Malware Data Classification

Kranthi sai Davuluri
Masters Student of Computer Science
Arizona State University
Tempe, Arizona
kdavulur@asu.edu, 6316376078

I. Introduction

Malware is any kind of program, that has malicious affect on system and user. It affects the system by infectives several files. It also effects the user by tracking users financial details and exporting details to some hacker via a malicious server. Thsi kind of activity resluts in huge financial loss and disrupt country economy. With the growing billions of terabytes of data, it is important for any anti malware vendor, to detect and classify malware. This helps in tracking new malware infected by system based on its properties. As malware authors often use signature based malware code. It is important to study and classify various malware signatures and its implementation details. Normally, authors use packed malware executables to hide malicious code. So Our first job in malware classificationis identifying packed executables. This often involves study of malware under its execution using environments like sandbox, virtual machines, etc. So that the execution of malware do not impact the system. In this project, out task is to classify malware into nine families. Dataset is prvided by Kaggle. We were given asm files of unpacked executables. and byte file corresponding to each malware file. Our task is to find the features from asm files and byte files and come up with good classification technique. We were given a set of malware files correspong to 9 different fsmilies.

− The 9 different families and their brief descriptions are as follows

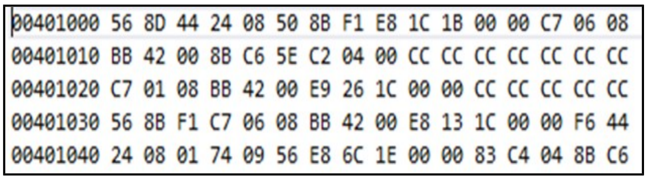
Ramnit
Lollipop
Kelihos_ver3
Vundo
SIMDA
Tracur
Obfuscator.ACY
Kelihos
Gatak

Our next step is to extract features from above files. Then apply various machine learning techniques to classify a new malware file corresponding to its family. Then we compared accuracy using cross validation and also found log loss as evaluation metrics.

II. DATASET

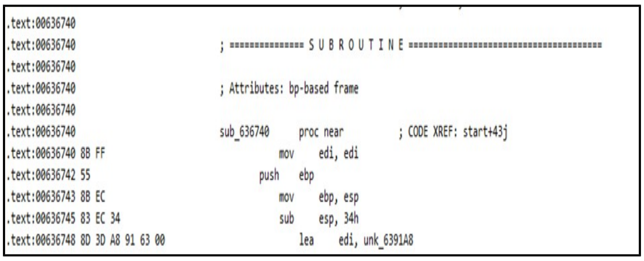
The dataset was provided by Kaggle. It contains train.zip and test.zip files. The train data contains 10868 malware files, each file classified into one of 9 different classes. For each malware file there were asm file and .bytes file. The .bytes file contains the hexadecimal representation of the file's binary content without portable executable (PE) header. The asm file contains the metadata manifest which contains various information extracted from binary content.

The test set contains 10873 files for which the class has to be predicted. The test set also contains the asm and byte file similar to training data. Sample data of both .byte and asm file are shown in Fig.1 and Fig.2 respectively.



```
00401000 56 8D 44 24 08 50 8B F1 E8 1C 1B 00 00 C7 06 08
00401010 BB 42 00 8B C6 5E C2 04 00 CC CC CC CC CC CC CC
00401020 C7 01 08 BB 42 00 E9 26 1C 00 00 CC CC CC CC CC
00401030 56 8B F1 C7 06 08 BB 42 00 E8 13 1C 00 00 F6 44
00401040 24 08 01 74 09 56 E8 6C 1E 00 00 83 C4 04 8B C6
```

Fig. 1. Sample .byte file data



```
.text:00636740
.text:00636740                    ; =============== S U B R O U T I N E =======================================
.text:00636740
.text:00636740                    ; Attributes: bp-based frame
.text:00636740
.text:00636740                    sub_636740      proc near          ; CODE XREF: start+43j
.text:00636740 8B FF                              mov     edi, edi
.text:00636742 55                           push    ebp
.text:00636743 8B EC                              mov     ebp, esp
.text:00636745 83 EC 34                           sub     esp, 34h
.text:00636748 8D 3D A8 91 63 00                       lea     edi, unk_6391A8
```

Fig. 2. Sample .asm file data

III. Methodology

Feature Extraction – This step is one of the most important step in our project phase. This step involves understanding of PE asm file and byte file. Along with this, there is need to study several research papers to classify malware. One such report which I studies was Symantec report on Ramnit virus. From this report, I understood how dll files make huge impact on type of various, as they access specific pots and specific severs. So this is one of the crucial step for analyzing several malware files. Results with this step is explained in the next section.

We have extracted the following types of features from .asm files

- Frequency of bag of words
- Frequency of .dll files
- Frequency of section headers
- Features extracted from 8086 instruction set

We have extracted the following types of features from .byte files

- Frequencies of Unigrams
- Frequencies of N-grams

Frequencies of .dll files – As explained before, we found the frequency of each dll function call from each file. We have evaluated this feature on different classification techniques as described below. However, this feature did not correspond to best feature as expected before. Then we found from other research reports that, malware authors use code obfuscation technique to not to detect malware based on dll function calls.

Frequency of section headers:

By understanding structure of PE file, we understood there nine predefined sections. However by analyzing PE files we found that there were other sections that were not there in pre defined sections. So we found the frequency of section headers.

Malware Classification: We have classified malware into their respective classes from the extracted features using the following classifiers.

- Multinomial logistic regression
- Random Forest
- Feed forward Multi perceptron Artificial Neural Networks
- Decision Tree
- Support Vector Machines.
- Genetic Algorithm.

Classification using Decision Tree Learning – Decision Tree Learning is one of the best model for supervised Learning classification. Since, the data as described in section 1 is labelled data, we use Decision tree as one of the base model for malware classification. After feature extraction, various experiments has been carried out on extracted features with feature aggregation. Decision Tree Learning shows promising results with about 97 percent accuracy in classifying malware files most of the times. For the extracted features,We have tried with decision tree learning on various combination of features.

IV. Results

We divided our data set into 70 percent training and 30 percent test set. Confusion matrix is used to compute Train set and test set accuracies. We have also used 10 fold cross validation in computing classification accuracies.

TABLE I
10 FOLD CV ACCURACIES OF DIFFERENT VARIANTS OF DECISION TREE LEARNING FOR BAG OF WORDS (2993) AND NGRAMS COMIBINED FEATURESET.

Training Accuracy	Test Accuracy	**Cross Validation Accuracy(10 fold)**
1	0.985510	0.983801

TABLE II
10 FOLD CV ACCURACIES OF DECISION TREE LEARNING FOR BAG OF WORDS(42)

Training Accuracy	Test Accuracy	**Cross Validation Accuracy(10 fold)**
0.99785	0.96664	0.971225

TABLE III
10 FOLD CV ACCURACIES OF DECISION TREE LEARNING FOR BAG OF WORDS(42) AND NGRAMS COMIBINED FEATURESET

Training Accuracy	Test Accuracy	**Cross Validation Accuracy(10 fold)**
1	0.9979	0.979976

Above are the training set, test set and cross validation accuracies for Decision Tree Learning for different combination of features.

The dataset as described is divided into 60% training data and 40% test data. Based on experiments on several feature aggregation, the combined features of 2999 bag of words features and 204 tri grams features, we obtained test set accuracy of 98.55% and 10 fold cross validation error is 98.38%. We used scikit library DecisionTreeClassifier() to implement Decision tree Learning classifier.

Learning Experience –

- Feature extraction is one of the most important task and required good understanding of the problem description.
- Also, it requires lot of patience, the features that we think might perform well, will not produce accurate results. For example, features like frequency of dll files and frequency of section headers. However,

feature aggregation sometimes produce better results than individual features.

- Decision tree learning algorithm provides a base model for analyzing malware classification. Also, this algorithm works fast with producing accurate results. Often, we will get training error close to 1 , so we get a doubt that model may over-fit the data. But, analyzing cross validation error and log loss proved the model to be good model.
- Analyzing features with various machine learning algorithms provides a good estimate of classification problem by comparing the results of log loss and accuracy.
- Dimensionality reduction is a good technique to look for Feature reduction. It reduces unimportant features that do not involve in classification. So provide better computational complexities.

TEAM MEMBERS –

1. Abhinav Kilaru
2. Akarsh Cholaveti
3. Aravinda Kumar Reddy Yempada
4. Kranthi Sai Davuluri
5. Phani Santhosh Vamsi Deepak Darbha
6. Venkata Guru Sai Srikanth Vemulakonda

7. REFERENCES

[1] Data dimensionality reduction based on genetic selection of feature subsets K.M. Faraoun 1, A. Rabhi2

[2] Babak Bashari Red, Maslin Masrom, Suhaimi Ibrahim, Subariah Ibrahim, "Morphed Virus Family Classification Based on Opcodes Statistical Feature Using Decision Tree ", Springer, Informatics Engineering and Information Science Communications in Computer and Information Science Volume 251, 2011, pp 123-131

[3] Yibin Liao, "PE-Header-Based Malware Study and Detection" -The University of Georgia