A review of polymorphic malware detection techniques

Joma Alrzini[1], Diane Pennington[2]
[1]University of Strathclyde, UK, joma.alrzini@strath.ac.uk
[2]University of Strathclyde, UK, diane.pennington@strath.ac.uk

ABSTRACT

Despite the continuous updating of anti- detection systems for malicious programs (malware), malware has moved to an abnormal threat level; it is being generated and spread faster than before. One of the most serious challenges faced by anti-detection malware programs is an automatic mutation in the code; this is called polymorphic malware via the polymorphic engine. In this case, it is difficult to block the impact of signature-based detection. Hence new techniques have to be used in order to analyse modern malware. One of these techniques is machine learning algorithms in a virtual machine (VM) that can run the packed malicious file and analyse it dynamically through automated testing of the code. Moreover, recent research used image processing techniques with deep learning framework as a hybrid method with two analysis types and extracting a feature engineering approach in the analysis process in order to detect polymorphic malware efficiently. This paper presents a brief review of the latest applied techniques against this type of malware with more focus on the machine learning method for analysing and detecting polymorphic malware. It will discuss briefly the merits and demerits of it.

Key words: anti-detection; polymorphic automated testing; abnormal threats; packed malicious file.

1. INTRODUCTION

Malicious software has become the most serious threat to the security of computational devices as a result of impressive growth in developing software application on these devices. Malware is the largest threat for all these applications; it harms the process of a user's activity and damages a user's device, whilst stealing information. In the second quarter of 2016, the number of unique web malicious antivirus detected in a web application by Kaspersky was more than 16,000,000, and the number of attempts to steal money via online banking was clocked on more than 1,000,000 computers. Furthermore, according to the AV test institution, there are 390,000 registered Attempts every day. In addition, a report from Kaspersky's lab shows that financial malware activity is increased by 15.6% higher than the first quarter of 2016 [3]. Furthermore, a prediction of cybercrime report estimated annual world cost of $6 trillion by 2021[2]. Based on state of malware report 2020, the detection threat is decreased by 2% over 2018, but in the detection increased by 13% in 2019, which means 1% increase from year to another [28]. These numbers imply the size of threats from malware and the importance of detecting and preventing these threats. However, these are detected by the regular approach of anti-virus detection programs based on the signature of malware; the obfuscation techniques of them are not considered. Moreover, the use of obfuscation techniques makes malware evade the traditional detection method.

The rest of the paper is organized as follows: an overview of polymorphic malware and its effectiveness, malware classification, the data mining detection method, string the signature algorithm, using sandbox analysis, machine learning algorithms, deep learning framework based on a hybrid malware analysis method, and feature engineering approach.

2. AN OVERVIEW OF POLYMORPHIC MALWARE

The first emergence of polymorphic malware occurred in 1990. It comes in several structures to create malicious code by using polymorphic engines. This type of malware is less likely to be detected by an antivirus application. The most commonly used techniques for writing polymorphic malware are encryption/decryption and data appending. These techniques use code obfuscations to evade antivirus scanners. Thus, an effective method has to be used to detect unknown malware with obfuscation; the machine learning method is now the most effective approach, particularly for abnormal malware [29].

The impact of polymorphic malware on software applications is more than that of normal malicious software that can be detected by anti-virus software. The first to emerge was able to change and decrypt itself; however, it generated a few malicious threats that cannot be detected with signature-based systems. Also, malware authors developed a countless number of malicious attempts in various ways every day. These developments have been achieved with the aid of obfuscation code and other methods such as code insertion. These writers use polymorphic toolkits such as mutating engine and polymorphic packer (which are called polymorphism engines) to change the non-obfuscation malware to polymorphic [1]. The malware created by this technique is detected after it has infected the victim machine. Hence, it is most likely to achieve its goal before it is detected.

Moreover, with the analysis techniques the polymorphic malware can escape from detection method. Even though with hybrid technique of two analysis method, the malicious program changes its behavior and with this method needs time to be detected as static and dynamic and dynamic analysis combined as hybrid method [27].

3. POLYMORPHIC MALWARE CLASSIFICATION

The meaning of classification herein is to test the malware as to whether or not it is polymorphic. According to Selamat and Othman [6], the dropped file detection framework can identify the malware into polymorphic and not polymorphic through this detection system. Their experiment framework, as depicted in Fig 1, executed the malicious file more than once and checked whether the malware generated a different file each time. This would mean that the malware changes its code in each execution, so it is thus polymorphic malware. On the other hand, if the malware generated the same file, this means it is not polymorphic malware. The researchers proposed this framework as the first step in developing a prevention system for polymorphic malware; this is the suggestion for future work [6].

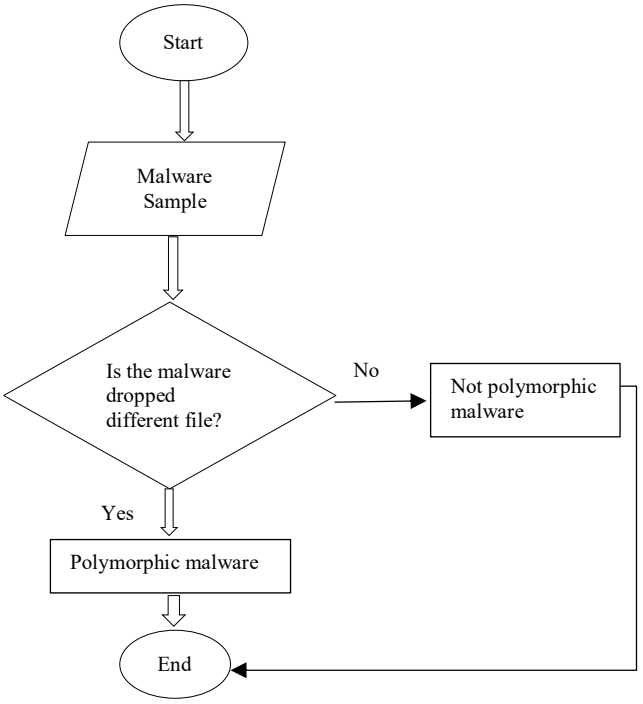

Fig1. *Flowchart of dropped file detection framework.*

4. DATA MINING DETECTION METHOD

Alazab et al. [4] proposed a new framework using data mining techniques with eight classifiers being applied to a large dataset. The results showed approximately 98% accuracy in detecting malware with automated data mining techniques. The experimental analysis of this study examined the performance of algorithms used in analysing the behaviour of malicious codes. Furthermore, in this study, Windows API calls and statistical measures were used to obtain more accurate results from the classification algorithms. The methodology used in this study is the zero-day malware detection methodology, which consists of three process phases. The first phase involves three steps: disassemble binary executables from packed files, then extract API calls from the assembly program, before finally analysing the behaviour of malicious files to get API sequence from the disassembled binaries. The second phase is concerned with updating the signatures in a database to generate the similarity report. In the third phase, relevant features are selected by using mutual information and the most relevance based on a set of Windows API functions [4].

In data mining techniques used for malware detection, many algorithms can be used, but the authors of malware attempt to evade these techniques. Thus, there is a need to evolve the use of these algorithms to build a robust detection system for this development in malware code [30]. However, using data mining techniques to detect and prevent malicious software can lead to some security problems, such as interference and privacy [7].

5. STRING SEARCHING ALGORITHMS

String algorithms examined the impact of techniques based on dynamic programming to provide better automatic detection of unknown polymorphic variants. This research utilised Needleman-Wunsch and Smith-Waterman's algorithms [5]. In addition, the string-based approach has been used in another research to identify the known polymorphic variants in the JS. Cassandra virus. This approach applies structural identification to determine whether or not code contains a malicious variant and to identify to which family that code can be classified. According to the signature extraction approach, one signature may capture all variants and variants may need more signatures to capture them. This means the patterns may be matched in changing the code arrangement by a similarity check. Moreover, the IS. Cassandra polymorphic variants are rewritten in 'String.formCharCode' (JavaScript function); this is very difficult to detect with reverse engineering techniques due to the encryption or protection of JavaScript code. The results show that all known variants of polymorphic malware were successfully detected, but there is a need to develop efficient software for new polymorphic malware using a syntactic approach [5]. Ojugo and Eboka [31] used Boyer Moore's algorithm as string matching algorithm for malware detection. This algorithm does liner research of the string by scanning it to get the matching character of string, then the non-matching character will be considered first for scan. This approach required knowing the string length and how the position of character to be counted.

6. MALWARE SANDBOX ANALYSIS

This involves the mechanism for executing the malware files to identify their behaviour; it is a virtual environment to execute the file without having any effect on the operating system. The program runs the malware file whilst analysing its behaviour. According to Messmer [9], this is one of the alternatives to signature-based detection. The behavioural observation has to be used with sophisticated code in order to reveal the new malware effectively [8]. Moreover, this is one of the virtualisation toolkits that keeps the actual system safe and observe the malware execution. However, malware authors can check the code to uncover the virtual environment; this is done by checking the Internet access. Osorio et al [10] proposed a novel approach based on segmented sandboxing to overcome the limitation of the sandbox method. The researchers combined static malware detection with segmented sandboxing; this is called a mixed approach. They designed the model based on traditional automated theory to formulate and implement a practical solution. Their results show that the approach is quite functional in minimising false negatives and false positives. Additionally, the experiment resulted in different types of malware showing the detection rate of 100% as highest and 82% as the lowest. These results indicate the robustness of combining static and dynamic approaches [10].

The above reviews of the sandbox method lead to the understanding that relying on one analysing method cannot give an accurate result for malware analysis frameworks. Thus, static analysis is still needed, even in polymorphic malware in order to obtain the most robust results. The sandbox method has some deficiencies: the efficacy is potential but there is no guarantee of showing all threats, the malware writers can evaluate and check the detection accuracy for the sandbox analysis by using evasion techniques, and also it is not the endpoint but it leads to the end [10].

7. MALWARE DETECTION USING MACHINE LEARNING

The use of machine learning algorithms is still in the early stages with some experiments being conducted using this method. This method has been employed since the pattern-matching approach failed to develop new malware due to the use of obfuscation techniques. Gavriluţ et al [11] proposed a detection framework utilising different algorithms. This study aimed to obtain a zero-false positive rate, but it had a few false positives. However, the running time was short for the large dataset and the malware samples used were not detected by the standard detection method used by anti-virus systems [11].

The hybrid clustering method was used to classify the malware into polymorphic and metamorphic by KNN machine learning algorithm. This experiment obtained 97% accuracy from the classifier algorithm [12]. Nevertheless, this algorithm does not detect malware; it only categorises it. This can be considered to be the first step in analysing malware, and then the appropriate method has to be applied to the categorised malware.

Another detection approach for detecting unknown malware is the Intelligent Malware Detection System (IMDS). It was employed to test 3000 malicious samples of polymorphic malware as a part of the whole experiment for Windows API executables. The researchers employed different object-oriented algorithms. Comparing the obtained result from this sample with the signature-based anti-virus software; the proposed system achieved greater accuracy in detecting polymorphic malware (unknown malware) [14]. Table 1 demonstrates the comparison accuracy obtained by this method with anti-virus systems.

Table 1: Unknown malware detection

Norton AntiVirus (N), MacAfee (M), Dr.Web (D) Kaspersky (K).

Software	N	M	D	K	IMDS
malware1	√	√	√	×	√
malware2	×	×	√	√	√
malware3	√	×	×	×	√
malware4	×	×	×	×	×
malware5	×	√	×	×	√
malware6	×	×	√	×	√
malware7	√	×	×	×	√
malware8	×	×	×	×	√
malware9	×	√	√	×	√
malware10	×	×	×	×	√
malware11	×	×	×	√	√
malware12	×	√	×	√	√
⋮	⋮	⋮	⋮	⋮	⋮
malware1000	√	×	×	×	√
Stat.	633	753	620	780	920
Ratio.	63.3%	75.3%	62%	78%	92%

The use of machine learning algorithms is still an early age particularly for this type of malware. For android malware, this approach is also being used to detect malware with obfuscation techniques such as polymorphic.

In 2013 Cesare and Xiang proposed a classification method for polymorphic malware called Malwise that uses the application-level emulation to unpack malware code. Furthermore, the classification was based on two flow graph algorithms. This emulation application gives an alternative approach for unpacking malware automatically; it requires the execution of the file. To simulate the packing malware, the execution is required, so that the unpacking method can detect the malware's hidden code and extract it from the process. This proposed method helps to simulate the architecture of the system and the call interface, which is the Windows API in the Windows operating system. Static analysis is used before the classification to extract attributes from the input binary, which is then used to check the matched signature. The control flow graph is built to generate that which will be compared with approximate matching. The speed of automatic classification was very high compared to the previous study, which is less than a millisecond for repeating one classification algorithm for the database containing 70,000 malwares. Moreover, the evacuation of this system showed it has high levels of effectiveness in identifying variants of real malware; a new malware is highly likely to be a variant of the current malware. Additionally, the researchers noticed that the efficiency of the Malwise system makes it appropriate for antivirus systems and other applications [17].

Boske [18] presented the implemented methods for polymorphic malware identification and its variants. The researcher described the process of the method of identification using a set of filters, determining incorrectly excluded samples and modifying the method to include these executed samples. The filters of the system are updated automatically to ensure that a new variant of polymorphic malware can be handled currently; this also ensures that it is not executed from being classified. The number of filters depends on the property of the executable file and the variety of filters that may be applied to the polymorphic sample, such as a dispositive scan. This scan for encryption decrypts a part of an executable file then identifies the signature that matches the polymorphic variant.

The operating systems of smartphones are also subject to malware attacks through their applications. Hence the detection of malware on these applications has become one of the hot topics attracting the attention of researchers. The number of new threats increases due to the growing number of mobile apps. For example, in 2015, more than 430 million new malwares were discovered. This is 36% more than those discovered in 2014 (Internet Security Threat Report). The recent detection method for androids uses machine learning algorithms. For instance, in 2016, ATICI and others proposed a static malware analysis approach based on control flow graphs and machine learning algorithms; they obtained a 99.15% detection rate for Droidkungfu malware. Moreover, Khan [23] proposed a method based on machine learning for android malware. This approach used the Androguard tool to extract features from applications; it could then use these features as training for one-class Support Vector Machine within the Scikt-learn framework. The result showed a low false-negative rate [22].

To evaluate the performance of machine learning algorithms (MLAs) in detecting malwares, recent research done by Vinayakumar et al. [25] exhibited that this method with static and dynamic analysis of malware behaviour takes time and is an unsuccessful approach particularly in a variation of a malware. Therefore, this research proposed deep learning as an advanced machine learning algorithm; this method utilised the image processing technique with two types of analysis. The researchers applied a new framework called ScaleMalNet which deployed deep learning on gathering malwares from end-users followed up by an analysing stage with two processing phases. The first step is classifying malwares with dynamic and static analysis technique, and the second is grouping malwares into the identical malware classes via image-processing. This framework is efficient for analyzing a large number of malwares in real-time. This was done by adding more layers to the deep learning architectures to enhance the detection model [26]. As a result, the hybrid deep learning framework outperforms traditional MLAs in detecting malwares.

Converting a malware to image techniques have been developed with classification to detect its binaries in represented colour. The research was done on this to evaluate Convolutional Neural Network Features and has concluded that CNN is a practical approach in malware classification as similar samples are detected when malwares are converted to image format [24].

Masabo 2019 proposed a new classification method using a machine learning approach called Novel Feature Engineering (NFE). This method classifies and detect polymorphic malware based on feature and behavior of it. three steps of classification are utilised in this approach (KNN, Liner analysis and Gradient Boosting machine). The classification of malware with this method achieved 94% accuracy. [33].

8. STRUCTURAL FEATURE ENGINEERING METHOD

Identifying the features and behaviour of malware is the management of the dete [Yes] approach. The most recent research mainly applied static and dynamic analysis methods, however, for polymorphic malwares, it is hard to detect it as it is obfuscated. A feature engineering method has been proposed to identify polymorphic malwares in statistical analysis. This method extracts features appearing in the analysis process and then transfers them to the feature selector algorithm. This method has achieved 98.7% accuracy in detecting polymorphic malwares with a small dataset. The process of feature extraction is explained in Figure 2, which starts with checking the packing state of malicious samples. If it is packed, it extracts it analyses it statistically, and then extracts features before transforming them through an algorithm to finally classify a feature set with a detection model. The limitation of this approach is the size of the utilised dataset and fewer classification algorithms which are suggestions for future research [32].

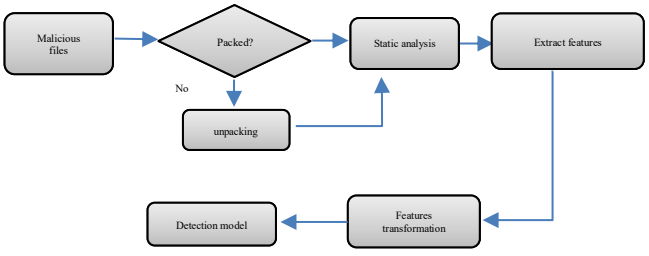

Figure 2. Feature extraction malware detection process

9. DISCUSSION AND CONCLUSION

This paper highlights the most recently applied methods for detecting polymorphic malware. The countermeasure method is needed to detect unknown malware. The availability of malware variants software toolkits puts the malware authors ahead of the development in anti-malware systems. Therefore, there is no guarantee of having a method with zero attacks and 100% accuracy, but the development of detection systems has to be persistent. Future trends are moving towards deploying machine learning algorithms in this subject due to their efficiency and speed of analysing the code. The reviewed studies indicate that:

- Polymorphic malware comes in forms such as Trojans, worms or viruses, but it changes its appearance with code propagation, encrypts and decrypt. Hence signature-based method cannot ensure their detection.
- The data mining approach gives a high accuracy rate with different machine learning classifiers. This approach is based on Windows API calls and this leads to the exploration of various obfuscations aspects in the future.
- The dropped file approach and the string algorithms effectively classify unknown malware. The Needleman-Wunsch and Smith-Waterman algorithms successfully detected polymorphic variants of JS.Cassandra malware; they needed to be applied to other variants.
- The sandbox analysis method provides a dynamic analysis tool, but it cannot be sufficient because the malware code can detect the virtualisation framework by checking Internet access. However, with the combination of static analysis, it can give an accurate high detection rate.
- Several machine learning algorithms have been utilised and they accomplish a high level of accuracy in classifying polymorphic malware. However, detection can be evaded by the novel approach of creating malware.
- The hybrid deep learning framework has used image processing with dynamic and static analysis by transforming a malware image in a previously determined sized image. However, the gap in this study is allowing any size of the inputted image by using a spatial pyramid pooling layer (SPP) according to the researchers' recommendation.
- The engineering approach helps in building structural analysis processing to develop an efficient malware detection framework. This approach uses static analysis by extracting features of polymorphic malwares.
- The presence of current detection techniques is not sufficient for detecting a wide variety of new malware occurring every day.

REFERENCES

1. B.B. Rad, M, Masrom, and S. Ibrahim. **Camouflage in malware: from encryption to metamorphism** *International Journal of Computer Science and Network Security*, 12(8), pp.74-83,2012.
2. Cybersecurity Ventures, 2017 cybercrime report, 2017.
3. Emm, D. Unuchek, R., Garnaeva, M., Ivanov, A., Makrushin, D. and Sinitsyn, F. **Its threat evolution in Q2** *Kaspersky Lab HQ*, 2016.
4. Alazab, Mamoun, et al. **Zero-day malware detection based on supervised learning algorithms of API call**

signatures. *Proceedings of the Ninth Australasian Data Mining Conference-Volume 121*. Australian Computer Society, Inc., 2011.

5. V. Naidu and A. Narayanan. **Using different substitution matrices in a string-matching technique for identifying viral polymorphic malware variants**, *IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, 2016, pp. 2903-2910. 2016.
6. N. S. Selamat, F. H. Mohd Ali and N. A. Abu Othman. **Polymorphic Malware Detection**. *6th International Conference on IT Convergence and Security (ICITCS)*, Prague, 2016, pp. 1-5. 2016.
7. M. Masud, L. Khan, and B. Thuraisingham. **Data mining tools for malware detection**. 2011 CRC Press.
8. D. Keragala. **Detecting malware and sandbox evasion techniques.** SANS Institute InfoSec Reading Room. 2016.
9. Messmer, Ellen. **Malware-detecting 'sandboxing' technology no silver bullet.** Networkworld, March 2013. Retrieved from http://www.networkworld.com/article/2164758/network-security/malware-deteting--sandboxing--technology-no-silver-bullet.html
10. F. C. Colon Osorio, H. Qiu and A. Arrott. **Segmented sandboxing - A novel approach to Malware polymorphism detection**, *10th International Conference on Malicious and Unwanted Software (MALWARE)*, Fajardo, 2015, pp. 59-68. 2015.
11. D. Gavriluţ, M. Cimpoeşu, D. Anton and L. Ciortuz. **Malware detection using machine learning**, *International Multiconference on Computer Science and Information Technology*, Mragowo, 2009, pp 735-741.
12. P. Sharma, S. Kaur, and J. Arora. **An Advanced Approach to Polymorphic/Metamorphic Malware Detection using Hybrid Clustering Approach** (IRJET) 2016.
13. R. Kaur and M. Singh, **A Survey on Zero-Day Polymorphic Worm Detection Techniques**. *IEEE Communications Surveys & Tutorials* vol. 16, no. 3, pp. 1520-1549, Third Quarter 2014.
14. Y. Yanfang, D. Wang, T. Li, and D. Ye. IMDS: **intelligent malware detection system**. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '07). 2007 ACM, New York, NY, USA, 1043-1047.
15. A. H. Sung, J. Xu, P. Chavez and S. Mukkamala. **Static analyzer of vicious executables (SAVE),** *20th Annual Computer Security Applications Conference*, 2004, pp. 326-334.
16. S. Cesare, Y. Xiang and W. Zhou, **Malwise&# x2014; An Effective and Efficient Classification System for Packed and Polymorphic Malware.** *IEEE Transactions on Computers*, vol. 62, no. 6, pp. 1193-1206, June 2013.
17. A. Boske, Symantec Corporation, 2013. **Systems and methods for identifying polymorphic malware,** 2013 U.S. Patent 8,479,291.
18. M. A. Atici, S. Sagiroglu and I. A. Dogru. **Android malware analysis approach based on control flow graphs and machine learning algorithms**. *4th International Symposium on Digital Forensic and Security (ISDFS)*, 2016, Little Rock, AR, 2016, pp. 26-31.
19. A. Sharma, and Sahay S.K., **Evolution and detection of polymorphic and metamorphic malwares: a survey**, 2014 arXiv preprint arXiv:1406.7061.
20. Wu, Songyang, Pan Wang, Xun Li, and Yong Zhang. **Effective detection of android malware based on the usage of data flow APIs and machine learning**. Information and Software Technology 75 2016, pp 17-25.
21. H. Razeghi Borojerdi and M. Abadi. **MalHunter: Automatic generation of multiple behavioral signatures for polymorphic malware detection**, *ICCKE 2013*, Mashhad, 2013, pp. 430-436.
22. J. Sahs and L. Khan, **A Machine Learning Approach to Android Malware Detection**, *European Intelligence and Security Informatics Conference*, Odense, 2012, pp. 141-147.
23. N. Peiravian and X. Zhu. **Machine Learning for Android Malware Detection Using Permission and API Calls**. *IEEE 25th International Conference on Tools with Artificial Intelligence*, Herndon VA, 2013, pp.300-305.
24. Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P. and Venkatraman, S., 2019. **Robust Intelligent Malware Detection Using Deep Learning.** *IEEE Access*, 7, pp.46717-46738.
25. Özkan, K., Işık, Ş. and Kartal, Y., 2018, March. **Evaluation of convolutional neural network features for malware detection.** *6th International Symposium on Digital Forensic and Security (ISDFS. 2018* (pp. 1-5). IEEE
26. Masabo, E., Kaawaase, K.S., Sansa-Otim, J. and Hanyurwimfura, D., 2017 November. **Structural Feature Engineering approach for detecting polymorphic malware**. *IEEE 15th Intl Conf on Dependable Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*. 2017 (pp. 716-721). IEEE.
27. Y. K. B. M. Yunus and S. B. Ngah, **Review of Hybrid Analysis Technique for Malware Detection, in *IOP*** *Conference Series: Materials Science and Engineering*, 2020, vol. 769, no. 1: IOP Publishing, p. 012075.
28. State of malware report 2020, retrieved from https://resources.malwarebytes.com/files/2020/02/2020_State-of-Malware-Report.pdf
29. Tajoddin, Asghar, and Saeed Jalili, **HM3alD: Polymorphic Malware Detection Using Program Behavior-Aware Hidden Markov Model**. *Applied Sciences 8*, no. 7 (2018): 1044.
30. Souri, A. and Hosseini, R., 2018. **A state-of-the-art survey of malware detection approaches using data mining techniques.** *Human-centric Computing and Information Sciences*, 8(1), p.3.
31. Ojugo, A. and Eboka, A.O., 2019. **Signature-based malware detection using approximate Boyer Moore string matching algorithm**. *International Journal of Mathematical Sciences and Computing*, 5(3), pp.49-62.
32. Masabo, E., Kaawaase, K.S., Sansa-Otim, J. and Hanyurwimfura, D., 2017, November. **Structural Feature Engineering approach for detecting polymorphic malware.** *In 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and TechnologyCongres(DASC/PiCom/DataCom/CyberSciTech) (pp. 716-721)*. IEEE.
33. Masabo, E., 2019. **A Feature Engineering Approach for Classification and Detection of Polymorphic Malware**

using Machine Learning *(Doctoral dissertation, Makerere University).*