Entropy and n-gram Analysis of Malicious PDF Documents

Himanshu Pareek
Research Scholar
JNT University
Hyderabad, India

P R L Eswari
Team Coordinator
C-DAC Hyderabad
Hyderabad, India

Dr. N. S arat Chandra Babu
C-DAC Banglore
Banglore, India

Abstract

Malware is a persistent problem in field of computer security and its complexity has increased multiple folds in past decade. In past few years malicious documents attacks have emerged as preferred method to bypass the security of a host computer. This work assumes that such kind of exploits do not carry any important information and hence should not be highly random. However use of randomness is not a deterministic approach for detection of malicious code but gives a useful indication to malware analyst. This research report presents two discrete analyses of malicious PDF documents. One is entropy and other being n-gram term frequency.

1. Introduction

Malware attackers continuously discover new vulnerabilities and attack computer users for various motives [1]. Users are generally aware of these threats and install firewall and antivirus software on computer system and keep their systems patched with latest software updates. Attackers bypass these defenses by sending malicious documents which exploit non-patched vulnerabilities in the application software. In this research work, we analyzed the malicious documents and present the results of entropy calculation and n-gram analysis done on such malicious documents. In past, Entropy was used to classify packed executables by Robert Lyda [2]. Degree of randomness in bytes of an encrypted executables should be high. In our research paper, belief is that degree of randomness in an exploit file should be less than genuine file of corresponding format. Researchers usually calculate entropy of their files under consideration. Brandon Dixon [3] has maintained entropy of a large collection of PDF files.

2. Datasets

We obtained various genuine documents mainly from research papers, reports and financial documents. We also obtained various malicious PDF documents using offensive computing.net, malware.lu and contagiodump.blogspot.com. We also acquired some malicious documents from Brandon Dixon.

3. Entropy Definition and our analysis

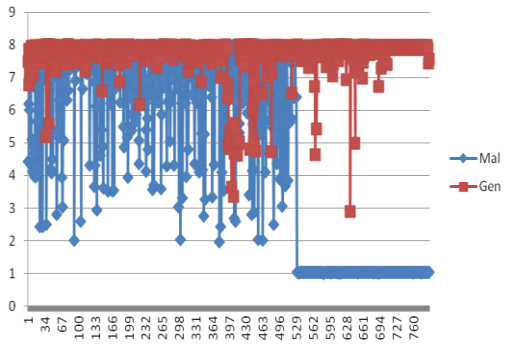
Entropy is a method of measuring randomness or uncertainty in a given set of data. For calculating the entropy of file, our data set is sequence of bytes in the file [4]. Entropy can be calculated by using following formula.

$$H(x) = -\sum_{i=1}^{N} p(i) \log_2 p(i)$$

Where $p(i)$ is the probability of i^{th} unit of information and specifically in our case it is i^{th} byte of file. The value of H(x) will be high if probabilities of occurrence of bytes are low and vice versa. Encrypted Files can be classified using such entropy analysis with understanding that probabilities of occurrence of bytes will be low. In our experiment, we assumed that malicious documents might be carrying false information and padding and hence entropy value should be low.

3.1 Results

We calculated the entropy of malicious PDF files with an intention that it should come low. Following results were obtained.



However as mentioned earlier, the entropy values cannot give a direct indication towards genuine or malicious but combined with other feature vectors can be useful. Based on our results, we deduce the confidence interval of entropy values where we can say that particular document is suspicious. This is shown in Table 1.

Table 1. Results of Entropy Calculation

Dataset	Avg	Min	Max	CI
Genuine	7.70	2.95	8.0	-
Malicious	4.80	0.99	8.0	0.0 – 2.7

4. N-gram and Our Analysis

N-grams are substrings of a large string of length N. They can be better understood as shifting windows over the large string. Suppose there is a string ENTROPY, 1-grams of this string will be E, N, T, R, O, P, and Y. If we take 3-grams of this string then they will be: ENT, NTR, TRO, ROP and OPY. Similarly consider the following sentence: **This is good enough**. Now if we take word as a unit and determine 2-grams then they will be "**This is**", and "**is good**" and "**good enough**". N-grams are used in machine learning and natural language processing related tasks.

We take PDF files and generate hex dumps of these files. Then from these hex dumps, we generate 2-grams from these files with word as a unit. To analyze this n-gram output, we make use of term frequency and TFIDF [5]. Both the terms are defined below.

TF, Term frequency is the numerical measurement of how often a term occurs in respect to other terms in the document. Term frequency of a particular term can be obtained by dividing its frequency by the frequency of maximum occurring term in the document. This can be obtained by following formula.

$$f = \frac{f(t,d)}{\max\{f(w,d): w \in d\}}$$

TFIDF, term frequency–inverse document frequency is a numerical measurement which reflects how important a word is to a document in a collection. This is obtained by multiplying TF by IDF.

$$TFIDF(t,d,D) = TF\ X\ IDF$$

Where IDF is inverse document frequency

$$IDF = \log_2 \frac{N}{DF}$$

Where N is the number of documents and DF is number of files in which the term appears. We collected 792 malicious documents for the task and same number of genuine documents. With this we did the experiment multiple times but program used to continue for a long time file size used to grow very large to be handled by other analyzer programs. So we kept a threshold term frequency of 0.05 and stopped entertaining terms with term frequency of less than a threshold value. The threshold value was chosen on the basis of failed experiments and is just big enough. A bit increase from here used to disrupt the analysis process. We analyzed this data using machine learning. We combined both the datasets final dataset contained following set as following:

$$tuple = \{tf, tfidf, mal0gen\}$$

Where mal0gen is term which can take two values, either 0 or 1. This is kept 0 for genuine documents and 1 for malicious document.

We used J48 algorithm which is open source implementation of C4.5 algorithm in WEKA Tool [6] [7]. This generated following results (Table 2).

Table 2. Results with J48 Decision Tree

	Classified correctly	Classified wrong
Genuine	46422	511
Malicious	65247	289

These results indicate that a model built with TF and TFIDF of PDF documents with the help of J48 algorithm is very efficient with 0.01 percent false positives and 0.0044 percent false negative rate.

5. Conclusion

In this work, we analyzed malicious PDF Documents and assumed that because of their exploit nature they should be prone to detection based on their byte patterns. We found two discreet methods of doing it. One is entropy which provided a confidence interval which clearly indicates a document being malicious. Another is TF and TFIDF which with the help of J48 algorithm can provide a efficient detection method. As part of future work, we intend to combine these two features with other features like JS etc. and detect malicious PDFs.

10. References

[1] Quick Heal Technologies Report, January 2013, available at http://www.quickheal.com/blog/news20/
[2] R. Lyda and J. Hamrock, "Using Entropy Analysis to Find Encrypted and Packed Malware," *IEEE Security and Privacy*, March/April 2007
[3] Brandon Dixon, blog.9bplus.com
[4] Mike Schiffman, "Cisco Blog- Information Entropy", http://blogs.cisco.com/security/on_information_entropy

[5] Salton, G., Wong, A., and Yang, C. "A vector space model for automatic indexing." Communications of the ACM 1975

[6] Ross Quinlan, "C4.5 Algorithm, Generating a Decision Tree, http://en.wikipedia.org/wiki/C4.5_algorithm

[7] Weka, Open source Machine Learning software, www.cs.waikato.ac.nz/ml/weka