. CNN al- gorithm was used to train and test the model. Figure 5 shows 3 hidden layers, each layer has own parameters (e.g., filter − size1 = 3, numf ilters = 32, etc), In this algo- rithm we used AdamOptimizer and the learning rate of the optimizer is le-4. The size of for all hidden layers for the

convolutional neural network are 3*3*32, 3*3*32, 3*3*64, respectively. For the validation, system was trained with 20 epochs.

3.4 Experiment Results

We took two datasets in considerations which are discussed in Section 2. One of two datasets consists of malicious code and one dataset is a benign file. For the first test, we com- bined the benign dataset with Malimg dataset and used the combined dataset to obtain the accuracy in terms of mal- ware code detection. The result obtained in the experiment shows an accuracy of 98% for the Dataset of Vision Research Lab shown in Figure 6 and 7.

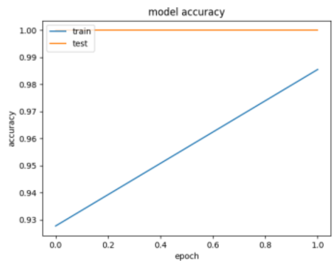


Figure 6. Accuracy of CNN model.

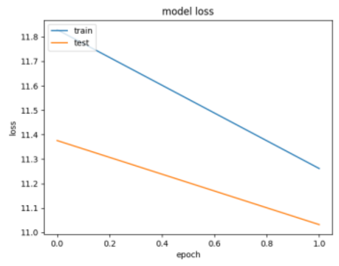


Figure 7. Loss of CNN model.

4. CONCLUSION

Being able to visualize the malicious code as a gray-scale image has been a great achievement. Many researchers have been using this technique for the task of malware classifica- tion and detection. However, other works have shown that this technique can be easily vulnerable to adversarial at-tacks and produce erroneous results. [9], [11], [12] have shown in their works how a small change in the image could lead to miss-classification of images. The biggest challenge is to find an efficient way to overcome the vulnerability of Neu- ral Networks. This could be achieved by carefully analyzing malware binaries.

5. REFERENCES

[1]. Vision reseach lab malimg dataset http://old.vision.ece.ucsb.edu/spam/malimg.shtml.

[2]. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

[3]. Adebayo, O.S. and Aziz, N.A., 2015. Static Code Analysis of Permission-based Features for Android Malware Classification Using Apriori Algorithm with Particle Swarm Optimization. *Journal of Information Assurance & Security*, 10(4).

[4]. Alme, C., Mcafee, Inc., 2012. Systems, apparatus, and methods for detecting malware. U.S. Patent 8,312,546.

[5]. Bennasar, H., Bendahmane, A. and Essaaidi, M., 2017, April. An Overview of the State-of-the-Art of Cloud Computing Cyber-Security. In *International Conference on Codes, Cryptology, and Information Security* (pp. 56-67). Springer, Cham.

[6]. Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., Wang, Z., Huang, Y., Wang, L., Huang, C., Xu, W. and Ramanan, D., 2015. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2956-2964).

[7]. Gavriluţ, D., Cimpoeşu, M., Anton, D. and Ciortuz, L., 2009, October. Malware detection using machine learning. In *Computer Science and Information Technology*, 2009. IMCSIT'09. International Multiconference on (pp. 735-741). IEEE.

[8]. Gavriluţ, D., Cimpoeşu, M., Anton, D. and Ciortuz, L., 2009, October. Malware detection using machine learning. In Computer Science and Information Technology, 2009. IMCSIT'09. International Multiconference on (pp. 735-741). IEEE.

[9]. Goodfellow, I.J., Shlens, J. and Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[10]. Nataraj, L., Yegneswaran, V., Porras, P. and Zhang, J., 2011, October. A comparative assessment of malware classification using binary texture analysis and dynamic analysis. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence (pp. 21-30). ACM.

[11]. Nguyen, A., Yosinski, J. and Clune, J., 2015. Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images-Nguyen_Deep_Neural_Networks_2015_CVPR.

[12]. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B. and Swami, A., 2017, April. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (pp. 506-519). ACM.