International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 06 | June-2016 www.irjet.net p-ISSN: 2395-0072

An Advanced Approach to Polymorphic/Metamorphic Malware Detection using Hybrid Clustering Approach

Priya Sharma[1], Satveer Kaur[2], Jyoti Arora[3]

[1] *Research Scholar, Department of CSE, Desh Bhagat University, Mandi Gobindgarh, Punjab, India*
[2] *Research Scholar, Department of CSE, Desh Bhagat University, Mandi Gobindgarh, Punjab, India*
[3] *Assistant Professor, Department of CSE, Desh Bhagat University, Mandi Gobindgarh, Punjab, India*

Abstract - The detection of Polymorphic/Metamorphic malwares is a very difficult task by the use of only signature based techniques. In this paper, a hybrid clustering approach is used for the detection of various malwares by using both Permission and Signature based techniques in analyzing the behavior of malwares.
Pattern matching technique is involved for comparing the various variants of malware. The patterns are primarily based upon the various system calls when the procedure runs and generates the output. The patterns are entirely based on the library executed by the malware. Instructions are extracted and code is executed for pattern matching between instruction sets.
The proposed model has been prototyped and evaluated using sliding window protocol technique in order to measure the accuracy of different files exist in PC. The results demonstrate malwares into 2 categories and differentiate between Polymorphic malwares and Metamorphic malwares.

Key Words: **Malware, pattern matching, polymorphic, metamorphic malware.**

1.INTRODUCTION Malware becomes a very serious problem in today's PC industry. The personal computers and the files should be protected from the malicious data which tries to infiltrate the contents and files of our system. This approach is based on the observation that the sequences found in the packed or hidden code can be identified in the malware instance when its execution is checked against its static code mode and then report is generated corresponding to it. It becomes crucial to protect the system from such malicious attacks
A virus by definition is a program which when run, rep roduces and infects the computer by causing a threat to the wholeness or completeness of the system. There are more than two types of computer viruses namely Sneaky silence viruses, computer worms, Trojan horses etc. that are turned into secret code and affects the system very badly and even stops the entire functioning of the system. A computer worm is a program, that copies itself and spreads to other comput ers by using the network to which the host machine is connected. Worms cause a lot of harm to the network by using the available ability unlike viruses which affect files in a target host. The nature of damage could be deleting the data from PC, damaging the programs or formatting the hard disk or entire malfunctioning of the system.
Types of Malware:

Figure 1: Various types of malware

Polymorphic Malware: Polymorphic harmful programs or apps is harmful, destructive or rude computer software such as a virus, worm, Trojan or spyware that constantly changes, making it very hard to detect with anti-harmful programs or apps programs.

Metamorphic Malware: Metamorphic malware is that malware which propagates by creating the transformed copies of its code. The generated code, is often called as a variant of the malware, is typically a working program which has the same functionalities as

the original. Metamorphism has been used by malware authors to detect the malwares by static-signature based anti-malware techniques. The malicious programs are virtually undetectable statically and will damage the target computing systems and personal data in our PC's.

2. LITERATURE SURVEY

Schmidt et al. [10] states a host based malware detection system for android platform. It is a signature detection method which applies static approach on executables.Blasing et al. [11] aims to perform static and dynamic approaches on android applications. In [5] a novel method has been developed to detect leaked sensitive phone-related information. Firstly, it decrypts Objective-C binary and generate control flow graph. Presence of leaks - paths arising from functions obtaining sensitive resources is checked in the graph. **Christodorescu and Jha[10]** states that such detection methods can be easily defeated bymetamorphism, whi ch uses code hiding ways of doing things to change the representation of programs. Metamorphic harmful programs or apps can hide (on purpose) their whole code in a variety of ways, such as control flow switching-around, substitution of equal instructions,(number or thing that changes) renaming, etc. This creates a (contest to see who can collect the most guns, etc.) between the metamorphic harmful programs or apps writers (or hiding engine such as Mist fall, Win32/Simile, and RPME[26]. **Zmist** states an advanced metamorphic virus that (shows or proves) a set of polymorphic and metamorphic code writing skills which include hiding/blocking, randomly at the entry points using an added polymorphic (change secret codes into readable messages)or, code combination and arrangement and code (combination of different things together that work as one unit). **Chouchane and Lakhotia[8]** proposed using "engine signatures" in order to detect metamorphic malwares. Basically, this technique evaluates the collected evidence from x86 code segments by the code scoring function. **Wagener et al. [8]** proposed a flexible and automated approach to extract harmful programs or apps behavior by noticing all the system function calls completed in a virtualized execution (surrounding conditions). (things that are almost the same as other things) and distances between harmful programs or apps behaviors are figured out/calculated which allows classifying harmful programs or apps behaviors. The classification process proposed by this work is using phylogenetic tree. However, this way of doing things still has a limitation due to the wrongly classified a few harmful programs or apps behavior. **Bayer et al.** was used an (able to be made bigger or smaller) clustering approach to identify and group harmful programs or apps samples that show almost the same behavior [10]. This approach also (does/completes) energetic/changing analysis to get the execution traces of harmful programs or apps programs using automated tools. The execution traces are generalized into behavioral profiles, which describe/show the activity of a program in more fuzzy and unclear terms. Then the profiles serve as input to a (producing a lot with very little waste) clustering set of computer instructions that allows handling sample sets larger than previous approaches in term of harmful programs or apps behaviors. **Bergeron et al.** proposed a new approach for the static detection of harmful programs or apps code in executable programs [10]. This approach carried out directly on binary code using (related to the meaning of words) analysis based on behavior of unknown harmful programs or apps. The reason for targeting binary executables is that the source code of those programs that need to detect evil and cruel code is often not available. **Lee et al [13]** proposed a malware clustering approach where a modified Levenshtein distance is used and a k-medoid partitional clustering[13]. The complexity of computing distances between malware in their method is quadratic in the number of system calls and may not scale for large number of files with a lot of variability. **Bayer et al [14]** have employed faster approximate nearest neighbor search using Locality Sensitive Hashing for comparison of the analysis reports with known behavior profiles that they have created (using data tainting methods to track system call dependencies).

3. PROBLEM FORMULATION

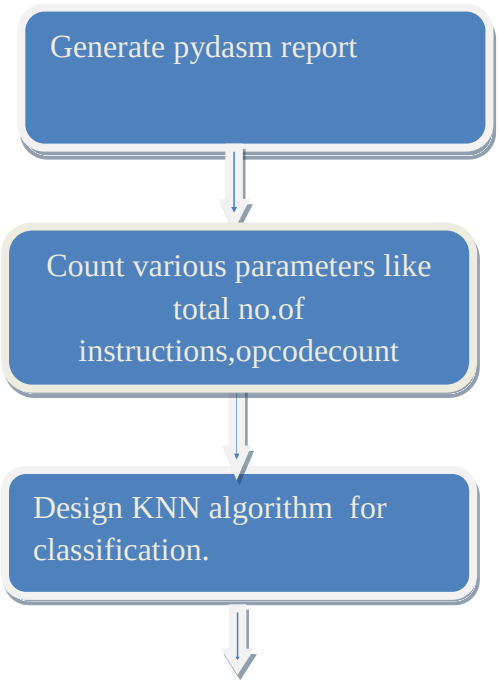
It is very difficult to understand the behavior and classification of malwares. It is very necessary to classify and analyze the behavior in different categories viz. Polymorphic/Metamorphic. The aim of this paper is to classify the most harmful malwares i.e. Poly/Metamorphic malwares by executing the code and extracting the instruction sets by assigning them diff. numbers such as 12, 31, 42, 35 etc. First task is to identify the most common class of malwares and after that pattern matching technique is applied . Then clustering is done to improve the accuracy and efficacy

of system. In the end, classification is done to divide the malwares into 2 separate categories.
The problem earlier was that only signature based technique was used but in that only authentication problem is solved but it does not provide accurate results instead creates complexity in deciding the type of malware. So in this paper we have used signature based technique alongwith pattern matching. This paper consists of parameters various parameters like number of host calls, affected system part, binary registry key, scores generated etc.

4. PROPOSED METHODOLOGY

This paper uses the technique of pattern matching along with signature based technique. In this, firstly the pydasm report of each instruction is generated. Then various no. of parameters that are used viz. total no. of instructions, opcodecount etc. are calculated. KNN algorithm is used for the classification process. Then apply classification technique to identify among polymorphic and metamorphic malwares. For malicious files, text mining technique is applied and for this preprocess all the documents and tokenization is done. Extraction of instructions is done and then instructions are put in the dictionary corresponding to pattern generated.

Workflow Diagram



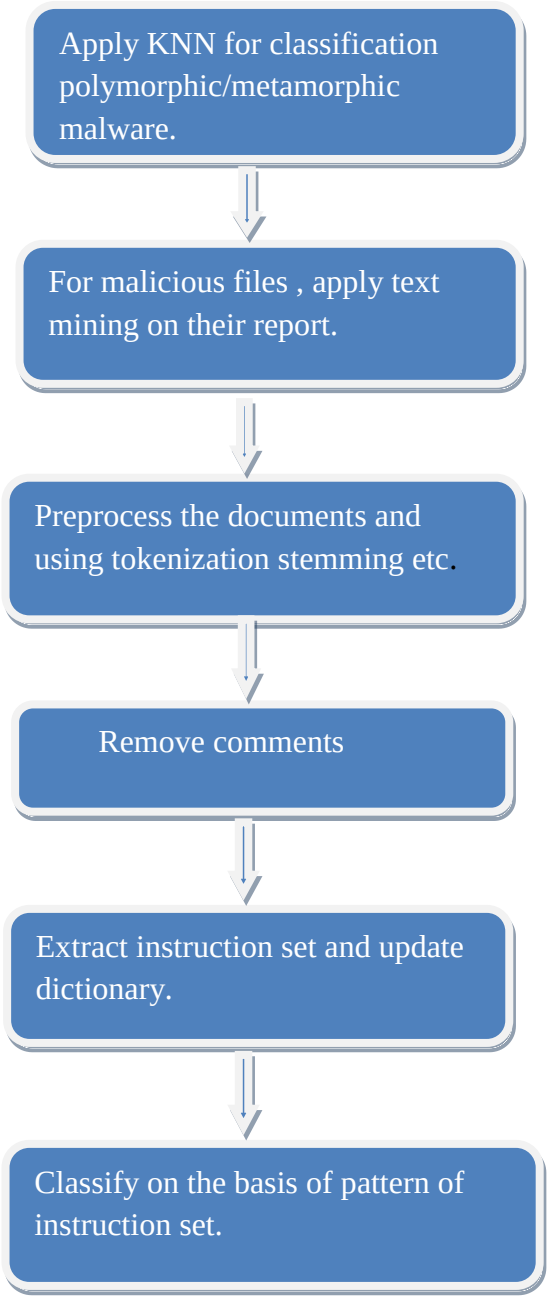


Figure 2: Workflow Diagram

KNN Algorithm: The **k-Nearest Neighbors algorithm** is lazy learning method or a non-parametric method used for classification and retrogression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression. KNN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is

deferred until classification. The *k*-NN algorithm is among the simplest of all machine learning algorithms. Both for classification and regression or retrogression, it can be useful to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of $1/d$, where d is the distance to the neighbor.

5.RESULTS

All the simulation is done with the help of different tools like SciPy, Numpy, Python, stats. The instructions are extracted and opcodes are generated based on given datasets. Two classes of datasets are used i.e. training dataset and testing dataset.

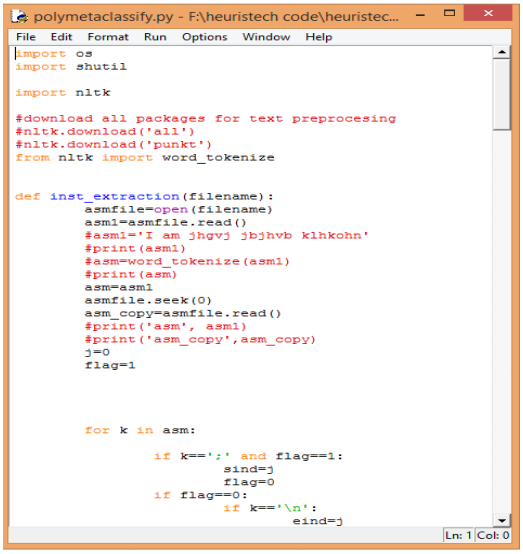


Figure 3: Representing instruction extraction

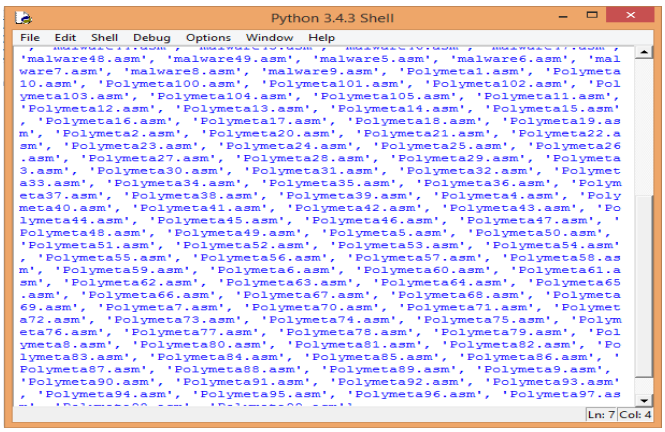


Figure 4: Representing Poly/metamorphic malware

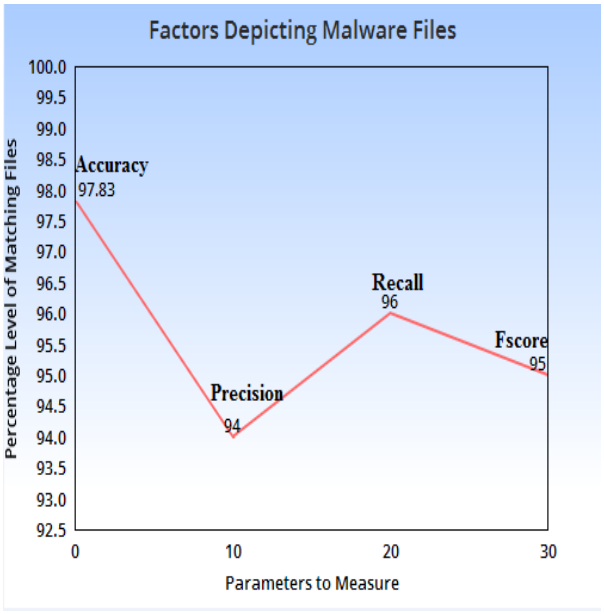


Figure 5: Parameters indicating malware files

6.CONCLUSION

The paper represents a hybrid clustering method of classifying malwares into Polymorphic and metamorphic malwares. The KNN algorithm has been successfully implmented for differentiating among the categories of malwares. Various different folders are made for distinguishing among poly/metamorphic malwares and in the end, classification process puts them according to their class in their respective folders.. This shows the efficiency of our algorithm.

REFERENCES

[1] Bayer, U., Kirda, E., Kruegel, C.: Improving the efficiency of dynamic malware analysis. In: Proceedings of the 2010 ACM Symposium on Applied Computing. SAC 10, pp. 18711878. ACM, New York (2010).

[2] K. Rieck, P. Trinius, C. Willems, and T. Holz: Automatic Analysis of Malware Behavior using Machine Learning, In Journal of Computer Security 2011.

[3] G. Wagener, R.State, A. Dulaunoy: Malware behavior analysis, Journal of Computer Virology(2008) 4:279-287

[4] S. Attaluri, S. McGhee and M. Stump: Pro_le hidden Markov models and metamorphic virus detection, In Journal of Computer Virology(2009) 5:151-169

[5] M. Apel, C. Bockermann, and M. Meier. Measuring similarity of malware behavior. In Proc. of 34th LCN 2009. IEEE Computer Society, 2009.

[6] Lee W, Stolfo S, Mok K. A data mining framework for building intrusion detection models. Proceedings of the IEEE symposium on security and privacy; 2014.