International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Analyzing Ransomware Attack & Improving Detection Rate

Mohd Sharay Ammar

MTech CSE Department of Computer Science and Engineering C-DAC NOIDA, INDIA

Abstract:
In today's society, data security is a major problem and data breaches, hacking, and other illicit activities are all too prevalent. In today's world, people are increasingly sharing data over the network, whether it's on shopping sites, making purchases using digital payment methods, or downloading data over the internet, for example. Ransomware is a type of sophisticated virus that has exploded in popularity in recent years, causing massive financial losses for a wide range of victims, including businesses, healthcare facilities, and individuals. Modern host-based detection systems require the host to be infected first in order to identify irregularities and malware. The purpose is to explore and analyse various malware detection approaches, as well as to propose a strategy to decrease false positives when separating ransomware attack files from benign (non-malicious) files. The goal of this study is to train a model that can detect any unknown file, that is not in the virustotal database, with high accuracy without any external assistance. Finally, we will cross-check with the Virustotal API for that specific Hash value to see whether it can identify it with a fair Threshold value. The resulting accuracy of 95.62 percent demonstrates the method's efficiency in detecting malware. This paper aims to collect data that will aid in filling research gaps in machine learning-based malware.

Keywords: Hashing, encryption, Machine Learning, Ransomware, API Call Sequencing, Malware analysis.

1. INTRODUCTION

Ransomware is a type of cyber-blackmail carried out by frequent phishing assaults carried out by money-motivated hackers, in which target data is held hostage in exchange for money. Typically, hackers encrypt the victim's data and refuse to disclose the decryption key until the ransom is paid. It prohibits you from accessing your computer's data. This hazardous spyware keeps your files captive, causing massive harm in larger enterprises.

Ransomware is one of the worst nightmares for a company's security since it destroys the core underpinnings of cybersecurity: -

- Confidentiality
- Integrity
- Information Availability.

As a result, it is better to identify ransomware attacks rather than cope with their consequences. Ransomware analysis is the process or method of discovering the sources and possible effects of a malware sample.

Infection vectors

Ransomware infestations are routinely carried out through a variety of infection vectors. The payload is distributed as an attachment in spam emails sent through botnets and other infected servers, which is the most typical vector. Exploit kits are another popular way to get infected. Exploit kits are software programs that scan a system for vulnerabilities before infecting it with dangerous malware. Drive-by downloads are another common technique of infection in which consumers are directed to illicit websites that contain dangerous malware.

Static analysis collects information from binary code without running it. Instead, it analyses all alternative execution routes. It is prone to obfuscation of code. The binary sample is executed in a virtual machine environment for behavioral-based dynamic analysis [15]. API calls are used to connect user programs to operating systems (OS) by requesting services from the OS kernel. API is a set of well-defined commands that must be invoked and used by applications. Deploying solutions that can learn and detect common behaviours and elements that continue to be repeated is a genuinely effective way of guarding against quickly developing threats.

2. Related Work

In this section, past literature on malware classification is overviewed.

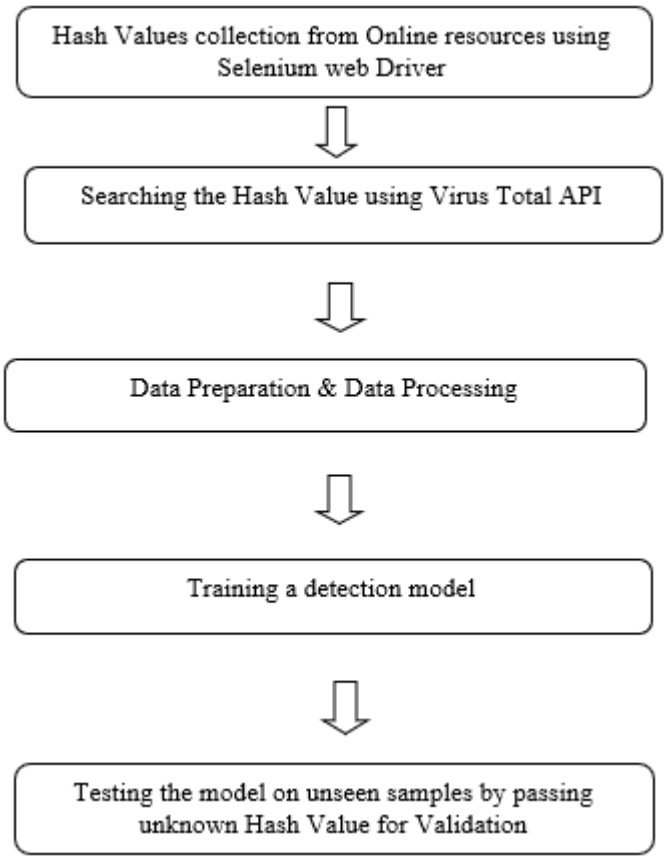
Baldwin and Dehghantanha [14] proposed a method like Zhang et al. [15] for ransomware detection based on Opcodes. Vidyarthi et al. [13] presented a static method for ransomware detection based on the PE header. They extracted features using some values from the PE header fields. Then, they trained a detection model based on Naïve Bayes, J48, and Random Forest classifier. This research uses header information for ransomware detection, while researches for malware detection have shown that most header sections contain important information and increase the accuracy of the method. But this approach can be time consuming as malicious information can be held at the end of the header during parsing thus not a good choice in edge cases.

Manavi and Hamzeh [6] proposed a method for ransomware detection based on PE header. They converted the header bytes into 32*32-pixel images and then fed them to the Convolutional neural network (CNN) to detect ransomware. Using a header is an advantage, but converting it into an image would require a network with more layers to extract its features.

In this work, Application Programming Interface (API) calls are extracted from the executable files and using key value pair information to flag ransomware files and using it to train a classifierto detect unknown ransomware.

3. Proposed Solution

In this paper, the focus is on ransomware and Malware detection using Hash Value analysis. The proposed method is based on MD5 Hashes extracted from the header of Portable executable file.



A. Data Collection via Web Scraping

First, we will be collecting hash codes from multiple sources on the internet, by web scraping, that will scrape hashes (MD5, SHA1) from different sources and displaying in a csv file.

B. Search the Hash Value using Virus total API

Application programming interface (API) call sequences are easily observable and are acceptable selections for malware classification criteria. Then, we will pass the collected hash codes to Virus total API (Virustotal is a service that analyse file and URL for all known kind of malicious file. Used for mainly computing Hash values) | (Api call is a step of operation/procedure performed by the malware/benign file. The Api call includes functions such as: - create, read, write, delete, Modify. Ransomware/malware file have a specific API call Sequence i.e., Unique order of calls API, also known as application programming interface, is a computer program that communicate with operating system using api calls.) and a python script which will pass the hash codes to Myrequest Function and retrieve a dataset as a result of which particular hash code is malicious or not. Thus, Api helps to extract data and functionality using function call.

C. Data Preparation and Pre-processing

In **Data Preparation**, we will be increasing the class of our target variable by flagging of ransomware in our dataset of malware. This technique is performedusing File fingerprinting technique (FFP). FFP uses, SSDEEP which is known as Context piecewise hashes technique. It is also called as Fuzzy Hashes. This technique matches hashes Input with same Input that have similarities. These Input have sequence of Bytes of same order in hashes. SSDEEP is used to compare the hashes with existing hashes from well-known Malware database, such as Malware bazaar, Virus share, Crymeu etc.

Most Antivirus Engines uses this Fuzzy Technique to find new Malware. This isdone via Sequential Pattern

matching and analysis.

In this technique,

- Hash Values are categorized based on hexadecimal values i.e. (a-z, 0-9),
- Iterating the data frame on each Hash code that is present,
- Counting frequent element of hash and populate into its respective column,
- This technique will be able to help in matching of sequence of Byte in Hash.

Now the file is flagged using FFP technique and JSON Value of anti-virus engines. In order to confirm that the malware flagged as ransomware is actuallya ransomware file. So, to avoid any False Positive, we will Use a concept called"Anti-virus Vendor Naming Scheme". It uses Virus Conventional Name to detect and confirm file. It was originated in 1991 as CARO (computer AntivirusResearch Organization). A new naming convention was agreed upon to providea means to avoid confusion on Naming convention of Virus and malware amongdifferent anti-virus software and to provide regulation in Virus Naming Process.

Syntax for virus/malware is as follows: -

Family_name. Group_name.Platform.Variant [: Modifier]

From this syntax, ransomware can be detected as ransomware files are an exception as it shows off its name along with its family name, variant and Modifier beforehand. Thus, this strategy can be used for malware analysis.

Furthermore, we will be conducting data preprocessing in which the data is cleaned and organized i.e., to eliminate bad data which can be redundant, incorrect, noisy data samples. Outliers in highly correlated features are Visualized and removed using IQR (Interquartile range) in which data samples that are out of range are removed from the list. Furthermore, we will be Performing SMOTE, which is Synthetic Minority Oversampling Technique, used to Produce Samples for Minority class by adding some features so that it does not give redundant data all along.

D. Training Detection Model

Now, we will start with our comparative algorithm study, where we will be training our detection model, we will be using various machine learning algorithm such as SVM, Gridsearch random forest, KNN, XGBoost and Ensemble techniques such as Ensemble Max Voting, Blending, Stacking etc.

Splitting the data into 70 percent training and 30 percent testing is the first step in creating the training and testing set. To guarantee that each malware family is fully represented on the split dataset, the dataset is split taking into account the relative populations of each malware family.

The model is evaluated using a k-fold Cross-Validation process, in which the training data is randomly partitioned into distinct subsamples of equal k sizes. The remaining k subsamples are utilized as training data, while one k subsample is kept as validation data. The process is then repeated k times (the number of folds), with each of the k subsamples serving as validation. A 10-fold Cross-Validation procedure performed on the training set to evaluate the model, afterwards the model is tested on the held-out testing set and evaluated for its performance.

E. Testing the Model on Unseen sample

Then we will, repeat the previous step and test the trained model on unseen sample data that was taken from trained data prior to training. Also, we will be passing unseen hash values for validation of our model for unknown samples.

Advantage in the proposed approach

- The reason for using this approach was to enhance the detection Rate and time consumption efficiency,

that was made by the existing approach as only the header file are used to feed directly into the LSTM network.

- Instead of a signature-based approach that was used in existing system, here, dynamic analysis will be used, in which behaviour-based approach to determine the functionality of the malware by studying the actions performed by the given malware.
- The dynamic detection technology can monitor the ransomware in real-time using sandbox environment, which overcomes the shortcomings of static detection technology that cannot detect ransomware with obfuscation and modification techniques.

4. Dataset and Results

This section describes the evaluation metrics, dataset,
experimental environment, and results.

A. Dataset

The dataset used in this paper includes 1,40,000 samples belonging to malware, ransomware and Benign families, which have been downloaded from the virusshare databases and Kaggle. The benign sample have been downloaded from freewarefile3, CryMeu and snapfiles5 websites.

B. Evaluation Metrics

Machine learning evaluation metrics such as Accuracy, F-measure, Precision and Recall have been used to evaluate the performance of the proposed method. To detect overfitting, the cross-validation technique with 10 folds was used in Ensemble Stacking Technique.

C. Experimental Setup

All experiments to the implementation of the proposed method and the works compared are conducted in a system with the following specifications.

- CPU: Intel (R) Core (TM) i7- 4790 with 3.60GHz
- RAM: 16 GB
- Programming language: Python 3.8 version, implemented on Jupyter Notebook Version 4.8.0

D. Experimental Results

After training our model on various classification algorithm i.e., Random Forest,Support Vector Machine, XGBoost, KNN, GridsearchRF. Also, we performed ensemble classification technique as well which will include classification algorithm such as Ensemble Max Voting, Ensemble Stacking & Ensemble Blending. We Performed a Comparative

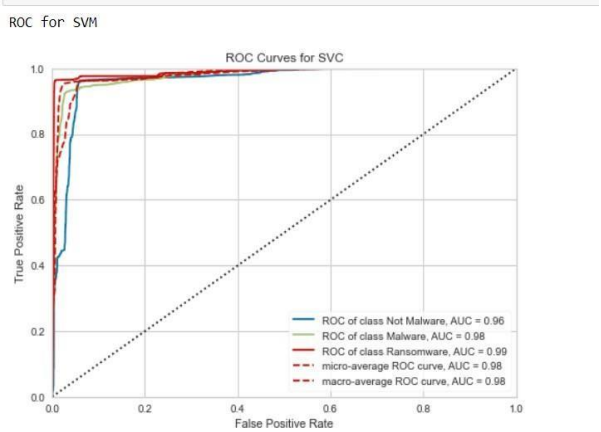


Fig.1: ROC Curve for SVM Classifier

Study on Various Classification algorithm to check whether which of the following algorithm gives out best performance out of the rest in terms of various Performance metrics such as Accuracy, Precision, F1 score etc

The results on classification on Test data on 70:30 train test ratio is as follows:

Train test split-70:30 ratio classification Algorithm used-

S. No	Classifier	Test Accuracy
1.	SVM	80.62%
2.	KNN	85.21%
3.	Gridsearch RF	84.77.%
4.	XGBoost	91.73%
5.	Ensemble Max Voting	93.03 %
6.	Ensemble Blending	92.20%
7.	Ensemble Stacking	90.31%

Table 1: Accuracy Obtained of each classifier in 70:30 train test Split

```
In [224]: print("Confusion Matrix --> Ensemble Max Voting")
          from sklearn.metrics import confusion_matrix

          mat = confusion_matrix(Ytest,y_pred_clf )
          sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False)
          plt.xlabel('True Label')
          plt.ylabel('Predicted Label');
```

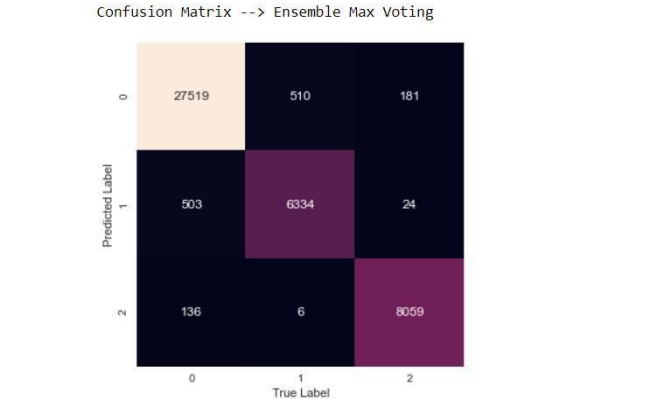


Fig.2: Confusion Matrix of Ensemble Max Voting classifier used to calculate Performcnce metric of that respective classfier.

ThePerformace metrics along with False Positve Rate of Malware class is calculated using the above confiusion matrix using the Formalue

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)}$$

$$Precision = \frac{TP}{(TP+FP)}$$

$$Recall = \frac{TP}{(TP+FN)}$$

$$F1 - Score = \frac{2*Precision*Recall}{(Precision+Recall)}$$

FPR= FP/(FP+TN)

Here, False positive and True Negative are calculated to be 516 & 35,895 samples respectively. These metrics can be used to calculate, False Positive Rate of Malware class. Same can be used to applied to ransomware and Benign class as well.

S.no	False Positive rate Obtained	FPR in Previous Research
1.	0.0142	0.0744 [13]

Table 2: False Positive Rate comparison with respective to previous research.

5. Conclusion

Detecting malware attack is more preferable than dealing with their repercussions, including as downtime, reputational harm or other issues. The technology being used by cybercriminals in growing numbers, in an attempt to lock down whole networks in hopes of extorting ransoms in the hundreds to millions of dollars range. Ransomware is a significant threat to any sector as it compromises the confidentiality**,** integrity and availability of information which is the pillar of cybersecurity itself. When a machine or a device is infected by ransomware, the files and other data are typically encrypted, access is denied and ransom is demanded.

In this paper, we are proposing an approach to prevent ransomware attack using dynamic analysis.

We used seven classifiers (KNN, SVM, Random Forest, XGBoost, ensemble (stacking, max voting and blending).

The results show that detection by its respective API calls is a robust approach towards detecting malware with (i) high accuracy, (ii) low false positive. Ensemble Max Voting classifier and XGBoost are the best classifiers that gave out the best performance metrics from its respective confusion matrix

Applications occur in a variety of industries, including IT, healthcare, and, in particular, data analytics and forensics, where data is the primary asset. This strategy should be suitable in that industry where this method is derived to identify malware assaults and save data from zero-day attacks.

References

1. Jyoti Landage, & M. P. Wankhade, (2019) “Malware and Malware Detection Techniques: A

Survey", International journal of engineering research and technology.
2. S. -J. Lee, H. -Y. Shim, Y. -R. Lee, T. -R. Park, S. -H. Park and I. -G. Lee, "Study on Systematic Ransomware Detection Techniques," 2021 23rd International Conference on Advanced Communication Technology (ICACT), 2021, pp. 297-301, doi: 10.23919/ICACT51234.2021.9370472.
3. S. Sheen and A. Yadav, "Ransomware detection by mining API call usage," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 983-987, doi: 10.1109/ICACCI.2018.8554938.
4. L. B. Bhagwat and B. M. Patil, "Detection of Ransomware Attack: A Review," in Proceeding of International Conference on Computational Science and Applications, 2020.
5. A. Kharraz, W. Robertson and E. Kirda, "Protecting against Ransomware: A New Line of Research or Restating Classic Ideas?" in IEEE Security & Privacy, vol. 16, no. 3, pp. 103-107, May/June 2018, doi: 10.1109/MSP.2018.2701165
6. F. Manavi and A. Hamzeh, "Static Detection of Ransomware Using LSTM Network and PE Header," 2021 26th International Computer Conference, Computer Society of Iran (CSICC), 2021, pp. 1-5, doi: 10.1109/CSICC52343.2021.9420580.
7. Gandotra, E., D. Bansal and S. Sofat, 2019. Malware Analysis And Classification: A Survey. Journal Of Information Security.
8. I.Firdausi, C.Lim, A.Erwin, and A.S.Nugroho (2018). "Analysis of Machine Learning Techniques Used In Behavior-Based Malware Detection", Advances in Computing, Control and TelecommunicationTechnologies (ACT), 2018 Second International Conference on (pp.201-203). IEEE infosecurity-magazine."https://www.infosecurity-magazine.com/ (accessed Nov. 25, 2021).
9. AhmetYazi, Ferhat Ozgur Catak, & EnsarGul, (2019) "Classification of Methamorphic Malware with Deep Learning (LSTM)", 10.1109/SIU.2019.8806571
10. Nwokedi Idika and Aditya P Mathur. A survey of malware detection techniques. Purdue University, page 48, 2007.
11. Rafiqul Islam, Ronghua Tian, Lynn M Batten, and Steve Versteeg. Classification of malware based on integrated static and dynamic features. Journal of Network and Computer Applications, 2012.
12. Mohammed, Ban. (2020). "Ransomware Detection using Random Forest Technique." ICT Express. 6. 325-331. 10.1016/j.icte.2020.11.001.
13. H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on N-gram of opcodes," Futur. Gener. Comput. Syst., vol. 90, pp. 211–221, 2019.
14. D. Vidyarthi, C. R. S. Kumar, S. Rakshit, and S. Chansarkar, "Static Malware Analysis to Identify Ransomware Properties," Int. J. Comput. Sci. Issues, vol. 16, no. 3, pp. 10–17, 2019.
15. J. Baldwin and A. Dehghantanha, "Leveraging support vector machinefor opcode density based detection of crypto-ransomware," in Cyber Threat Intelligence, Springer, 2018, pp. 107–136.
16. Rhodes, M., Burnap, P., Jones, K.: Early-stage malware prediction using recurrent neural networks. Comput. Secur. **77**, 578–594 (2018). arXiv:1708.03513 [cs.CR]