CS 260 : Machine Learning Algorithm
Project Proposal

Manika Mittal and Ronak Sumbaly

November 4, 2015

MOTIVATION

Malicious Software a.k.a Malware is a code snippet or an intrusive software which is written with an intent to disrupt computer operations, gather sensitive information from systems, inject computer viruses, trojans etc. While several schemes have been proposed to detect malware, obfuscation and polymorphism is used in order to elude the detection mechanisms, thus rendering the standard practices of identifying similarities/patterns in the malware software useless. Introducing these techniques means that the malware belonging to the same families, with similar forms of behavior are being changed constantly to look different. Thus we need to rely on analysis of the binaries that are generated by malwares, to classify the malwares into different families and hence detect them. This is the motivation on which our project has been proposed.

BACKGROUND

Excellent technology already exists in the field of detection of malicious executable in the form of various anti-virus softwares such as Avast, Norton etc. All these anti-viruses use signature based detection where they check the contents of a file against a dictionary of known virus signatures. Although this approach can effectively contain the known virus outbreaks, this technique fails when the system encounters an unknown virus and thus analysis of binaries, generated by malware, forms a more reliable means of malware detection. Since analysis of binaries is a general field it is host to wide range of approaches. In literature most of the work follows the framework of extracting features from malware binaries, applying a machine learning system to generate a model to classify malware and finally analyzing aspects of scaling up the framework to identify malware for large data-sets.

Most of the data-sets previously used to construct the classification model only comprised of the malware binaries. In this project we are also considering the ASM file generated via the IDA disassembler tool, giving us more ground to find patterns.Feature extraction in the existing models is based on manual selection of unique features which were identified by "looking" at the binary files. Most of the models have not employed a systematic machine learning approach to extract the frequently occurring features. The few who have, have intuitively applied the n-gram model [1], with n = 2, 4, 10 etc. The selection of n did not have any machine learning basis. We intend to use frequent item-set approach which should help us construct a solid framework to use the n-gram model.

Previously employed machine learning systems, are Gradient Boosting [2], Random Forest, SVM [3], Naive Bayes [4] and Decision Trees [5] which have achieved comparable accuracies.We intend to use Random Forest and SVM to model our classifier because of their appropriateness to the problem statement. Our aim is to improve these accuracies, given the huge data-set which contains not only the binaries but also the ASM files. If time permits our aim is to test the performance of Deep Learning in this situation, since it hasn't been used before. However, our main focus will remain feature extraction because we feel that significant attention has not been given to it.

PROPOSED WORK

A high-level description for the proposed work includes extraction of features from our data-set, selection of the most befitting features which would then be employed for the construction of the classifier. The classifier will then be evaluated using the test data.

1. **Feature Extraction**
   The current dataset consists of malware binaries and OPCODE. We plan to use the technique of frequent itemset mining in order to identify frequently occurring patterns and sequences in the files. Since malware belonging to the same families have similar behavior, it is expected that they have similar patterns in their binaries and asm. Apart from the commonly occurring patterns, we also aim to use properties like file size, line count, their distribution etc. We hope to find interesting patterns by using the frequent item-set technique and move forward from there.

2. **Feature Selection**
   Initially we plan to employ a Greedy algorithm to obtain our feature vector. We do this to get a full comprehensive set of features that could occur in the data-set. Since wrapper-based approaches (using cross validation) are compatible with Random Forests, we plan to use it to tweak our initial feature vector.

3. **Classification Algorithms**
   We intend to assess the adequacy of our feature vector by applying different classification algorithms such as Random Forest, SVM and if time permits Deep Learning. Since our data-set is different from the ones used before, certain modifications will have to be done in the existing algorithms to take advantage of the diverse data.

TIMELINE

The timeline for performing the project is divided on a weekly basis as shown below.

Table 1: Work Timeline

Week	Time Period	Description
1	Nov 5^{th} - Nov 12^{th}	Feature Extraction & Selection
2	Nov 13^{th} - Nov 19^{th}	Applying Random Forest approach
3	Nov 20^{th} - Nov 26^{th}	Applying SVM approach
4	Nov 27^{th} - Dec 4^{th}	Comparison of accuracies (Optional: Deep Learning)
5	Dec 5^{th} - Dec 10^{th}	Fine tuning, Report & Poster design

DELIVERABLES / EVALUATION

The expected outcome of this project is to create a model which can efficiently classify malwares into different families which itself can be used in malware detection softwares. The model is evaluated using the multi-class logarithmic loss. For each file in the training data-set a label is assigned denoting the family of malware that it belongs to and a set of predicted probabilities for each class label. The logarithmic loss is calculated using,

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\ log(p_{ij})$$

where, N = # of test data files. M = # of labels. $y_{ij} = 1$ if correctly observed and 0 if incorrectly observed. p_{ij} predicted probability that observation i belongs to class j.

DATA

For this project we are using a subset of the Microsoft Malware Classification data set which consists of known malware files representing a mix of 9 different malware families. Each unique file is segregated on the basis of two features.

1. **ID** #: Made up of 20 character hash value, giving each file an unique identity.
2. **Class Label** : An integer ranging from $[1-9]$ representing the family name of the malware file.

Each malware file further comprises of a,

1. **Binary File**: Raw data representing the file's binary content (hex dump) in **hexadecimal format**.
2. **ASM File**: Meta-data manifest consisting of a log of the various meta-data information obtained from the binary file.

The entire data-set has been partitioned into **5000 x 2** (Binary + ASM) files for the Training Model and **5000 x 2** (Binary + ASM) files for the Testing Model.

Sample Data Representation

Filename : $0A32eTdBKayjCWhZqDOQ$ **Class Label** : $2 - \{Lollipop\}$

Snapshot Binary File - $0A32eTdBKayjCWhZqDOQ.bytes$

00401200	04	51	52	FF	D0	C7	05	AC	49	52	00	00	00	00	00	B8
00401210	02	00	00	00	C2	08	00	CC	CC	CC	CC	CC	CC	CC	CC	CC
00401220	6A	04	68	00	10	00	00	68	66	3E	18	00	6A	00	FF	15

Snapshot ASM File - $0A32eTdBKayjCWhZqDOQ.asm$

.text:00401217	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	align	10h
.text:00401220	6A	04								push	4	
.text:00401222	68	00	10	00	00					push	1000h	

SOFTWARE TOOLS / LIBRARIES

The entire project will be written in **Python 2.7.9** and the libraries that will be used include:

1. *scikit-learn 0.16.1* - data analysis and machine learning algorithms.
2. *numpy 1.9.2* - scientific computing of feature vectors, N-dimensional arrays etc.
3. *scipy 0.15.1* - supplements the numpy library for optimization, statistics etc.
4. *pandas 0.16.0* - data manipulation and analysis.
5. *pylearn2* - machine learning library for scientific application.

TEAM

The project team comprises of:

1. Manika Mittal - 104566287
2. Ronak Sumbaly - 604591897

PRIOR DISCUSSION

The project proposal was discussed with Nikolaos Karianakis on 3^{rd} November.