ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (250-254), Month: October - December 2015, Available at: www.researchpublish.com

# Improved Signature-Based Antivirus System

Osaghae E. O.

Department of Computer Science Federal University, Lokoja, Kogi State, Nigeria

***Abstract:*** **The continuous updating of antivirus database with malware signatures degrades the efficiency of the antivirus system. Existing antivirus researchers are finding ways making the malware signatures database gets currents signatures of merging malware threats. However, the astronomical increase in the number of malware signatures update, thereby reducing the performance of the computer system. Consequently in this paper, an attempt was made to group individual malware signatures into a similar pattern, called the family malware signatures. Each group of family malware signatures, cancels all the individual malware signatures with similar patterns into that group of malware signature. The first advantage of this single grouping of malware signature is, the searching time to attempt matching a pattern of executable file with malware signature will be reduce. Another advantage of this approach is in the reduction of many individual malware signatures into a single group of malware signature, hence, reducing the number of signatures in antivirus database and at the same time, making antivirus database more scalable.**

***Keywords:*** **Antivirus, signatures database, Malware signatures, malware code.**

## I. INTRODUCTION

Malicious code is any code that adds changes or removes parts from a software system in order to intentionally cause harm or subvert the intended function [14]. Malicious software can be divided into two categories: those that need a host program, and those that are independent. The former are essentially fragments of programs that cannot exist independently of some actual application program, utility or system program. Some examples of malware software are Viruses, Worms, Trojans logic bombs and backdoors [1], [10] and [11]. According to Kaspersky laboratories in April 2009 alone, 45190 unique instances of malware where found on their customers computers. This worryingly high number is only likely to increase, especially as the malware author's incentives for writing such software is now mainly a financial one. Due to the significant loss and damages induced by malicious executables, the malware detection becomes one of the most critical issues in the field of computer security. Currently, most widely-used malware detection software uses signature-based method to recognize threats. Signatures are sequences of bytes in the machine code of the malware. The inability of traditional signature based malware detection approaches to catch polymorphic and new, previously unseen malwares has shifted the focus of malware detection research to find more generalized and scalable features that can identify malicious behavior as a process instead of a single static signature [12] and [9].

Accordingly, virus techniques grew increasingly throughout all the years, from plainest methods to some more advanced strategies. New vulnerabilities in the system are discovered every few days. These vulnerabilities are fixed by the software vendors who provide patches and updates for the system. Unfortunately, our current ability to defend against new viruses is extremely poor and the basic approach of detection, characterization, and containment has not changed significantly over the years. The complexity of modern malware is making this problem more difficult. Detection Methods have some major problems. Firstly, they are only good against known viruses and not very good against evolutionary or new viruses. Secondary, they tend to take a noticeable amount of time to scan a system or networks for the patterns. Thirdly, a scanner or its virus pattern database must be updated very often to remain effective [3] and [2].

Signature-based detection technique identifies the presence of a malware infection, by matching at least one byte code pattern of the software in question with the database of signatures of known malicious programs. This detection scheme is based on the assumption that malware can be described through patterns (also called signatures). Classic virus-detection techniques look for the presence of a virus-specific sequence of instructions, called a virus signature, inside a program. If the signature is found, it is highly probable that the program is infected. In the specific case of searching for a particular

malicious code instance, it is not only possible, but performed daily by anti-virus software [4]. Pattern based signatures are the most common technique employed for malware detection. Implicit in a signature-based method is a priori knowledge of distinctive patterns of malicious code. The advantage of such malware detectors lies in their simplicity and speed. One of the most common reasons that the signature-based approaches fail is when the malware mutates, making signature based detection difficult [5], [8] and [7].

Existing literature shows that improvements on signature-based detection technique is focus on how to astronomically generate malware signatures for the antivirus database and still, the database continue to increase and not scaled. In this paper, there is an attempt to group individual malware signatures that have similar pattern in the database, into groups of family malware signatures. The idea to have individual malware signatures forming a generic signature is to make the malware signature to scalable.

## II. RELATED WORKS

In this section, we are going to review some of the related research works attempting to group individual malware signatures into a generic signature:

[13] Proposed a detection system by training a set of known values used to represent a file. In other to achieve this, they collected a set of malicious software and benign programs. Specifically, the malware is made up of different kind of malicious software (i.e. computer viruses, Trojan horses, spyware, etc). They used a technique called N-gram, for set of file that acts as a signature based, and then the detection system can classify unknown instances into malicious software and benign program. They classify the unknown malware instance using *k-nearest neighbor algorithm* (Fix and Hodges, one of the simplest machine learning algorithms that can be used in classifying issues). This algorithm relies on identifying the *k* most nearest (say most similar) instances, to later classify the unknown instance based on which class (malware or benign) are the k-nearest instances. The first advantage of this detection system is that it is effective detection system. Second advantage is in the use of n-grams-based signatures methodology, which can achieve detection of new or unknown malware. And last, this method provides a good detection ratio and the possibility to control the false positive ratio. They suggested future research direction in the use of n-gram analysis for malware detection through the use of different and large collection of malware [13].

[5] Presented a method to generate signatures for malware classes to detect previously unknown malicious programs. In their detection system, rather than creating a new signature for every variant in a malware family, they created a single signature that reflects the behavior of the entire family. The beauty of this detection approach is that it reduces the human effort required to generate a signature for a new malware. Also, it is able to detect malicious programs with common obfuscations. Their malware detection approach is space efficient and accurate detection of future variants of a malware family. They observed that the detection error rate for new malware in broad classes such as Trojans and backdoors seems high in their experiments but the results are encouraging. The limitation of their approach is that it does not work for packed malware [5].

[6] He observed the techniques currently used by metamorphic generators are not producing variants that challenge hidden Markov models (HMM). He further noticed that obfuscation process should be able to replace one or more instructions with a different set of equivalent instructions, performing same functions. He used the ideas to implement a signature-based detection system, by analyzing the same set of viruses features based on hidden Markov models (HMM). He observed that HMM-based technique easily detects the viruses and this fact enable them to perform a five-fold cross validation. Among these five sets, four sets were used for training the HMM model and the excluded set was used to test the model. Since it follows five-fold cross validation, five different models were generated and tested for efficient results. For instance, HMM model was able to detect the opcode patterns in these viruses even after obfuscation [6].

## III. MODEL FOR IMPROVE SIGNATURE-BASED ANTIVIRUS SYSTEM

Figure 1 shows the Architecture for Improved Signature-Based Antivirus System. The architecture is made up of the following components namely: *Executable file*, *Antivirus Engine*, *benign*, *Malware*, *Scalable Malware Signature Database*, *Malware Features Updating pool*, *Family malware signatures Reduction Engine* and *Family malware signatures Reduction Buffer Database*.

**i) *Executable File*:** is the file that is submitted to the antivirus system for analysis to check if it is infected by a malicious codes or it is a benign.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (250-254), Month: October - December 2015, Available at: www.researchpublish.com

**ii) *Antivirus Engine*:** is the heart of malware detection system. This is where the executable file is unpacked if packed, disassembles and analysis compare the pattern of executable file with the malware signatures in the scalable malware signature database section. If any signature in the scalable malware signature database matches with an executable file pattern, the executable file is classified as malware, otherwise, it is classified as benign.

**iii) *Benign*:** is one of the classification results from the Antivirus Engine. An Executable file is classified as benign if it no malware signature from the *Scalable Malware Signature Database* matches with the a pattern of the executable file.

**iv) *Malware*:** unlike benign, is also a classification result from the Antivirus Engine. Contrary to a benign classification, an executable file is classified as malware if a malware signature from Scalable Malware Signature Database, marches with a pattern from the executable file being examined.

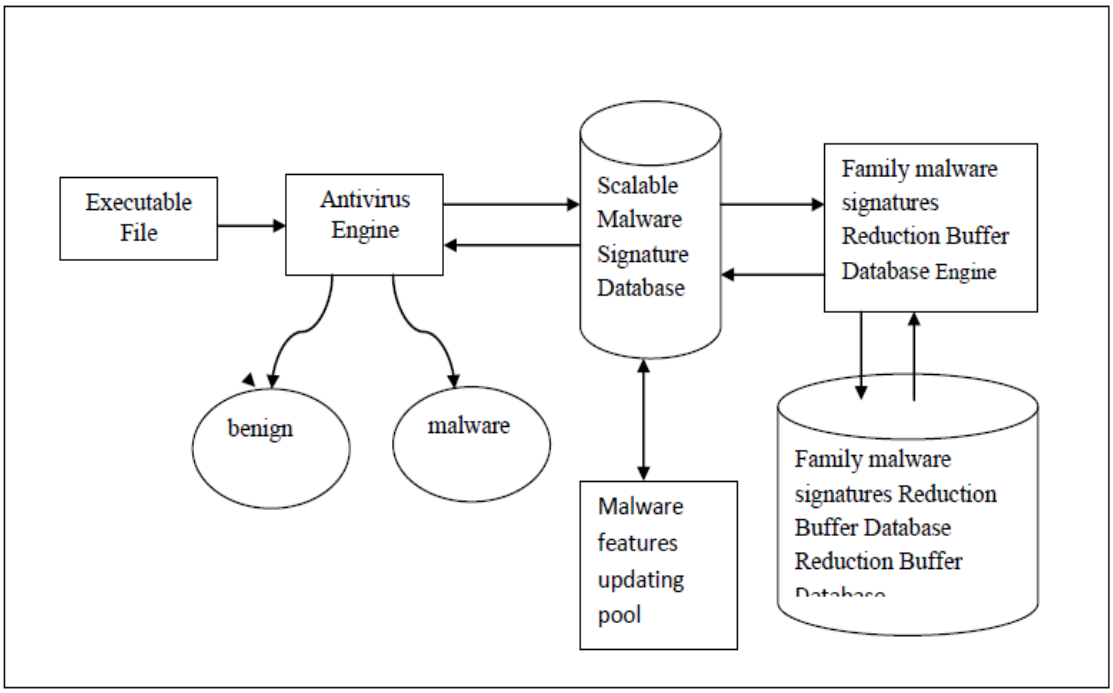


**Fig 1: Architecture for Improved Signature-Based Antivirus System**

**v) *Scalable Malware Signature Database*:** is the database responsible for storing either the individual malware signatures or a pattern of signatures for malware families. This database works with Family malware signatures Reduction Engine and Malware features updating pool, to enable it to either update the individual malware signatures or family malware signatures.

**vi) *Malware Features Updating pool*:** is the pool that enables the antivirus experts to update the Scalable Malware Signature Database with new malware signatures. Also at this point, the antivirus experts can update the detection system when individual malware signatures are found. The expert also update the Scalable Malware Signature Database with signatures for malware families.

**vii) *Family malware signatures Reduction Engine*:** this is the engine that is responsible for attempting to add an individual malware signature to signature for malware family, whenever any is found. After an individual malware signature is added to a signature family, the individual malware signature is discarded from the Scalable Malware Signature Database, thereby reducing the number of malware signatures in the antivirus database. This engine is aided in carrying out its functions by Family malware signatures Reduction Buffer Database, which it uses as a temporary buffer for combining the individual malware signatures into a family malware signatures.

**viii) *Family malware signatures Reduction Buffer Database:*** is an aiding database to Malware Family Reduction Engine. It is a temporary buffer database that periodically accepts individual malware signatures, supplied from Scalable Malware Signature Database, through Malware Family Reduction Engine. At the end of an attempt to transform individual malware signatures into malware signatures families, whether an individual malware signature has found its family or not, the contents of the Buffer Database, is discarded.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (250-254), Month: October - December 2015, Available at: www.researchpublish.com

In Figure 1, the behaviour of Improve Signature-based Antivirus System starts in the Malware features updating pool section. In this section, the antivirus expert updates the Scalable Malware Signature Database with individual malware signatures, and forms the groups of malware signatures families. Then, the Family malware signatures Reduction Engine attempts to automatically checks for individual malware signatures in the Scalable Malware Signature Database, which belong to any of the formed groups of malware signatures families. When Family malware signatures Reduction Engine finds an individual malware signature having a pattern similar to a group signature, that individual malware is discarded from Scalable Malware Signature Database and the group signature now represents its signature. The Family malware signatures Reduction Engine makes use of Family malware signatures Reduction Buffer Database to as a temporary storage to help form a malware group signature from individual malware signatures.

When an executable file is sent to the *antivirus engine* for malware analysis, the executable file is attempted to be unpacked (if it is packed), disassembled and compared if it has signatures in the Scalable Malware Signature Database. The signatures that are being compared with are the individual malware signatures and signatures for families. If there is a match with any of the signatures, the antivirus engine declares the executable file as malware and otherwise, it declares it as benign.

## IV. THE SCALABLE MALWARE SIGNATURE DATABASE SYSTEM

Figure 2, shows the scalable malware signature database system. However, the database system consists of the following:

i) $S_1, S_2, S_3, \ldots S_n$ are the individual malware signatures generated and updated by the antivirus experts, periodically. $S_i$ is an arbitrary malware signature, where $i$ is the signature number and $n$ is the current maximum of signatures.

ii) $F_1, F_2, F_3, \ldots, F_m$ are set of patterns of malware signatures representing Family malware signatures.

In Figure 2, the individual malware signatures $S_1, S_2, S_3, \ldots S_n$ are of different patterns hence, that is why they are numbered from *1, 2, 3, ..., n*. The family malware signatures $F_1, F_2, F_3, \ldots, F_m$, have different patterns to represent a groups of signatures, and that is why the shapes are looking different to signifies different patterns. The behaviour of model in Figure 2, starts with antivirus experts continuously updating the scalable malware signatures database system with newly found malware signatures, using the *malware features updating pool*. The antivirus expert also tries to form new family malware signatures and update them in the scalable malware signatures database system. A family malware signature is a pattern of signatures, used to identify more than one malware. The duty of malware family reduction engine is to automatically search the individual malware signatures for the ones who have similar signatures, which can be grouped into family malware signatures.

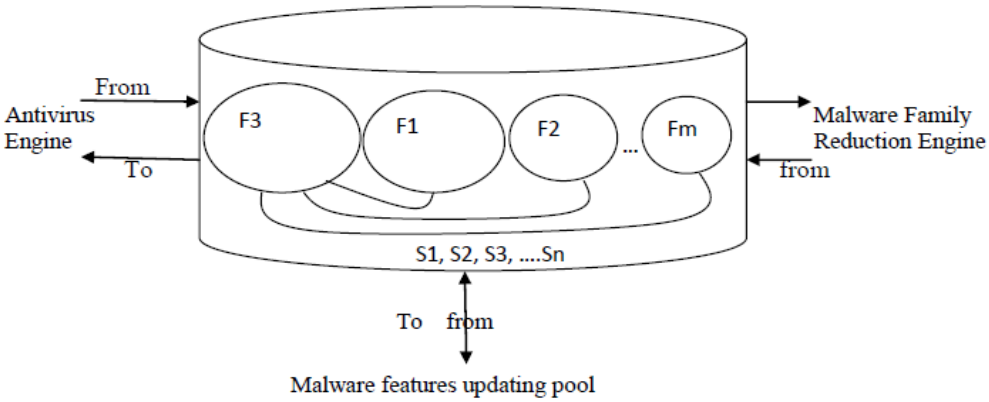


**Fig 2: Scalable Malware Signatures Database System**

The antivirus engine accepts an executable file, unpacks it, disassembles it and attempts to see if there is a match of the signatures in Scalable Malware Signatures Database System, with a pattern in the file. The first search for signatures is in family malware signatures and when there is not match, the individual malware signature is searched. When an individual malware signature is discovered to have a pattern belonging to a family malware signature, the individual malware signature is deleted from the Scalable Malware Signatures Database System. Whenever an individual malware signature is deleted from the Scalable Malware Signatures Database System because, the size of the Scalable Malware Signatures

Database System reduces continuously. Apart from the size of the Scalable Malware Signatures Database System being continuously reduced, due to the inclusion of individual malware signatures to a malware signatures family, searching for a match between a pattern in an executable file and malware signature in the Scalable Malware Signatures Database System, reduces search time.

## V. CONCLUSION

In this paper, attempts were made to form groups of family malware signatures from individual malware signatures. After each group is formed, the individual malware signatures that form the group signature are discarded. So, in the Scalable Malware Signatures Database System, there is a continuous reduction of the individual malware signatures, due to the discarding of the ones that is now part of the signature group. The first advantage of the proposed antivirus system is in the time reduction of malware signatures search in database against executable file patterns. The second advantage is in the gradual reduction of individual malware signatures, which were discarded because, they were found to have similar finger patterns related to existing group of family malware signatures. The gradual reduction of the number of individual malware signatures into theirs group of family malware signatures, reduces the total number of malware signature in the signature database, thereby, makes the antivirus database scalable.

## REFERENCES

[1] G. H. Leela, T. D. Dongale, and R. K. Kamat, "Techniques Adopted for Containment of Polymorphic Worm- A Review", International Journal of Electronics Communication and Computer Technology, Vol. 2, Issue 6, pp. 300-303, 2012.

[2] B. B. Rad, M. Masrom, and S. Ibrahim, "Evolution of Computer Virus Concealment and Anti-Virus Techniques:" A Short Survey, International Journal of Computer Science Issues, Vol. 8, Issues 1, pp. 113-121, 2011.

[3] E. D. Daoud, I. H. Jebril, and B. Zaqaibeh, "Computer Virus Strategies and Detection Methods", International Journal of Open Problems in Computer and Mathematics, Vol. 1, No. 2, pp. 122-129, 2008.

[4] I. Y. Yoo, and U. Ultes-Nitsche, "Non-Signature based virus Detection", Springer-Verlag, France, 2006.

[5] V. S. Sathyanarayan, P. Kohli and B. Bruhadeshwar, "Signature Generation and Detection of Malware Families", Springer-Verlag, Belin, Heidelberg, pp. 336-349, 2008.

[6] S. Venkatachalan, "Detecting Undetectable Computer Viruses", MSc Thesis, The Faculty of the Department of Computer Science, San Jose State University, USA, 2010.

[7] A. Mujumdar, G. Masiwal, and B. B. Meshram, "Analysis of Signature-Based and Behaviour-Based Anti-Malware Approaches", International Journal of Advanced Research in Computer Engineering and Technology, Vol. 2, Issue 6, pp. 2037-2039, 2013.

[8] K. M. Alzarooni, "Malware variant Detection", PhD Thesis, University College, London, United Kingdom, 2012.

[9] R. Choudhary and R. Saharan, "Feature Detection Approach from Virus Through Mining", Journal of Global Research in Computer Science, Vol. 2, No. 6, pp. 21-22, 2011.

[10] N. K. Dixit, L. Mishra, M.S. Charan, and B. K. Dey, "The New Age of Computer Virus and Their Detection", International Journal of Network Security and Its Applications, Vol. 4, No. 3, pp. 79-96, 2012.

[11] A. Berkat, "Metamorphic Computer Virus Detection by Case-Based Reasoning Methods", International Journal of Software Engineering and Applications, Vol. 2, No. 4, pp. 1-10, 2011.

[12] A. Altaher, S. Ramadass and A. Ali, "Computer Virus Detection Using Features Raking and Machine Learning", Australian Journal of Basic and Applied Sciences, Vol. 5, No. 9, pp. 1482-1486, 2011.

[13] N. N. Phalke, S. Shamrao, A. Priyadarshi and V. B. Shinde, "Malware Detection Using N-GRAM Base File Signature Based Methods", International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 2, Issue 11, pp. 3793-3795, 2014.

[14] H. A. Ali and D. J. Hussain, "Computer Virus Detection Based on Artificial Immunity Concert", International Journal of Emerging Trend and Technology in Computer Science, Vol. 3, Issue 2, pages 68-74, 2014.