Advanced Persistent Threat Data Generation and Detection System

TABLE OF CONTENTS

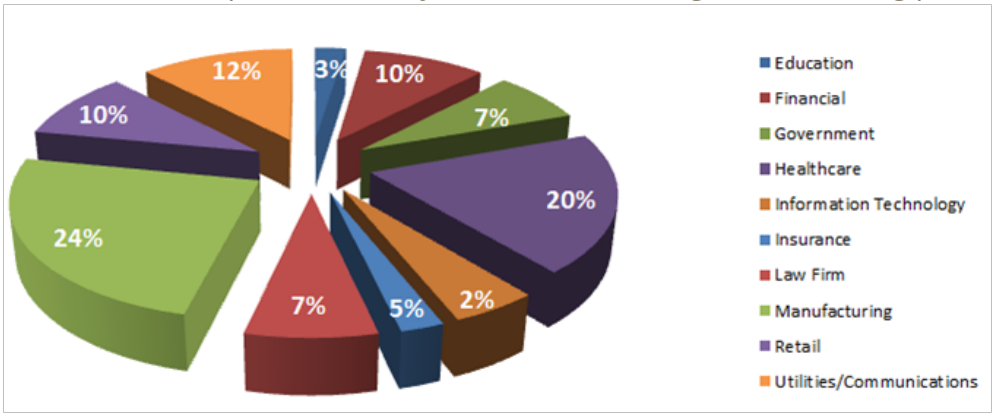
1. Abstract

An advanced persistent threat (**APT**) is a network attack in which an unauthorized person gains access to a network and stays there undetected for a long period of time.
Our goal is to create a solution which can help us analyze local traffic data and classify a stream of packets as malign (attack) or benign (good/safe) stream. We try to achieve this by creating an optimal classifier, capable of discerning between safe and attack packets, with a good accuracy level. We will use a unique approach here of using IP packet sizes/lengths here.

2. Introduction

2.1. APT:

An advanced persistent threat (**APT**) is a network attack in which an unauthorized person gains access to a network and stays there undetected for a long period of time.
The intention of an **APT** attack is to steal data rather than to cause damage to the network or organization. Therefore, it has become very important to detect these attacks as early as possible.
The effect of APT's on the present society can be seen through the following picture.



2.2. Goals:

Our goal is to create a solution which can help us analyze local traffic data and classify a stream of packets as malign (attack) or benign (good/safe) stream.
It would bring us one step closer to countering the many APT attacks that are happening nowadays. As part of the project, we were asked to discern the APT problem, and implement a way to detect an APT from a stream of packets.

- We have used the approach of analyzing the IP packet lengths in a stream here as our predictor variable.

- This means that packet length is the main feature in our classification method.
- In one approach,
    - We operate on the frequency distribution of the IP packet length.
    - And in other the pattern generated by the length of the packets in a stream.

3. Data Collection, parsing & sampling

a. The data was collected using real world, traffic dumps using tcpdump and a tool called wireshark.
b. Each data is either categorized as '***attack'*** or '***benign'***.
c. '***Benign'*** data refers to safe network data. '***Attack'*** n/w data refers to packets captured over the wire while the system was under attack, hence potentially dangerous stream of packets.
d. We collected data & placed them in their respective folders for attack and benign. Each data is a stream of packets in the '***pcap'*** or '***cap'*** format.
e. We then parse the packets and coalesce the data from all the sources into a single training file.
f. The packets are parsed using Python, and are labelled according to the folder it was coming from E.g.: ***attack*** or ***benign***, and we create a file 'train.tsv' with our IP packet length feature from the respective packets.
g. Sample Training Data:

Id	type	Data
0	attack	54 54 60 1109 85 60 .....
1	attack	42 60 343 342 54 .....
2	benign	111 74 74 66 339 66 .....
3	attack	74 74 66 339 1392 .....

h. We then sample our training and testing data from the same training data.
i. We also use a unique random train file which contains data from a totally external source, to see how well our classifier does.

4. Classification Approaches Used

We took two approaches to using the IP packet length as the determining factor for the classifier.

I. Using the normalized frequency distribution of the observable packet lengths.

II. Creating a pattern using Packet Length, for a particular stream.

I. Using the normalized frequency distribution of the observable packet lengths:

a. Based on the results of our preliminary studies, we decided to operate on the frequency distribution of the IP packet length. Here we neglect information regarding packet order and timing.
b. Our instances closely resemble the typical document representation in the domain of text mining known as the ***Bag of Words*** model.
c. In Bag-of-words, documents are represented as term frequency vectors which are similar to our packet size frequency vectors. And each stream of packets can be considered as a document. We need to classify this stream into a 'attack' or 'benign' document.
d. We apply some additional transformations to increase our accuracy.
e. We then apply our classifiers. Here we have used, Naïve Bayes, KNN, Random Forest & SVM.

Additional Transformations Used :

1. TF (term frequency) Transformation
   - Using the raw occurrence frequencies, the decisions of the MNB classifier are biased towards classes which contain many packets and/or packets with high frequencies.
   - In text mining this problem is solved by a sublinear transformation of the frequencies: $f^*_{xj} = \log(1+f_{xj})$. This is referred to as term frequency (TF) transformation.

2. IDF (inverse document frequency) Transformation
   - The **inverse document frequency** is a measure of how much information the word provides, that is, whether the term is common or rare across all documents.
   - The MNB classifier treats all attributes (packet sizes) equally, neglecting their relevance.
   - In fact, some packet sizes are part of every instance and do not confer much information about the class. This is similar to the classification of text documents, where this problem is alleviated using the inverse document frequency (IDF) transformation.

II. Creating a pattern using Packet Length, for a particular stream

a. Here we try to use the pattern in which lengths of a packet occur in a stream as a predictor.

b. Feature Used: Length of all packets that make an attack or benign traffic. A string of lengths of all the packets delimited by space.
c. Intention was to train a NaiveBayes classifier considering only the packet lengths for classification.

5. Results

Using Packet Length as a Pattern:

Model Used: Naïve Bayes Classifier
Accuracy: 3-4%

Using Packet Length Frequency:

- We see that after applying tf-idf the accuracy increases.

Results for test data sampled from training data.

tf-idf?	Clf	Accuracy
Without Tf-idf	Multinomial Naïve Bayes	90%
With Tf-idf	Multinomial Naïve Bayes	100%

- We see that Accuracy for almost all the classifiers is 100%.

Results for test data sampled from training data.

Clf with tf-idf	Accuracy
Multinomial NB	99.88%
SVM	100%
Random Forest	100%
KNN	100%
SVM	100%

- We see that Accuracy for unknown data falls drastically.

Results for test data sampled for random data.

Clf with tf-idf	Accuracy
Multinomial NB	33.33%
SVM	66.67%

6. Conclusions

Using Packet Length Frequency:

- We got a very high prediction rate with the use of Frequency of the packets. Hence we can conclude that this is a very valid & good predictor for classification of malign or benign network stream.
- We also showed that applying the *Tf-idf* transformations to our data increases the prediction accuracy.
- The only drawback is that it needs a lot of training data for the classifier to correctly predict the values, i.e. it will not be as good for new attacks as it will be for old ones.
- Though we also showed that using SVM instead of Multinomial Naïve Bayes, we can increase the chances of predicting a new attack with a better accuracy.

Using Packet Length as a Pattern:

- The classifier performed with a very low accuracy because only one feature, length of packets, was considered for building the model.
- The observed difference in packet length between attack and benign traffic was significantly less and thus could not make a good feature for use in classification.
- For a probabilistic classifier like NaiveBayes we need more significant features that help distinguish between classes of data. A single feature that is a weak differentiator reduces the model accuracy as shown.

7. Inference

- Hence we have showed how we can apply text mining techniques to analyze the network data.
- We successfully treated the problem as a bag of words problem and got good accuracy results.
- Taking frequencies of occurrences of packet lengths in a stream of data as a feature is shown to give a good model.
- This topic leaves a lot to be explored, and there are many possibilities.
- For an APT system, we can use packet length as a classifier, but we should certainly be adding more features to make it more robust.

8. References

[1] https://en.wikipedia.org/wiki/Advanced_persistent_threat
[2] Changing the game: The art of deception against sophisticated attackers. Nikos Virvilis, Oscar Serrano
[3] Sherlock Holmes and The Case of the Advanced Persistent Threat. Ari Juels, Ting-Fang Yen
[4] Network Traffic Classification under Time-Frequency Distribution. Andr´e Riboira, Angelos K. Marnerides.
[5] https://en.wikipedia.org/wiki/Bag-of-words_model
[6] https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words
[7] http://scikit-learn.org/stable/modules/feature_extraction.html#the-bag-of-words-representation