Global Journal of Computer Science and Technology: E
Network, Web & Security
Volume 16 Issue 5 Version 1.0 Year 2016
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Review of Contemporary Literature on Machine Learning based Malware Analysis and Detection Strategies

By G.Bala Krishna, Dr. V.Radha & Dr. K. Venu Gopala Rao
KMIT/JNTUH

Abstract- Malicious software also known as malware are the critical security threat experienced by the current ear of internet and computer system users. The malwares can morph to access or control the system level operations in multiple dimensions. The traditional malware detection strategies detects by signatures, which are not capable to notify the unknown malwares. The machine learning models learns from the behavioral patterns of the existing malwares and attempts to notify the malwares with similar behavioral patterns, hence these strategies often succeeds to notify even about unknown malwares. This manuscript explored the detailed review of machine learning based malware detection strategies found in contemporary literature.

Keywords: malware detection, malware signature, API call sequence, anomalies, static analysis, dynamic analysis, machine learning.

GJCST-E Classification : C.2.0 D.4.6 H.2.7

Strictly as per the compliance and regulations of:

Review of Contemporary Literature on Machine Learning based Malware Analysis and Detection Strategies

G.Bala Krishna [α], Dr. V.Radha [σ] & Dr. K. Venu Gopala Rao [ρ]

Abstract- **Malicious software also known as malware are the critical security threat experienced by the current ear of internet and computer system users. The malwares can morph to access or control the system level operations in multiple dimensions. The traditional malware detection strategies detects by signatures, which are not capable to notify the unknown malwares. The machine learning models learns from the behavioral patterns of the existing malwares and attempts to notify the malwares with similar behavioral patterns, hence these strategies often succeeds to notify even about unknown malwares. This manuscript explored the detailed review of machine learning based malware detection strategies found in contemporary literature.**

Keywords: malware detection, malware signature, API call sequence, anomalies, static analysis, dynamic analysis, machine learning.

I. Introduction

The term "Malware" stands for malicious software, and it usually specifies as hostile software application. According to G. Mc Graw et al., [1] there are multiple causes as code added, changed, or removed from the software it get corrupt and it deliberately causes harm and disrupt normal computing activity. A virus had a broad range of destructive software applications such as viruses, Trojans, Spywares and other intrusive code [2].

The malware can discriminate by the capability of replication, propagation, self-execution and corruption of the operating system. If the computer system gets extortion it influence on confidential information, integrity and denial of assistance. In malware Replication is a crucial component as it assure its existence.

In some cases Replication generates consumption and continuation of system resources (e. g. hard disk, RAM). If confidential assets are being used by any other malware types other than the user, to conceal themselves from anti-malware detector they use a technique called polymorphic or metamorphic techniques.

Author α: Assistant professor, Dept. of IT, KMIT Hyderabad. e-mail: govind.krishna83@gmail.com
Author σ: Assistant Professor, IDRBT Hyd e-mail: vradha@idrbt.ac.in
Author ρ: Professor, Dept. of CSE, GNITS Hyderabad. e-mail: kvgrao1234@gmail.com

The operating system gets corrupt through data transfer from desecrate device to another protected device familiar, such as executable files, boot records of disk drives or exhausting network bandwidth, by using local or network files system. In such case malware makes operating system susceptibility and few software bugs are faults and it starts its life cycle at the same system and infected system simultaneously by remotely control.

According to a McAfee simplified report (year 2013) says that "malware continues to grow" [3] and by G Data and king [4] [5] soft Laboratory declare n-number of innovative malware will emerge promptly and to build an anti-malware the analyzers and constructors are enhanced by their unique techniques and methods [6]-[10]. To construct a malicious software the techniques which are been used to categorized and estimate in groups such as obfuscation techniques, invocation methods, platform, spreading and propagation techniques.

To actuate a program has a malicious attentive or not, malware detection system is used. In this detection system there are two different functions, detection and analysis [12]. Detection system is a protecting one as it may or may not be prevail in the same system [13] and the tasks can be split into client and server as it analogous in cloud-based antivirus [8, 12]. A numerous renovations are made on detection and analysis functions [5], [12], [15]-[19].

In malware detection system specialized solutions are added to expansion in success and achievement. Such as cloud computing [10], network based detection system [20], web, virtual machine [21], [22], agent technology [23]-[29] or by the use of hybrid methods and technologies.

II. Review of Contemporary Literature and Contributions

In earlier stage malware had come up with signature based detection. But now in this stage malware signature has introduced an automatic generation and it is pretended to be important and it increased its pattern in similar speed.

The signature based detection system has some imperfection as follows to continue the updates of

signature it requires high maintenance cost. By inclusion such methods it could be evenly avoided by malware in polymorphic form [30]. To conquer the imperfection, it embraces code in normal vision to grab the consecrate original maliciousness. To vary the polymorphic techniques and apply, this grabbed malicious code is used but still it is anemic to detecting obfuscated malware. Apart from this some execution paths can be explored execution [31] [30].

Due to certain requirements the malware analysis is all ways conserved the techniques in the prior, consequently dynamic analysis was considered. To identify and execute a complicate malware dynamic analysis methods are used. In dynamic analysis the malware shows how it operates and recognize the unknown malware which is identically operates like a known malware [32]. There are two familiar primary dynamic methods are control flow analysis and API call analysis. [33] [34].

API call data display how the malware gets operates and it can be obtained by both static and dynamic approaches. The API list and PE format of the executable files can be derived by the static approach [35] [36] [37] [38]. In dynamic approach. [39] [40] [41], [42] [43] [44] API calls can be recognized by running executable files it usually run in virtual machine.

In API call there two familiar ways to evaluate the data accumulated by static approach. The first one implements simple statistical analysis, for example, to count the frequency of API call which is aspect to organize malware [35]. The second approach is to gather the API call data through data mining or machine learning techniques. In another way the API call sequence data which gathered by the dynamic approach are helpful to creates a behavioral patterns. The information accumulated by the dynamic approach also operates simple statistics such as frequency counting [39] and data mining or machine learning [40] [42] [44].

In other way, researchers are analyzed more ways to develop API call sequence information. In earlier research API call had introduced API call graph [45] with various kinds of call graph analysis. To get more consequential features for call graph analysis, the analyzer had espoused the mechanism of social network analysis. [46] According to analyzers the affinity among API call sequences is based on cosine similarity function and lengthy jaccard measure. Due to modern research [33] [34] [47] [48] more information had been added such as control flow information and API argument information to inflate the efficiency in the mining process.

The API call analysis been done with API call approaches. In this abstraction the dynamic method is applied to excerpt API call sequences. To obtain austerity patterns, DNA sequence alignment algorithms (MSA and LCS) are adapted. With API call sequence patterns and the critical API call sequence, we can recognize the unknown malware or its variation with elevated efficiency.

Anderson et al. [49] defined a malware detection strategy that builds a set of graphs from the given instruction set and then analyzes these graphs to notify the proneness of the malware activity. In order to build the graphs the markov chains were defined on 2-gram sequences. The graphs defined form the training set further used to build a similarity matrix using graph kernel. The graph kernel is the mix of Gaussian and spectral kernels, which are in use to assess the similarity between graph edges and similarity between graphs respectively. Further the support vector machine that learns from the similarities between graph edges and graphs is used to classify the input call sequences.

By using such liberal malware software the multiple kernel is achieved and learning design used in this work to exhibit selective refined differences occurrence of malware. The inadequacy of this approach is computed consequence is very high, hence the use of this approach is discouraged.

Bayer et al. [50] prospect a technique that groupsthe call sequences generated by Anubis [51]. The behavior adequately of the call sequences is considered as objective to cluster the call sequences by Locality Sensitivity Hashing (LSH) [52]. The constraint of the model is that LSH is capable to generate probabilistic clusters.

Biley et al. [53] argued that malware prototyping is not consistent among the notable antivirus products available. In order to this the authors devised a novel classification strategy that classifies the malware according to the changes observed at system state. A strategy that prototypes the behavior of the malware is used, further the malwares are classified according to these behavior prototypes. The distance between a class and a malware is assessed by the distance metric called “normalized compression distance (NCD)”. The constraint observed in empirical study of this model is that the behavior prototype definition is static and limited to malwares that are not fall in zero-day category (unknown malwares). park et al. [54] defined a classification strategy that classifies malware based on the graphs generated from the call sequences. Further graph similarities between confirmed malware call sequence graph and unknown call sequence graph will be assessed. The similarity index is the “max number of subgraphs identified in both graphs”. The malwares those controls the system privileges without initiating the system call sequences are not traceable by this classification model, which is a significant constraint of this model. Firdausi et al. [55] defined a machine learning model for malware detection. The said model initiates the process by exploring the behavioral patterns of the malware samples given for training, which is done

by the model called Anubis [51]. Further these observed behavioral patterns will be organized as sparse vectors and learns the behavior prototypes. The malware samples given for testing will be classified, which is based on the behavioral prototypes learned in training phase. The performance of the model is estimated through benchmark classifiers and they are "j48", "multilayer perception neural networks (MLP)" "Naïve Bayes", "Support Vector Machine (SVM)" and "k- Nearest Neighbors (kNN)". The experimental results indicating that the J48 classification delivered much classification accuracy.

Nari et al. [56] devised a network flow behavioral analysis framework for malware detection. The network transactions obtained from PCAP files were considered to extract the network flows. Further a network activity representation graph is drawn from these network flows. The given network flows labeled as malware were used in training phase. Further this framework learns representation of the features such as size of the graph, average, maximum and root level out-degree and count of specific nodes of the network activity graphs of the given input network flows. Further these features specific information uses to classify the input malware samples in testing phase. In order to perform the classification, the WEKA library [57] was used. The experimental study indicating that the J48 is the best classifier among all classifiers available in WEKA library.

Lee et al. [58] explored a machine learning based malwares clustering. The training phase builds the behavioral profiles of the malware samples given as training data and the profile includes the system resources invoked by the system calls and their arguments. Further the similarities between behavioral profiles were considered distance function to cluster the malwares, which was done by k-medoids. The outliers are adjusted to the clusters based on the nearest neighbor strategy. This approach is the combination of static and dynamic clustering strategy that clusters known features by k-medoid and unknown and new features by nearest neighbor approach. This strategy is evincing that hybrid approach is more robust in order to classify the known as well as unknown features effectively.

Another hybrid approach for malware detection was devised by Santos et al. [59]. This approach tracks the known features (static features) through the analysis of the sequence of operational codes in given malicious executable and the unknown features (dynamic features) were noticed from the observation of exceptions and operations in system calls. The experimental study was done under various classifiers and results obtained were evincing the significant accuracy in malware classification Islam et al. [60] explored a similar strategy that extracts static and dynamic features to classify the executables into malevolent or benevolent. The features such as function length, function executable frequency and length of the strings involved are included in known features and the features such as function identity and function arguments are included in unknown features. The experiments were done using the classifiers called Support Vector Machine, Decision Tree and Random Forest and results evincing that the random forest is the best classifier among all considered.

The malware classification method devised by Anderson et al. [61] is using the divergent input sources such as control flow graphs, static call sequences, portioned executables, dynamic call sequences and file signatures. Further this model learns the weight of these input combinations from the given training set of malevolent and benevolent executables. The observed weights of these input combinations are used further to classify the malevolent and benevolent executables during testing phase. The process overhead is the significant constraint of this model observed against dense and high speed network streams.

III. Conclusion

The current era of internet and computer systems are prone to serious security threats due to the malicious software which are also referred as malware. Hence the significant research contributions aimed to define malware detection and prevention strategies in contemporary literature. All of these contributions are fall in the categories of either anomaly based, signature based or call sequence analysis based detection. The signature based models are capable to notify and prevent the malwares that are notified earlier. In contrast to this the anomaly models and call sequence analysis models are capable to identify the malwares based on the similarities learned from previous malware attacks. The difference between the anomaly and call sequence analysis models is that the anomaly based learning models can adopt user defined features, whereas the call sequence analysis models notify the similarities learned from the call sequences of 2-gram, 3-gram or n-gram. This manuscript aimed to affirm the objectives and limits of the contributions found in recent literature. The conclusion of the review evincing that the machine learning based models that learns from either anomalies or call sequence are tolerable the constraints observed in signature based malware detection strategies. The anomaly and call sequence learning models found in contemporary literature are not adequate to defend the challenges evincing from the vibrant and unjust network data. Hence the significant contributions are in demand to handle the challenges evinced in current era of internet and computer system usage.

References Références Referencias

1. McGraw, G. and G. Morrisett, Attacking Malicious Code: A Report to the Infosec Research Council. IEEE Softw., 2000. 17 (5): p. 33-41.
2. Xufang, L., P. K. K. Loh, and F. Tan. Mechanisms of Polymorphic and Metamorphic Viruses. in Intelligence and Security Informatics Conference (EISIC), 2011 European. 2011.
3. Mcafee and Lab, 2013 Threats Predictions. 2013.
4. Berkenkopf, R. B. S., G-Data Malware Report. 2010.
5. Ye, Y., et al., Intelligent file scoring system for malware detection from the gray list, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009, ACM: Paris, France. p. 1385-1394.
6. Rieck, K., Malheur A novel tool for malware analysis 2012.
7. Pinz, C. I., et al., Improving the security level of the FUSION@ multi-agent architecture. Expert Syst. Appl., 2012. 39 (8): p. 7536- 7545.
8. Ammar Ahmed E. Elhadi, M. A. Maarof, and A. H. Osman, Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph. American Journal of Applied Sciences, 2012. 9 (3): p. 283-288.
9. Kevadia Kaushal, P. S., Nilesh Prajapati, Metamorphic Malware Detection Using Statistical Analysis. International Journal of Soft Computing and Engineering (IJSCE), 2012. 2 (3).
10. Yanfang Ye, T. L., Shenghuo Zhu, Weiwei Zhuang, EgemenTas, Umesh Gupta, MelihAbdulhayoglu, Combining file content and file relations for cloud based malware detection, in Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. 2011, ACM: San Diego, California, USA. p. 222- 230.
11. Christodorescu, M., et al., Semantics-Aware Malware Detection, in Proceedings of the 2005 IEEE Symposium on Security and Privacy. 2005, IEEE Computer Society. p. 32- 46.
12. Yin, H., et al., Panorama: capturing systemwide information flow for malware detection and analysis, in Proceedings of the 14th ACM conference on Computer and communications security. 2007, ACM: Alexandria, Virginia, USA. p. 116-127.
13. Vinod, P., et al., Survey on Malware Detection Methods. 2009.
14. Zeltser, L., what is cloud Anti-Virus and how it does work.
15. Jiang, X., X. Wang, and D. Xu, Stealthy malware detection through vmm-based "outof- the-box" semantic view reconstruction, in Computer and communications security. 2007, ACM: Alexandria, Virginia, USA. p. 128-138.
16. Automated dynamic binary analysis. 2007.
17. Deepak Venugopal, G. H., Efficient signature based malware detection on mobile devices. Mob. Inf. Syst., 2008. 4 (1): p. 33-49.
18. Kolbitsch, C., et al., Effective and efficient malware detection at the end host, in Proceedings of the 18th conference on USENIX security symposium. 2009, USENIX Association: Montreal, Canada. p. 351-366.
19. Zhou, S. T. a. M., A Heuristic Approach for Detection of Obfuscated Malware,. IEEE, 2009.
20. [20] Ahmed, M., et al. NIDS: A Network Based Approach to Intrusion Detection and Prevention. in Computer Science and Information Technology - Spring Conference, 2009. IACSITSC '09. International Association of. 2009.
21. Garfinkel, T. and M. Rosenblum, A virtual machine introspection based architecture for intrusion detection. 2003: p. 191--206.
22. Lagar-Cavilla, H. A., Flexible Computing with Virtual Machines. 2009.
23. Gorodetsky, V., et al., Multi-agent Peer-to- Peer Intrusion Detection Computer Network Security, V. Gorodetsky, I. Kotenko, and V. A. Skormin, Editors. 2007, Springer Berlin Heidelberg. p. 260-271.
24. Ye, D., An Agent-Based Framework for Distributed Intrusion Detections. 2009.
25. Ou, C. -M. and C. R. Ou, Agent-Based immunity for computer virus: abstraction from dendritic cell algorithm with danger theory, in Proceedings of the 5th international conference on Advances in Grid and Pervasive Computing. 2010, Springer-Verlag: Hualien, Taiwan. p. 670-678.
26. Bijani, S. and D. Robertson, Intrusion detection in open peer-to-peer multi-agent systems, in Proceedings of the 5th international conference on Autonomous infrastructure, management, and security: managing the dynamics of networks and services. 2011, Springer Verlag: Nancy, France. p. 177-180.
27. Dong, H., et al. Research on adaptive distributed intrusion detection system model based on Multi-Agent. in Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on. 2011.
28. Ou, C. M., Multiagent-based computer virus detection systems: abstraction from dendritic cell algorithm with danger theory. Springerlink, 2011.
29. Paritosh Das, R. N., A Temporal Logic Based Approach to Multi-Agent Intrusion Detection and Prevention. 2012.
30. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection, " in Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC '07), pp. 421–430, December 2007.
31. P. Okane, S. Sezer, and K. McLaughlin, "Obfuscation: the hidden malware, " IEEE Security & Privacy, vol. 9, no. 5, pp. 41–47, 2011.

32. S. Cesare and Y. Xiang, Software Similarity and Classification, Springer Science & Business Media, 2012.
33. M. Rajagopalan, M. A. Hiltunen, T. Jim, and R. D. Schlichting, "System call monitoring using authenticated system calls, " IEEE Transactions on Dependable and Secure Computing, vol. 3, no. 3, pp. 216–229, 2006.
34. M. Abadi, M. Budiu, U. Erlingsson, and J. Ligatti, "Control- ´flow integrity," in Proceedings of the 12th ACM Conference on Computer and Communications Security, pp. 340–353, November 2005.
35. S. Sathyanarayan, P. Kohli, and B. Bruhadeshwar, "Signature generation and detection of malware families, " in Information Security and Privacy, Springer, Berlin, Germany, 2008.
36. Sami, B. Yadegari, H. Rahimi, N. Peiravian, S. Hashemi, and A. Hamze, "Malware detection based on mining API calls," in Proceedings of the 25th Annual ACM Symposium on Applied Computing (SAC '10), pp. 1020–1025, ACM, March 2010.
37. Y. Ye, 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, pp. 1043–1047, ACM, August 2007.
38. M. Alazab, S. Venkatraman, and P. Watters, "Zero-day malware detection based on supervised learning algorithms of API call signatures," in Proceedings of the 9th Australasian Data Mining Conference (AusDM '11), vol. 121, pp. 171–182, Australian Computer Society, December 2011.
39. R. Tian, M. R. Islam, L. Batten, and S. Versteeg, "Differentiating malware from cleanware using behavioural analysis," in Proceedings of the 5th International Conference on Malicious and Unwanted Software (MALWARE '10), pp. 23–30, Nancy, France, October 2010.
40. M. Shankarapani, K. Kancherla, S. Ramammoorthy, R. Movva, and S. Mukkamala, "Kernel machines for malware classification and similarity analysis, " in Proceedings of the International Joint Conference on Neural Networks (IJCNN '10), pp. 1–6, July 2010.
41. M. K. Shankarapani, S. Ramamoorthy, R. S. Movva, and S. Mukkamala, "Malware detection using assembly and API call sequences," Journal in Computer Virology, vol. 7, no. 2, pp. 107– 119, 2011.
42. F. Ahmed, H. Hameed, M. Z. Shafiq, and M. Farooq, "Using spatiotemporal linformationin API call swithmachinelea rning algorithms for malware detection, " in Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence, pp. 55– 62, November 2009.
43. Y. Qiao, Y. Yang, J. He, C. Tang, and Z. Liu, "CBM: free, automatic malware analysis framework using API call sequences," in Knowledge Engineering and Management, pp. 225–236, Springer, Berlin, Germany, 2014.
44. Y. Qiao, Y. Yang, L. Ji, andJ. He, "Analyzing malware by abstracting the frequent item sets in API call sequences, " in Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom '13), pp. 265–270, July 2013.
45. J. Bergeron, M. Debbabi, J. Desharnais, M. M. Erhioui, Y. Lavoie, and N. Tawbi, "Static detection of malicious code in executable programs, " in Proceedings of the Symposium on Requirements Engineering for Information Security (SREIS '01), 2001.
46. J. -W. Jang, J. Woo, J. Yun, and H. K. Kim, "Mal-netminer: malware classification based on social network analysis of call graph, " in Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion (WWW Companion '14), pp. 731–734, International World Wide Web Conferences Steering Committee, 2014.
47. K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning, " Journal of Computer Security, vol. 19, no. 4, pp. 639–668, 2011.
48. C. M. Linn, M. Rajagopalan, S. Baker, C. Collberg, S. K. Debray, and J. H. Hartman, "Protecting against unexpected system calls," in Proceedings of the 14th USENIX Security Symposium, pp. 239–254, Baltimore, Md, USA, August 2005.
49. Anderson, B., Quist, D., Neil, J., Storlie, C. and Lane, T. (2011) Graph Based Malware Detection Using Dynamic Analysis. Journal in Computer Virology, 7, 247-258. http://dx. doi. org/10. 1007/s11416-011-0152-x
50. Bayer, U., Comparetti, P. M., Hlauschek, C. and Kruegel, C. (2009) Scalable, Behavior- Based Malware Clustering. Proceedings of the 16th Annual Network and Distributed System Security Symposium.
51. Anubis. http://anubis. iseclab. org/
52. Indyk, P. and Motwani, R. (1998) Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality. Proceedings of 30th Annual ACM Symposium on Theory of Computing, Dallas, 24-26 May 1998, 604-613.
53. Biley, M., Oberheid, J., Andersen, J., Morley Mao, Z., Jahanian, F. and Nazario, J. (2007) Automated Classification and Analysis of Internet Malware. Proceedings of the 10th International Conference on Recent Advances in Intrusion Detection, 4637, 178-197. http://dx. doi. org/10. 1007/978-3-540-74320-0_10
54. Park, Y., Reeves, D., Mulukutla, V. and Sundaravel, B. (2010) Fast Malware Classification by Automated Behavioral Graph Matching. Proceedings of the 6th

Annual Workshop on Cyber Security and Information Intelligence Research, Article No. 45.
55. Firdausi, I., Lim, C. and Erwin, A. (2010) Analysis of Machine Learning Techniques Used in Behavior Based Malware Detection. Proceedings of 2nd International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT), Jakarta, 2-3 December 2010, 201-203.
56. Nari, S. and Ghorbani, A. (2013) Automated Malware Classification Based on Network Behavior. Proceedings of International Conference on Computing, Networking and Communications (ICNC), San Diego, 28-31 January 2013, 642-647.
57. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. (2009) The WEKA Data Mining Software: An Update. ACM SIGKDD Explorations Newsletter, 10- 18.
58. Lee, T. and Mody, J. J. (2006) Behavioral Classification. Proceedings of the European Institute for Computer Antivirus Research Conference (EICAR'06).
59. Santos, I., Devesa, J., Brezo, F., Nieves, J. and Bringas, P. G. (2013) OPEM: A Static- Dynamic Approach for Machine Learning Based Malware Detection. Proceedings of International Conference CISIS'12- ICEUTE'12, Special Sessions Advances in Intelligent Systems and Computing, 189, 271- 280.
60. Islam, R., Tian, R., Battenb, L. and Versteeg, S. (2013) Classification of Malware Based on Integrated Static and Dynamic Features. Journal of Network and Computer Application, 36, 646-556. http://dx. doi. org/10. 1016/j. jnca. 2012. 10. 004
61. Anderson, B., Storlie, C. and Lane, T. (2012) Improving Malware Classification: Bridging the Static/Dynamic Gap. Proceedings of 5th ACM Workshop on Security and Artificial Intelligence (AISec), 3-14.