The Technique for Metamorphic Viruses' Detection Based on its Obfuscation Features Analysis

George Markowsky¹, Oleg Savenko^{2[0000-0002-4104-745X]}, Sergii Lysenko^{2[0000-0001-7243-8747]} and Andrii Nicheporuk^{2[0000-0002-7230-9475]}

> ¹ Missouri University of Science and Technology, Rolla, USA ¹ markov@maine.edu ² Khmelnitsky National University, Khmelnitsky, Ukraine {savenko_oleg_st@ukr.net, sirogyk@ukr.net, andrey.nicheporuk@gmail.com }

Abstract. The paper presents an approach for the metamorphic viruses detection based on its obfuscation features analysis. The obfuscation features were obtained on the basis of the equivalent functional block search in the suspicious program and its modified version. The results of the research demonstrated that the efficiency of metamorphic viruses detection based on the proposed obfuscation quantitative features depends on the choice of the similarity metric at the stages of the search and the choice refinement of the equivalent functional blocks. The adequate choice of similarity metrics at both stages allowed increasing the detection efficiency of the metamorphic viruses.

Keywords: Metamorphic Virus, Obfuscation, Equivalent Functional Block, Opcode, Distance metrics

1 Introduction

Today the problem of cyber security is very important. The malware detection is a very important task because of the information pilfering, anonymous access to network, spy actions, spamming are observed. The most hard to detect type of the malware is metamorphic viruses [1]. The main feature that distinguishes the metamorphic viruses from other types of malware is the usage of the code obfuscation techniques. Applying of the code obfuscation allows the syntax modification of the program code, however the semantic base of the virus algorithm functioning is unchanged. Taking into account the mentioned feature of metamorphic viruses, the creation of a virus code signature is impossible [2]. The way to detect the metamorphic viruses efficiently the heuristic techniques are used.

2 Related Works

Today solutions to the problem of the metamorphic viruses' detection are widely presented in the literature.

An article [3] presents the main 189 obfuscation features of the metamorphic viruses, extracted from the all sections of the executable file. In [4] the authors are focused only on the section attributes, and to draw the conclusion about metamorphic viruses presence a set of rules for the files' sections analysis was proposed.

Another line of research of the metamorphic viruses' detection is the opcodes analysis [5-8]. In [5] for the inferencing about metamorphic viruses' presence a weighted directed graph based on the program's opcodes is constructed and a comparison of the built graph with the graphs of known metamorphic viruses using the similarity metrics is carried out.

In [6] in order to detect the metamorphic viruses the histograms of the opcode appearance are built. It makes it possible to find the similarity between the known histogram of metamorphic viruses and such histogram of the suspicious program.

Another approaches are based on the analysis of the opcodes, that describe the key features of the metamorphic viruses. Thus, in [7] the authors made an attempt to find the most commonly used opcodes for metamorphic viruses, while in [8] the authors assert that the code is compiled by the compiler differs from the mutated code.

To process the received characteristic features and to make decision about the of a metamorphic viruses presence, the machine learning methods, such as J48, multi-agent approaches, neural networks, design tree, k-nearest neighbors algorithm and Naive Bayes [5-10] are involved. The main drawback of mentioned techniques is high rate of the false positives.

An alternative approach for the metamorphic viruses detection is the features extraction, based on the control flow analysis. In [11] the detection process involves a graph construction of the metamorphic virus control flow, and the transformation into the vector space. The classification of the formed vector is carried out by known classifiers [5-8]. Another approach for the metamorphic viruses' detection is the monitoring of the API calls, used by programs [12]. In the work, the authors proposed a static approach to form the signature of the metamorphic viruses, based on the calculation of the API calls number corresponding to known metamorphic viruses. The conclusion about the metamorphic virus presence is made on the basis of searching of the similarity between the formed signature and the signatures in the base of known viruses using the similarity metrics [12].

The main drawbacks of the mentioned techniques are low level of the detection, the complexity and high rate of false positives. The way to eliminate these disadvantages is involvement of the obfuscation features analysis, based on the assessment of the quantitative characteristics of the metamorphic viruses' code modification.

3 The Technique for Metamorphic Viruses' Detection Based on its Obfuscation Features Analysis

The paper [2] presents an approach for the metamorphic viruses detection in the local networks. It is based on the analysis of the potentially suspicious behavior of the programs on the host, and on the analysis of the obfuscation features, which are demonstrated during the program's functioning. The obfuscation features were obtained on

the basis of the equivalent functional block search in the suspicious program and its modified version. The conclusion about the metamorphic viruses' presence is drawn by the fuzzy inference system placed on the server, using the gathered information from all hosts of the network.

The technique involves the following stages: a data preprocessing; a localization of a place for the search of the equivalent function blocks (EFBs) in the executable; a search of the EFBs, the choice refinement of the EFBs; the obtaining of the quantitative features of the code obfuscation, based on the comparison of EFBs of suspicious program and its modified version; a drawing the conclusion about metamorphic viruses presence (Fig. 1). Let us take a closer look at each step of the technique.



Fig. 1. The technique overview

3.1 Data Preprocessing.

In order to detect metamorphic viruses the main stage is the obfuscation features extraction. We will deal with quantitative features of the code obfuscation, which will allow estimating the difference between two copies of the metamorphic virus. For this purpose a disassembly of the suspicious program and the obtaining of a code sample before the emulation are performed. In order to create the modified version of the suspicious program it is launched in the emulator. A result of the data preprocessing stage are two opcodes lists of the suspicious program and its modified version.

3.2 The Localization of a Place for the Search of the Equivalent Function Blocks in the Executable.

The features extraction of the metamorphic viruses requires the localization of a place for the search of the EFBs in the executable. For this purpose the definition of the program's entry point and the choice of the section where it is located are carried out. We will choose the localization of a place for the search of the EFBs in such cases: if there are call or jump instructions near the program's entry point and the operands contain the address of another section; there is a section with non-standard name; there is an executable access attribute of the section.

3.3 Search of the Equivalent Functional Blocks.

After the place of the search for the EFBs of the suspicious program and its modified version is determined, the next stage involves the partitioning of the opcode listings into the blocks. The process of the partitioning of the opcode listings into the functional blocks (FBs) involves the disassembly of the suspicious program and its modified version, and the separation of the opcode sequences, arranged between the commands of conditional or unconditional jumps.

Let us denote the suspicious program as F_b and its modified version F_a as the sets $F_b = \{F_{b_1}, F_{b_2}, ..., F_{b_m}\}$ and $F_a = \{F_{a_1}, F_{a_2}, ..., F_{a_n}\}$ accordingly. To present the obtained FBs as a statistical model, the rating matrix of the opcodes appearance for each FBs was construct. For this purpose, to each functional block the Term Frequency–Inverse Document Frequency (TF-IDF) statistical metric was applied:

$$s = \frac{n_i}{\sum_k n_i} * \log(\frac{N+1.0}{n_j}) \tag{1}$$

where n_i – a number of *i*-th appearance of the opcode in the functional block *F*; $k = \overline{1, k_a}$ – a number of opcodes in the functional block, where k_a – a total number of the assembler instructions; n_j – a number of functional blocks with some opcode n_i ; *N* – a total number of the functional blocks.

Then, the rating matrices of the opcodes appearance in the FBs of the suspicious program F_b and its modified version F_a can be presented as follows:

$$F_{a} = \begin{pmatrix} F_{a_{1}} \\ F_{a_{2}} \\ \cdots \\ F_{a_{n}} \end{pmatrix} \Rightarrow \begin{pmatrix} s_{11} & s_{21} & \cdots & s_{k1} \\ s_{12} & s_{22} & \cdots & s_{k2} \\ \cdots & \cdots & \cdots & \cdots \\ s_{1n} & s_{2n} & \cdots & s_{kn} \end{pmatrix} \quad F_{b} = \begin{pmatrix} F_{b_{1}} \\ F_{b_{2}} \\ \cdots \\ F_{b_{m}} \end{pmatrix} \Rightarrow \begin{pmatrix} s_{11} & s_{21} & \cdots & s_{g1} \\ s_{12} & s_{22} & \cdots & s_{g2} \\ \cdots & \cdots & \cdots & \cdots \\ s_{1m} & s_{2m} & \cdots & s_{gm} \end{pmatrix} \quad (2)$$

Thus, as a result the two sets of FBs are obtained, which are represented as the rating matrices of the opcodes appearance in function blocks for the suspicious program and its modified version. The search of the EFBs involves a pairwise comparison of each FB from the set F_b with each FB from F_a using the similarity metric. The result of the comparison the similarity score E_r between the two FB is obtained. If the value of the similarity score E_r for two FBs is less than the specified threshold δ , then the recalculation of similarity score between the functional block of the program and the next block that follows the block is performed. The above actions are repeated until the value of the similarity score is less or equal to the threshold δ . The threshold value δ is determined experimentally.

Thus, at the stage of determination of the equivalent functional blocks, there may be a situation when the several function blocks may correspond to one FBs $F_{b_i} \sim \{F_{a_i} | E_r(F_{b_i}, F_{a_i}) < \delta, j = \overline{1,q}\}$, where q – a number of the FB from the set F_a , equivalent to FB F_{bi} , δ – a similarity threshold for two functional blocks.

Therefore, in order to eliminate the uncertainty and to define the equivalent FBs unambiguously, the choice refinement of the EFBs is to be performed.

3.4 The Choice Refinement of the Equivalent Functional Blocks and its Comparison.

In order to perform the choice refinement of EFBs, a probability matrix for the opcodes sequence of the functional blocks, which are defined as equivalent, is built. The columns and the rows of the matrix determine the opcodes that are present in the function block. Each cell of the matrix consists the ratio of the number of the opcodes pair appearance to the total number the opcodes in the row.

The procedure of the choice refinement of the EFBs involves the comparison of the probability matrices for the opcodes sequence of each functional blocks F_b with the each FB from the set F_a , which are equivalent to F_{b_i} , and the choice of the minimum value of the similarity score E_p : min $(E_p(F_{b_i}, F_{a_1}), \dots, E_p(F_{b_i}, F_{a_a})) \Rightarrow F_{b_i} \sim F_{a_i}$.

The similarity score is evaluated using the similarity metric. In order to evaluate the quantitative features of the code obfuscation, the comparison of the obtained pairs of the EFBs using the metric of Damerau-Levenshtein is carried out. As a result, the following quantitative obfuscation features are obtained: the Damerau-Levenstein distance, the number of insertion, deletion, transposition and the matches of opcodes.

Drawing the conclusion about the similarity degree of suspicious program to the metamorphic virus, based on the obtained features, is performed by the means of the fuzzy inference system [13, 14].

The approach demonstrated the efficiency of the metamorphic viruses detection at the rate of 79%. The main reason of such low detection rate is the wrong choice of the EFBs during the search of the EFBS and its choice refinement. To achieve the better result of the proposed approach is to correctly choose the similarity metric.

4 Experimental Results

According to the approach described in section 3, the obtaining of the obfuscation quantitative features deals with the determining of the equivalent functional blocks in the suspicious program and its modified version. It is implemented using the similarity metrics. Thus, the main interest of the research is to investigate the impact of involvement of the similarity metrics for the right choice of the equivalent functional blocks for comparison and the correctly (at two stages of the search of the EFBs and the choice refinement of the EFBs).

In order to implement the metamorphic viruses' detection efficiently the analysis of the quantitative features of the code obfuscation was performed. For this purpose a number of experiments were conducted. As a test data 540 samples of the metamorphic viruses using the NGVCK, VCL32 and G2 metamorphic generators were generated (VX Heavens). The entire study process was conducted in the modified emulator environment [2] based on Qemu.

In [2] as the similarity metric to evaluate the similarity scores and the squared Euclidean metric was used. In this research as the similarity metrics the following metrics were used: the Euclidean metric (3), the squared Euclidean metric (4), the taxicab distance (5), the Chebyshev distance (6), and the Minkowski distance (7).

$$m1 = E_r(F_a, F_b) = E_p(F_a, F_b) = \sqrt{\sum_{i=1, j=1}^{\max(m, n)} (F_{a_i} - F_{b_j})^2}$$
(3)

$$m2 = E_r(F_a, F_b) = E_p(F_a, F_b) = \sum_{i=1, j=1}^{\max(m, n)} (F_{a_i} - F_{b_j})^2$$
(4)

$$m3 = E_r(F_a, F_b) = E_p(F_a, F_b) = \sum_{i=1, j=1}^{\max(m,n)} |F_{a_i} - F_{b_j}|$$
(5)

$$m4 = E_r(F_a, F_b) = E_p(F_a, F_b) = \max(|F_{a_i} - F_{b_j}|)$$
(6)

$$m5 = E_r(F_a, F_b) = E_p(F_a, F_b) = \sum_{i=1, j=1}^{\max(m, n)} |F_{a_i} - F_{b_j}|^r \text{ with } r=3$$
(7)

where, m and n – the numbers of the EFBs in the suspicious program F_b , and its modified version F_b , respectively.

The best results of the NGVCK samples' detection were succeeded applying the conditions: usage of the taxicab metric at the stage of the EFBs search with the value of the similarity metric threshold $\delta = 0.6$; usage of the Chebyshev metric at the stage of the choice refinement of the EFBs (detection efficiency - 94%, false positives 4%).

The best detection results of the VCL32 samples' were achieved applying the conditions: usage of the squared Euclidean metric at the stage of the EFBs search with the value of threshold $\delta = 0.6$; usage of the Chebyshev metric at the stage of the choice refinement of the EFBs (detection efficiency - 98%, false positives 3%).

In addition, the best results for the G2 samples' detection were succeeded applying the conditions: usage of the taxicab metric at the stage of the EFBs search with the value of the similarity metric threshold $\delta = 0.6$; usage of the Chebyshev metric at the stage of the choice refinement of the EFBs (detection efficiency - 100%, false positives 2%). The experimental results are presented in figure 2. It should be noted, that the decreasing of the similarity metric threshold value leads to the increasing of the number of EFBs.



Fig. 2. Efficiency of the metamorphic viruses detection and false positives rates *FPR* for the NGVCK (a), VCL32 (c), G2 (e); the dependency of the equivalent functional blocks number on the value of the similarity threshold δ ; the dependency of detection efficiency on the metric selection *mi* for the NGVCK (b), VCL32 (d), G2 (f)

5 Conclusion

The paper presents an approach for the metamorphic viruses detection based on its obfuscation features analysis. The obfuscation features were obtained on the basis of the equivalent functional blocks search in the suspicious program and its modified version. As a result of the comparison, the following quantitative obfuscation features

are obtained: the Damaerau-Levenstein distance, the number of insertion, deletion, transposition and the matches of opcodes.

The experiments results demonstrated that the detection efficiency depends on the choice of the similarity metrics at the stages of the search and the choice refinement of the equivalent functional blocks. The best results of the metamorphic virus detection were achieved with usage of the taxicab metric at the stage of the EFBs search and the Chebyshev metric at the stage of the choice refinement of the EFBs. It allowed increasing efficiency from 79% to 94% in comparison with the previous research.

References

- 1. McAfee Labs Threat Report. December 2017. Availabe: https://www.mcafee.com /us/resources/reports/rp-quarterly-threats-dec-2017.pdf
- Savenko, O., Lysenko, S., Nicheporuk, A., Savenko, B.: Approach for the Unknown Metamorphic Virus Detection. In: 9-th IEEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems. Technology and Applications, Bucharest, Romania, pp. 453-458 (2017)
- Shafiq, M.Z., Tabish, S.M., Mirza, F., Farooq M., PE-Miner: Mining Structural Information to Detect Malicious Executables in Realtime, International Workshop on Recent Advances in Intrusion Detection, pp. 121-141 (2009)
- David, B., Filiol, E. Gallienne, K. Structural analysis of binary executable headers for malware detection optimization. Journal of Computer Virology and Hacking Techniques, Vol. 13(2), pp. 87-93 (2017)
- Runwal, N., Low, R.M., Stamp, M.: Opcode Graph Similarity and Metamorphic Detection. Journal in Computer Virology, Vol. 8, pp. 37-52 (2012)
- Rad, B. B., Masrom, M., Ibrahim, S. Opcodes histogram for classifying metamorphic portable executables malware. In: International Conference on e-Learning and e-Technologies in Education, Lodz, Poland, pp. 209-213 (2012)
- Canfora, G., Iannaccone, A. N., Visaggio, C. A. Static analysis for the detection of metamorphic computer viruses using repeated-instructions counting heuristics. Journal of Computer Virology and Hacking Techniques, Vol. 10 (1), pp. 11-27 (2014)
- Fazlali, M., Khodamoradi, P., Mardukhi, F., Nosrati, M., Dehshibi, M.M. Metamorphic malware detection using opcode frequency rate and decision tree. International Journal of Information Security and Privacy, Vol. 10(3), pp. 67-86 (2016)
- Bezobrazov, S., Sachenko, A. Komar, M., Rubanau, V. The methods of artificial intelligence for malicious applications detection in Android OS. International Journal of Computing 15 (3), pp. 184-190 (2016)
- Lysenko, S., Savenko, O., Kryshchuk, A., Kljots, Y. Botnet detection technique for corporate area network. In: Proc. of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, pp. 363-368 (2013)
- Eskandari, M., Hashemi, S. A graph mining approach for detecting unknown malwares. Journal of Visual Languages and Computing, Vol 23 (3), pp. 154-162 (2012)
- 12. Kaushal, K. Swadas, P., Prajapati N. Metamorphic Malware Detection Using Statistical Analysis. International Journal of Soft Computing and Engineering, Vol. 2 (3), pp. 49-53 (2012)
- Savenko, O., Lysenko, S., Nicheporuk, A., Savenko, B.: Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search. ICT in Education, Research and Industrial Applications, Integration, Harmonization and Knowledge Transfer, Vol. 1844, pp. 555-569 (2017)
- Kondratenko, Y., Kondratenko, V.: Soft Computing Algorithm for Arithmetic Multiplication of Fuzzy Sets Based on Universal Analytic Models. In book: Information and Communication Technologies in Education, Research, and Industrial Application. Communications in Computer and Information Science 469, pp. 49–77 (2014)