First International Congress on Technology, Communication and Knowledge (ICTCK 2014)
November, 26-27, 2014 - Mashhad Branch, Islamic Azad University, Mashhad, Iran

Role-opcode vs. Opcode: the New method in Computer Malware Detection

Zahra Ghezelbigloo
Department of Computer Engineering
Imamreza University of Mashhad
Mashhad, Iran
Zahra.Ghezelbigloo@yahoo.com

Majid VafaeiJahan
Department of Computer Engineering
Mashhad Branch-Islamic Azad University
Mashhad, Iran
VafaeiJahan@mshdiau.ac.ir

***Abstract*—One of the common methods in the area of combating with malwares is the use of opcodes-sequence exist in the malwares' assembly code. In this study, a new method has been used based on the structural classification of opcodes to detect malwares and its efficiency has also been put into investigation compared to the opcodes method. For this purpose, two different methods are to be applied for eliciting the content-based features of the assembly files. Two approaches were, then, analyzed on an equal basis using different classifications. The results, thereof, have indicated that the efficiency and the accuracy of different classifications do not decrease by using structural classification of opcodes. Additionally, the number of features, computational complexity, the time and the memory consumption would dramatically be decreased.**

I. NTRODUCTION

As the software applications evolve, some developers may implement security measures to be ensured that their products are protected with high security. Malwares, however, are increasingly threatened systems. Just as the technology improves to combat malwares, malwares, too, developed to infiltrate into codes.

Scanning and signature-based methods are among the earliest ways which were established to combat with various types of malwares. This method was used commercially in different types of anti- Malwares and anti-viruses. However, one of the problems with applications like anti-virus is to detect and clean viruses after being run or appeared on computers. Moreover, since the vulnerability of signature-based methods was so high, this method did not work for the next generation of viruses. Later on, the code-based techniques were propounded to combat with malwares. Attempts were made in these methods by analyzing the files at the level of code to make the diagnosis of types of malicious codes, viruses and … possible before any action. Many efforts have been made in relation with malicious codes by using statistical analysis of files at the level of binary codes [1, 2, 3]. Having analyzed the malwares of binary codes, the idea of using opcodes exists in assembly code was propounded.

One of the most important measures carried out in line with the analysis of opcodes and computer malware detection was done by "Bilar" et al. in 2007 [4]. They have analyzed various types of opcodes including widely used cases and rare samples in their study and they have also showed that there is a major difference in the frequency of rare opcodes in malicious files as compared to the healthy files toward the widely used opcodes. Meskovich (2008) [5] was later on used the n-gram model of opcodes to detect the computer malwares. They have indicated in their study that the accuracy of over 99% would be achieved if the percentage of malicious files is about 15% of the total samples. "Santos" et al. 2010 [6] proposed a system to identify various types of opcode-based malware. He continued his research on class learning and opcode-sequance-based semi-supervised detection methods in the following year and achieved favorable results [7, 8]. They have completed their measures on malware detection in 2013. They have presented a method for examining the frequency of each opcode-sequence based on the opcodes-sequence frequency appeared in the source code. In this study, data mining methods was used for malware detection [9].

In this very present study the opcodes extracted from the assembly code of each sample are classified in 10 categories in structural manner and in accord with their role. Thereupon, instead of using and analyzing opcodes-sequence, the role-opcode-sequence is applied. For the purpose of evaluating the efficiency of the presented method, two diverse approaches have been used for extracting the contenct-based features of the assembly files. The first one is based on the frequency of opcodes appeared in the source code and the second one is based on the frequency of consecutive role-opcodes appeared in the source code. Two methods were, then, brought under scrutiny on an equal basis using various classifications. The present article can be divided into the following parts:

In the next section, the method suggested by this paper has been propounded. In part 3, mention has been made of the manner of characteristics extraction while the results coming from the samples' classification over and above the comparison in between our suggested method and its similar

978-1-4799-8021-5/14/$31.00 ©2014 IEEE

counterpart has been conducted within the bounds of part 4. And the conclusion to the article has been brought in the last section.

II. II. DEFINE ROLES - OPCODE

For the purpose of extracting the role-opcode, each file, initially, converted to the sequence of machine instructions entitled assembly code by means of using disassembler. Such instructions consist of two parts: operands and opcode. The following example shows the assembly code related to an executable file.

```
push esi
call SUB_L0100D36F
pop ecx
test eax,eax
jnz L0100D281
push 0000001Ch
call SUB_L0100D330
pop ecx
```

Since opcodes, itself, could also indicates the application of a file, the analysis of operand section has been canceled and only the opcodes-sequence of each file is extracted. In this case, the extracted opcode-sequence would be like {jnz, push, call, pop} for the above example. The average number of opcodes in various executable files is approximately 250 cases. The frequency of opcodes or their histogram is then extracted as features of a file. Using of n-gram statistical model is a very common method for extracting features in this area. But, because of the high number of opcodes and the dimensional feature space, using of this model is difficult for values over n and it is not possible in many cases. Writers, therefore, use this model, in many cases, for values below n to reduce space or apply feature selection methods to reduce dimensional feature space.

In this study, a new method has been presented using opcodes classification to reduce their numbers. Unlike other existing methods, which are classifying features based on the statistical characteristics such as frequency and/or… using data mining methods, here some kind of semantic classification is used which is based on tasks and defined roles of each opcode. In this method, first, different roles of opcodes would be defined and then, they would be classified in to the categories which are proportionate to the role (task) of each opcode. To this purpose, opcodes are classified in their respective groups after being defined in terms of their type and the number of intended roles in accord with authoritative sources [10, 11]. For instance, in accord with the definition presented in the X86 Opcode and instruction reference home, the role and the task of "And" and "Or" opcodes is logical operation [12]. In this study, a number would be assigned to each opcode as its role. Fig. 1 shows the role of some existing opcodes using color separation.

For example, the purple color indicates opcodes whose tasks are to carrying out the input/output operation and the green color represents opcodes which conducted logical and mathematical operation.

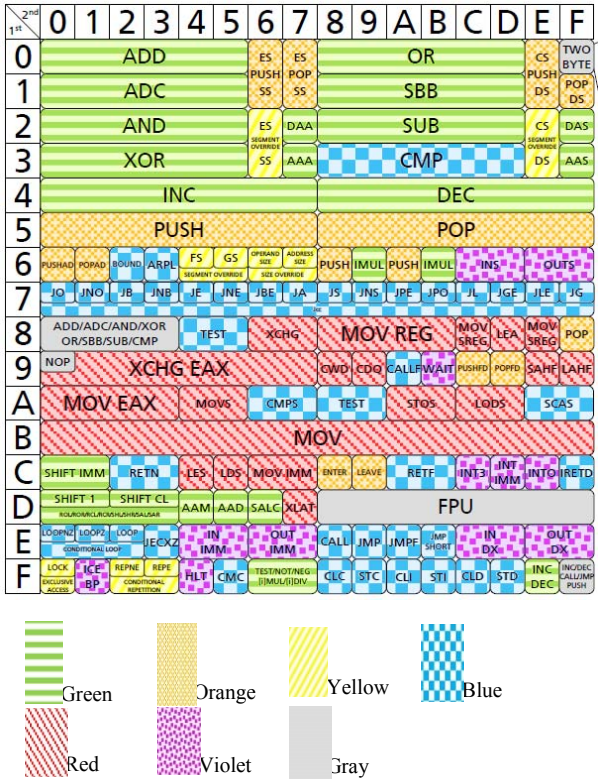

Figure1. The role of some existing opcodes using color separation [13]

Then, the number of roles is counted to be as vocabulary and the role-opcode-sequence is analyzed. In this present study, 10 different roles have been considered for extracted opcode sets. Table 1 illustrates the roles defined in this article by giving example of the existing opcodes.

TABLE1. THE ROLE DEFINED WITH EXAMPLES OF OPCODE

	Defined roles for the opcode	Sample opcode
1	Logical Operations	Add, or
2	Memory-related Operations	Mov, Stos
3	Stack Operations	Push, Pop
4	The control and status Operations	Test, Call
5	Opcodes prefixed	Fs, Repe
6	System operation and input / output	Out, In
7	Several structural and developed opcodes	Nop,Fpu
8	Logical Operations (The letter "F" begins.)	Fmul, Fsub
9	Specific opcodes	Punpckhbw
10	Other opcode	…

III. FEATURE EXTRACTION

Feature extraction is of the most important parts in diagnosis, classification, prediction and in data mining methods. Generally, the dimensional feature space and the complexity of the problem followed by would increase with

the increase of the number of features. To this purpose, finding the lowest and the most influential number of features had always been one of the major challenges in data mining issues. This article made attempts to find the most effective features in detecting malwares from executable files by having the lowest number of features and by decreasing the computational load of the issue. In both methods, no special technique would be used to extract features.

A. Role-opcode

In this article, after extracting the role-opcode-sequence in accord with the method presented in part III, the frequency of consecutive role-opcode-sequence of the length n –observed in the assembly source code of each file as of the feature of that file – has been put into use. For example, the role-opcode-sequence of the length 2 related to the example mentioned in part III, is equal to:

{(push,call),(call,pop),(Pop,test),(Test,jnz),(Jnz,push),(Push,call),(call,pop)}

It would be like the following sequence after becoming a role-opcode.

{(3-4),(4-3),(3-4),(4-4),(4-3),(3-4),(4-3)}

Thence, the frequency of n-gram model (consecutive sub-sequences of the length n) of role-opcode, would be analyzed as a feature. For further clarity of the subject, the frequency of the consecutive sub-sequences of role-opcode of the length one and two has been elucidated. To this description made of the 1-gram model, only 10 features would be extracted. The average number of frequency for sub-sequences of the length one is put forth in figures 2 & 3. It can be observed through the figure that the frequency of prefixed opcodes in malwares and opcodes whose tasks would be to run logical operations, is more than healthy files. Furthermore, we could say that the number of opcodes whose tasks would be to run logical operations and opcodes whose tasks would be related to stack is almost equal; yet, in healthy files, opcodes which are related to stack are relatively more than opcodes whose task would be to run the logical operations.

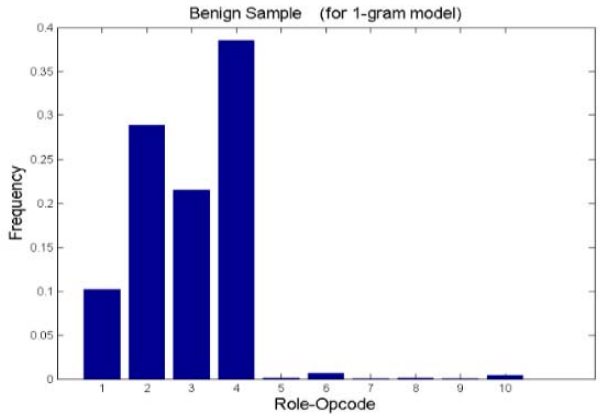

Figure2. Average Frequency Model 1-gram of role - opcode (Benign)

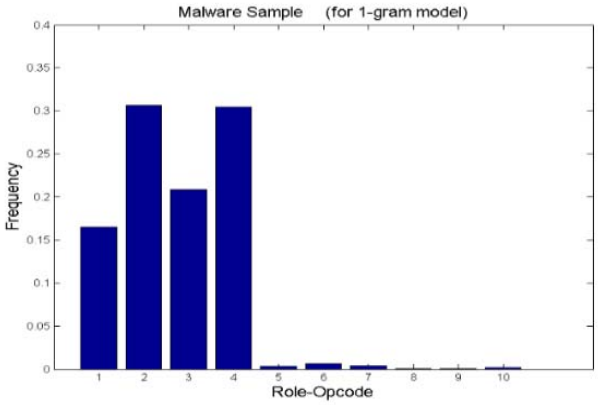

Figure3. Average Frequency Model 1-gram of role - opcode (Malware)

For 2-gram model (sub-sequences of the length 2), the number of extracted features are totally 100 samples; in some cases, none of them were observed in total samples. In this study, of the total of 100 features, 98 of which have been observed in cases under testing. The average frequency of the role-opcodes sub-sequences of the length 2 is presented in figures 4 & 5.

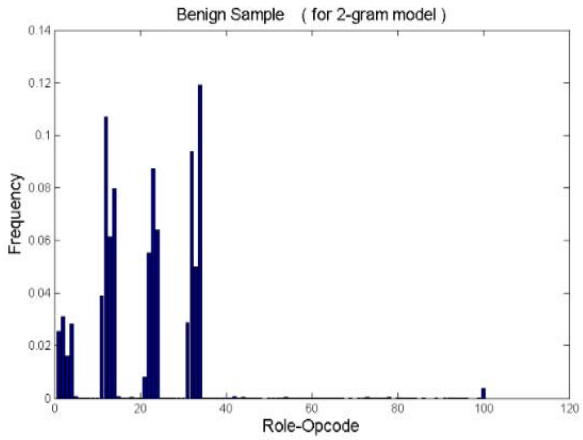

Figure4. Average Frequency Model 2-gram of role - opcode (Benign)

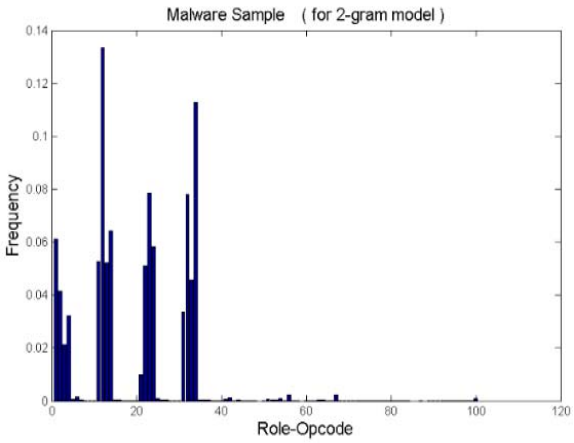

Figure5. Average Frequency Model 2-gram of role - opcode (Malware)

It can be realized from figures 4 & 5 that in many cases, the average frequency of role-opcodes of the length 2 are very low; however, there are differences between the healthy files and the malicious files. For example, the frequency of the pair consecutive samples of opcodes in malicious files whose tasks is related to the logical operations and the operations related to memory is much more than in healthy files; while the frequency of pair consecutive samples of opcodes in healthy files whose tasks are respectively the operations related to memory-control and also the control-memory is more than the executable files.

B. Opcode

In the second method, only the existing opcode sequences are extracted and their frequency would be used to be compared to the first method. The number of extracted opcodes is 215 cases in this study. In accord with the previous state, the frequency of sub-sequences would be extracted with the specified length using n-gram model. For example, there are 215 states in 1-gram model. The average frequency of the opcodes sub-sequence of the length one has been illustrated in the following figures. Opcodes with numbers from 1 to 10 are to be counted as widely used ones. They were numbered in accord with the English alphabet. According to figures 6 & 7, it can be seen that the average frequency for opcodes from 10 to 20 is more different in healthy and malicious files than other opcodes, while the average frequency of the widely used opcodes is less diverse.

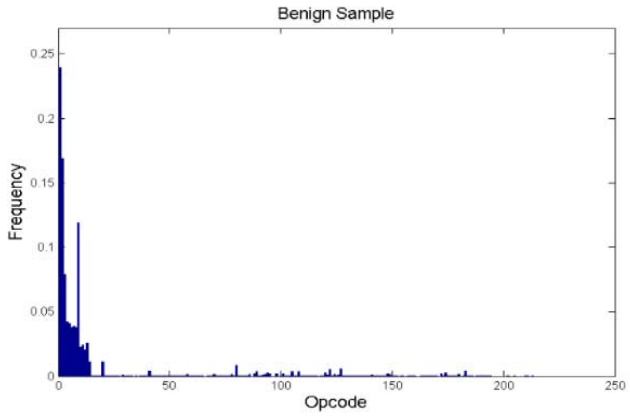

Figure6. Average Frequency Model 1-gram of opcode (Benign)

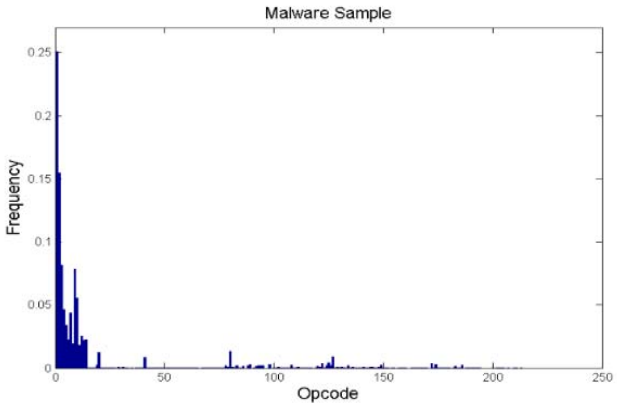

Figure7. Average Frequency Model 1-gram of opcode (Malware)

IV. THE EXPERIMENTALANALYSIS

In this study, a database including 3445 sample of malicious file and 1497 sample of healthy file has been used which are totally equal to 4942 samples. The healthy files have been brought under scrutiny by means of anti-virus related to the Microsoft Co. [14]. The malware sets are from several reputable sites such as virusshare, Malwarelu and Malshare which have been received in the first 6 months of the year 2013 [15, 16, 17].

In the first method, the appeared opcodes-sequence has been extracted from each sample; 215 cases of opcode have been totally observed. The probable frequency of each sample is to be calculated after the extraction of opcodes-sequence. The dimensional feature space is equal to 215 in the first method.

In the second method, the role-opcode sequence is to be achieved in accord with the method presented in part II & III after the opcode-sequences being extracted from each sample. Afterwards, the probable frequency of the consecutive role-opcodes' sub-sequences of the length one and two would be calculated.

Tests have been done separately on three states of 1-gram with 10 features, 2-gram with 97 features and the combination of 1-gram and 2-gram models with 107 features. For the purpose of comparing two methods presented throughout in this article, two criteria of accuracy and the time required for making this model have been used. All the results have been obtained through using Weka data mining software. The system to be applied consists of 8 GB of RAM, Intel Core i7, and Windows 8 professional 64-Bit.

A. Classification accuracy

The degree of accuracy and precision of different classifications is shown forth in figure 8 for the number of different features and the two provided methods. Having been considered the figures, it is so obvious that in so many cases the results obtained from the opcode method are so close to the results related to the role-opcode method (for the state of 107 features). The role-opcode method, however, has more

favorable results in relation to the support vector machine classifier.

The best result observed in this diagram is related to the role-opcode method using random committee classifier for 107 features, which is equal to 94.5%. Additionally, the best result observed for the opcodes method is related to the random forest classifier for 215 features, which is equal to 94.5%.

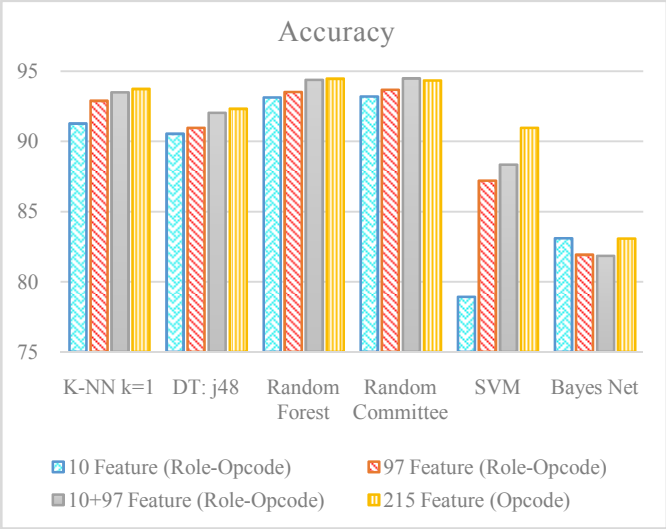

Figure8. AccuracyChartofthe variousclassificationmethods

B. Time to build a model

The diagram related to the time required for the model-making in different classifications, is shown in figure 9. For more clarity of the numbers and high differences of the results related to the support vector machine classifier, the results to be obtained from this classifier have been separately shown forth in figure 10.

Having considered the figure 9, the minimum time required for model-making was related to the role-opcode method and for 10 features, which is very natural due to the low number of features. Noteworthy in this figure is the reduction of time required for model-making for combining the features related to 1-gram and 2-gram models. In this state, although, the number of features has been increased to 107 cases, the time required for model-making is less than the state with 97 features. Also, the maximum time spent is related to the opcode method.

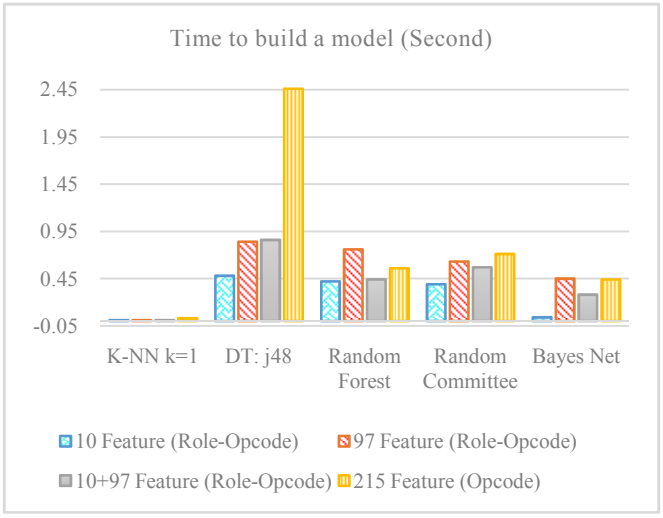

Figure9. Time required to build the model in the variousclassificationmethods

One of the notable points in figure 10 is that in addition to the fact that the time duration for making the combination of 1-gram and 2-gram model features (similar to figure 9) is less than the time required for model-making in opcodes method with 215 features and the role-opcodes related to 2-gram model, it is also less than the time required for making 1-gram model.

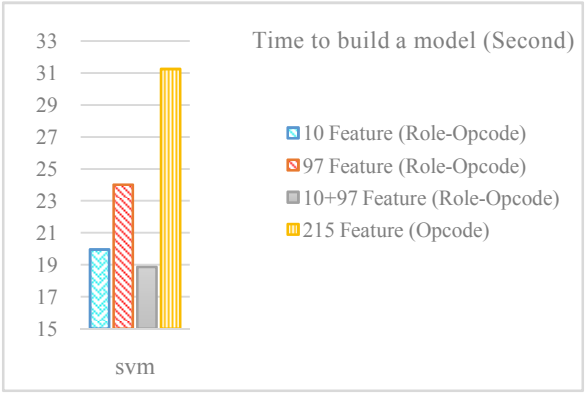

Figure10. Time required to build the model in the SVMmethod

V. CONCLUSIONS

In this study, for the purpose of appraising the performance and the functionality of the new method based on the role and the task of opcodes, titled "Role-opcode", this method is compared to its similar one named opcode. To compare these two methods, two diverse approaches have been used for extracting the content-based features of the assembly files. The first approach is based on the frequency of opcodes appeared in the source code and the second one is based on the frequency of consecutive role-opcodes appeared in the source code. Two approaches were, then, put into examination on an equal basis and by using different classifications. Considering the fact that the average number of opcodes in different sources is 250 cases, the amount of

features, memory consumption, computational complexity in feature selection and the time required for model-making would exponentially be increased with the increase of the amount of *n* in the n-gram model. However, in the new method, because of having the low number of roles, number of features, memory consumption, calculation complexity and the time required for model-making, the role-opcode would be decreased significantly. Results relating to the tests of accuracy of different classifications indicate that the opcodes method – despite the fact of being able to analyze the samples' classification in more details – does not have any special priority over the role-opcode technique. Thus, we can be optimistic about achieving favorable outcomes in detecting types of computer malwares by using the role-opcode method, calculating the larger values of n and applying appropriate feature selection methods.

REFERENCES

[1] K. Kaushal, P. Swadas and N. Prajapati, "Metamorphic Malware Detection Using Statistical Analysis," *International Journal of Soft Computing and Engineering (IJSCE),* vol. 2, no. 3, pp. 49-53, July 2012.

[2] S. M. Tabish, M. Z. Shafiq and M. Farooq, "Malware detection using statistical analysis of byte-level file content," in *ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics*, New York, NY, USA, 2009.

[3] B. Jochheim, "On the Automatic Detection of Embedded Malicious Binary Code using Signal Processing Techniques Project Report," Hamburg, October 17, 2012.

[4] D. Bilar, "OpCodes as predictor for malware," *International Journal of International Journal of,* vol. 1, no. 2, pp. 156-168, 2007.

[5] R. Moskovitch, C. Feher, N. Tzachar, E. Berger, M. Gitelman, S. Dolev and Y. Elovici, "Unknown Malcode Detection Using OPCODE Representation," in *1st European Conference on Intelligence and Security Informatics*, 2008.

[6] I. Santos, F. Brezo, J. Nieves, Y. K. Penya and B. Sanz, "Idea: Opcode-sequence-based Malware Detection," in *Engineering Secure Software and System*, 2010.

[7] I. Santos, F. Brezo, B. Sanz, C. Laorden and P. G. Bringas, "Using opcode sequences in single-class learning to detect unknown malware," *IET Information Security,* 2011.

[8] I. Santos, B. Sanz, C. Laorden, F. Brezo and P. G. Bringas, "Opcode-sequence-based Semi-supervised Unknown Malware Detection," *Computational Intelligence in Security for Information Systems,* vol. 6694, pp. 50-57, 2011.

[9] I. Santos, F. Brezo, X. Ugarte-Pedrero and P. G. Bringas, "Opcode sequences as representation of executables for data-mining-based unknown malware detection," *Information Sciences,* vol. 231, pp. 64-82, 2013.

[10] MazeGen, "X86 Opcode and Instruction Reference," 20 1 2009. [Online]. Available: http://ref.x86asm.net/.

[11] "x86 instruction listings," 3 Junuary 2014. [Online]. Available: http://en.wikipedia.org/wiki/X86_instruction_listings.

[12] "X86 Opcode," 2013. [Online]. Available: http://ref.x86asm.net/coder32.html.

[13] A. Albertini, "x86 Opcode Structure and Instruction Overview," 30 8 2011. [Online]. Available: http://net.cs.uni-bonn.de/fileadmin/user_upload/plohmann/x86_opcode_structure_and_instruction_overview.pdf.

[14] "kaspersky lab," 2013. [Online]. Available: http://www.kaspersky.com/pure.

[15] P. Rascagneres, "malware.lu," 3 2013. [Online]. Available: http://www.malware.lu/pages/company.html.

[16] J.-M. Roberts, "VirusShare," 24 4 2013. [Online]. Available: http://virusshare.com/.

[17] D. Wood, "VirusSign," 4 2013. [Online]. Available: http://www.virussign.com/downloads.html.