ISSN:0975-9646

Ransomware Classifier using Extreme Gradient Boosting

Niranjan Agnihotri

Computer Science Graduate, (2017)
Walchand College of Engineering, Sangli, India

Abstract— **Ransomwares are notorious threats that have become rampant in the cyber world, the peculiarity about them is that they encrypt user's critical data using strong encryption algorithms. The encrypted data can only be recovered by paying a ransom in bitcoins to the attacker. A traditional signature based detection approach is slow and time consuming and cannot handle the zero-day attacks. Therefore, in this paper, an attempt is made to build a pro-active Machine Learning based classifier which is trained to detect the ransomwares based on the static attributes of the PE files.**

Keywords— Ransomware, machine learning, gradient boosting, Portable Executable file, classification, cyber security

I. Introduction

Manual malware analysis is a multi-staged and a time-consuming process. With the ever-expanding threat landscape, it becomes difficult to handle the increasing volume of threats while keenly adhering to the manual malware analysis techniques. Ransomware, as a threat, stands out of all other types of malwares because it hijacks user's critical data by encrypting it using the modern cryptographic algorithms. Ransomware, has proved to be one of the deadly threats because it's malicious changes cannot be undone without paying the ransom to the attacker. Therefore, with the advent of several machine learning techniques, it becomes obvious and simultaneously necessary to utilize the huge repository of data and samples to figure out some method to train a machine learning model to detect ransomwares quickly.

It is evident that most of the malwares are authored to target the windows platform, where most of the software is distributed in the famous Portable Executable [1] (PE) format. This is a structured format, therefore, it's structure can be exploited to fetch several attributes which can be helpful to train a machine learning model. Several attempts [2] have been made to extract various attributes from the PE file format. The attributes extracted are mostly in the numerical format or categorical format, which makes the work of training the classifier easy.

In this attempt of building a machine learning based ransomware classifier, we collect a pool of clean samples and a pool of ransomware samples from the internet. PE file attributes are collected from these samples with the help of an attribute extractor [3] script and a dataset is built. This dataset is then split into training and test sets. A supervised machine learning model is then trained on the training set and is evaluated on the test set.

Nowadays, machine learning has been used widely for threat detection. Various models are built using supervised learning. Researchers have developed several approaches in using machine learning for malware detection may it be by training a neural network by feeding [4] it a whole portable executable file or as done by researchers at Invensia Labs by training a deep neural network for Malware detection using two-dimensional binary program features [5]. The rapid use of machine learning techniques in malware detection and the rampant problem of ransomwares motivates to work upon building a classifier which is specifically trained to identify ransomwares.

There could be several approaches to build a classifier to detect ransomwares. This work restricts itself at building a machine learning based classifier which is trained upon static attributes of portable executable files.

II. Challenges in Manual Analysis and signature based Traditional approach

Several challenges are involved in manually analysing the samples. Some of them are as follows -

A. The Volume of emerging threats

With the ever-increasing threat space, it becomes difficult to handle the volume at which new threats are created and distributed. Not just new threats, but malware authors also release the threats into different variants, who show similar traits but have slight differences. With the increasing volume of malwares, there are several new ransomwares like Petya, WannCry, TeslaCrypt etc. which pose a potent threat to the user's critical data.

B. The Time required to manually analyse a sample

It requires a lot of manual effort and time to analyse a sample and classify it as a specific kind of threat. Time delay in the case of ransomware processing implies that all the infected users would lose their valuable data.

C. Zero Day Attacks

Manual Analysis generally entails a detection based on signature. Therefore, initially, some people always get infected and suffer from a new threat, until the threat is manually analysed and an anti-virus signature is available through a definition update. Therefore, a traditional signature based detection is ineffective against zero-day attacks. Therefore, users who face the ransomware attacks from new threats have no protection from signature based technologies.

D. Signature Evasions

The antivirus signatures are written for specific threats or families of threats. Therefore, it becomes easy for the malware author to perform trivial tweaks in his malicious code and redistribute the malware until a new signature against the new version of the malware is released. Therefore, traditional anti-virus signatures can be easily evaded. Such evasions in the case of ransomwares would be severely detrimental for the end-users and their critical data.

Therefore, a machine learning model trained to detect ransomwares would be very useful for protecting the user's data from encryption.

III. Proposed System

To utilize the information from the static attributes of a large pool of portable executable files, a machine learning classifier is to be trained.

Firstly, an attribute extractor [3] script will extract various numerical values from the executable files for the corresponding attributes. These values would be then fed as an input to a trained classifier which will predict if the given sample is a ransomware or a non-ransomware.

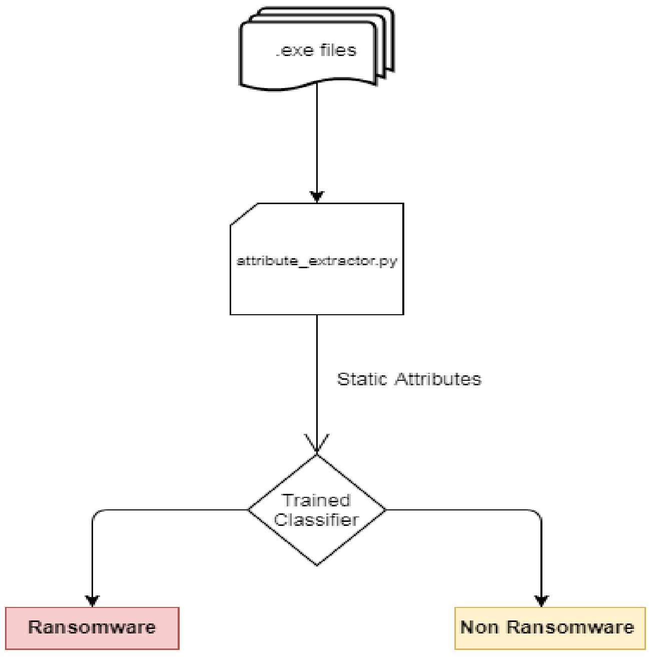

Fig. 1 The workflow for Ransomware Classifier.

The workflow diagram in Fig. 1 adumbrates the overall workflow from extracting the static attributes from samples of portable executable files to their classification.

IV. Methodology

There are several steps in building the model which are enlisted as follows.

A. *Data Collection and Attribute Extraction*

Firstly, samples were collected form the various sources [6] [7] on the internet in the executable format. The samples that were used for training the model were taken from two different classes. The first class contained non-ransomware samples and the second class contained several families of ransomwares. These samples were first maintained in two different folders '/ransomware' and '/non-ransomware'. With the help of open-source attribute extractor codes [3] available on the git-hub, static attributes were computed on these samples whose values were recorded in the csv file.

B. *Making the data ready for machine learning*

In this step, all the rows having NANs were removed. A pairwise Euclidean distance was found among all the rows in the dataset. One of the rows among every pair whose Euclidean distance lied below 0.00005 were removed from the dataset. This threshold was decided by manually checking the rows with a very less Euclidean distance and by understanding that there was a very little change in the values of those rows. At the same time all the exact duplicates were also removed from the dataset, thus maintain all the unique rows.

The dataset was then split into train and test sets. The data set was scaled to expediate the process of training the model. Finally, the samples from the 'ransomware' family were labelled as '1' and the samples from the 'non-ransomware' family were labelled as '0'

C. *Feature selection and Ranking*

The attribute extractors extracted 205 different attributes from the samples. Therefore, to reduce the dimensionality of the dataset and to study which attributes contribute strongly towards predictions feature selection process was done using some techniques enlisted below.

a) At first, Karl Pearson's correlation coefficient was found out among all the possible pairs of attributes. Around 20 pairs of highly positively corelated attributes were found among which 10 attributes were eliminated.
b) Attributes with very low or zero variance attributes were also found in the dataset. Seven of such attributes were removed as they had no information for predicting ransomwares.
c) Feature Selection Using Recursive Feature Elimination – In this technique, several random forests were built using the training set, each time selecting a different subset of the 188 remaining features. The set of attributes corresponding to the forest with highest accuracy were considered the most informative attributes. With this technique it was found that out of 188 features, 135 features were helpful in predicting the outcome. These 135 features had a good variance and were not corelated with each other conforming that they had some unique information.
d) Feature Selection using XGBoost [9] Algorithm –
A simple model was built on the train set using the XGBoost [9] algorithm. This is one of the most sophisticated algorithms in machine learning and has a feature of attribute ranking. According to this algorithm 139 attributes were most informative in predicting ransomwares. All the 135 important attributes figured out by Recursive Feature Elimination were present in the 139 attributes suggested by XGBoost [9] feature ranking algorithm.

Finally, the 139 features suggested by the XGBoost model were finalized for training the machine learning model.

D. *Model Selection and Building*

Three tree based models namely: CART (Classification and Regression Trees), Random Forests [11] and XGBoost. [9] were tried. But as XGBoost [9] outperformed the other two, it was selected for the classification task. XGBoost [9] is an ensemble based algorithm which has been helping a lot of data scientists to win Kaggle competitions. It is an extremely fast algorithm and supports parallel building of the forest. It works on the principle of Gradient Boosting.

At first a basic XGBoost [9] forest with its default parameters was built on the training set. It gave an accuracy of 83% with a False Positive rate of 4%. After building this preliminary model, parameter tuning [10] was done in order properly utilize different features of the model. In the process of parameter tuning [10] around 10 different parameters, viz. the learning rate, max depth etc. were tuned.

V. RESULTS

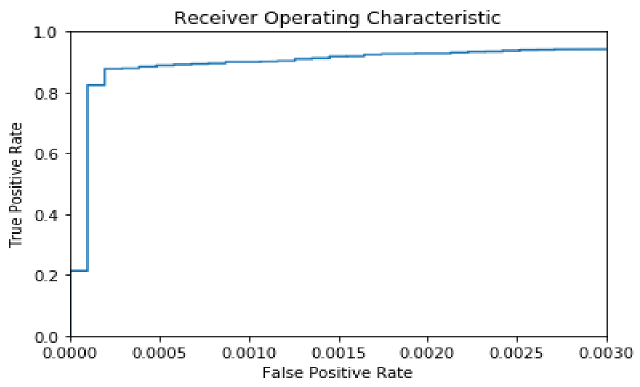

Fig. 2 The graph above depicts the ROC curve for the completely tuned final XGBoost Model.

Fig. 1 shows Receiver Operating Characteristic curve for the final model of the classifier. It is evident from the curve that a 0.3% False Positive rate is achieved. Some other results on the test set with a completely tuned XGBoost [9] model are as follows.

a. Test set size – 19,901 rows with 139 attributes as columns.
b. Final Model Accuracy – 89.80%
c. Final Model False Positive Rate – 0.3%
d. Final Model True Positive Rate – 88.08%

VI. FUTURE SCOPE

This is a small ensemble based classifier powered by a powerful algorithm like XGBoost [9]. But, with the advent of deep learning it was tempting to use neural networks for the above classification problem. Due to lack of computational resources this work had to be kept limited to the use of ensemble technique, whose results are good. But, there seems to be a scope of improvement for using more convolute models like neural networks to get yet more promising results.

The limited size of data set was also a big hurdle in training the classifier. Therefore, it would be highly interesting to witness the results obtained by training the classifier on a bigger dataset. Due to limited availability of ransomware samples, the dataset was skewed therefore techniques to handle skewed dataset have been left to the future scope.

VII. CONCLUSIONS

A machine learning based classifier trained for classifying ransomwares would be of a great use for cyber security organizations. This would cut-short a lot of time needed to analyze samples manually. This classifier would also help in proactively detecting the threats that have never been seen before thus mitigating zero-day attacks.

Malware authors and anti-malware experts being always in a cat and a mouse game, there could be techniques [12] to evade machine learning models too, as shown by the researchers at Endgame. But as every technology has its own limitations, overall, the ransomware classifier can be very helpful in coping with a notorious threat as ransomware.

ACKNOWLEDGMENT

I would like to thank malshare.com for being such a great repository of threats without which this research would have not got a diverse variety of samples based on which the dataset was built. I would also like to thank the git-hub repository creators of ‘the Zoo’ for providing various families of ransomwares.

I would also like to thank endgame for their open source code for attribute extraction. The tutorials from ‘machine learning mastery’ were highly useful in training and parameter tuning of the model.

REFERENCES

[1] Peering inside the PE by Matt Pietrek
[Online]- https://msdn.microsoft.com/en-us/library/ms809762.aspx

[2] Kartik Raman, “*Selecting features to Classify Malware*”,
[Online] - http://www.covert.io/research-papers/security/Selecting%20Features%20to%20Classify%20Malware.pdf

[3] Your Model is Not Special, from Endgame. Inc. (Attributes extractor)
[Online] - https://github.com/endgameinc/youarespecial

[4] Edward Raff, Jon Barker, Jared Sylvester, Robert Brandon, Bryan Catanzaro, Charles Nicholas., *Malware Detection by Eating a Whole EXE*, [Online]- https://arxiv.org/abs/1710.09435

[5] Joshua Saxe, Konstantin Berlin, “Deep Neural Network Based Malware Detection Using Two-Dimensional Binary Program Features,” *IEEE 2015 10th International Conference on Malicious and Unwanted Software: “Know Your Enemy”*.

[6] Source 1 for Ransomware and Non-Ransomware Samples
[Online] - www.malshare.com.

[7] Source 2 for Ransomware and Non-Ransomware Samples
[Online] - https://github.com/ytisf/theZoo

[8] Json Brown Lee *“Feature Selection For Machine Learning in Python”*. [Online] : http://www.ieee.org/

[9] Tianqi Chen and Carlos Guestrin *“XGBoost: A Scalable Tree Boosting System”* [Online]. https://arxiv.org/pdf/1603.02754.pdf

[10] *Complete Guide to Parameter Tuning in XGBoost (with codes in Python)*
[Online] https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

[11] Leo Breiman “Random Forests”
[Online]-
https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

[12] Hyrum S. Anderson, Anant Kharkar, Bobby Filar, Phil Roth “*Evading Machine Learning Malware Detection*,”
Black Hat US 2017
[Online] - https://www.blackhat.com/docs/us-17/thursday/us-17-Anderson-Bot-Vs-Bot-Evading-Machine-Learning-Malware-Detection-wp.pdf