 ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Malware Analysis, Clustering and Classification: A Literature Review

[1]Nirmal Singh, [2]Dr. Sawtantar Singh Khurmi
[1]Dept. of Computer Science, Desh Bhagat University, Mandi Gobingarh, Punjab, India
[2]Ex. HOD, Dept. Of Computer Science, Bhai Maha Singh College of Engg., Muktsar, Punjab, India

Abstract
Malware is major security threat on the Internet now-a-days. Anti-Virus companies receive large number of malware samples every day. Malware samples are classified and grouped for further analysis. There are different type of malware analysis, clustering and classification methods available. The purpose of this study is to examine the available literature on malware analysis, clustering and classification.

Keywords
Malware, Malware Analysis, Malware Classification, Malware Clustering.

I. Introduction

The term malware or malicious software (also known as computer virus) is used for any computer program that is designed with explicit intent to compromise system and to perform various types of frauds. The first-ever PC virus (malware) Brain appeared in 1986 [1]. Since then, motivation behind developing them has changed from fun to financial benefits. Now-a-days, malwares are not created just for fun but for financial benefits. Malwares are used to send spam e-mails, to perform web frauds, to steal personal information like credit card information and for many other nefarious tasks like Ransomware [2] and fake antivirus software [3]. According to the report published by McAfee [4] Cybercrimes are costing up to US$500 billion annually. The widespread diffusion of malicious activities has resulted in the biggest problems for the Internet community today.

II. Malware Analysis

There are basically two malware analysis techniques exits i.e. dynamic malware analysis and static malware analysis [5]. In static malware analysis, malware analysis is done without executing the malware. If the malware is written in any scripting language like JavaScript, PHP, Perl then malware analyst can easily read the code and can understanding the working of the suspicious file. Also static analysis tools can be used to find out the suspicious and risky functions. But static analysis tools can fail to give correct results for obfuscated code. Various limitation of static malware analysis has been described in [6]. There has been presented different methods of obfuscation also in [6].

To overcome limitations of static malware analysis for obfuscation code, dynamic malware analysis techniques are used. In dynamic malware analysis, a malware is executed in virtual machine [7] or in an emulated environment [8]. In [9], there has been provided a detailed survey of various existing automatic dynamic malware analysis systems and a comparison of their analysis inputs and capabilities. Most of the dynamic analysis systems [10-12] monitor the behaviour of the malware by monitoring the system calls/APIs used by the malware. These automated malware analysis systems provides analysis reports as an output that describes the behaviour of the given malware sample. By observing the behaviour report a malware analyst can decide whether a given sample is malware or not.

III. Malware Clustering and Classification

Malware clustering and classification methods can be divided into three categories –

- Based on Static Analysis
- Based on Dynamic Analysis
- Based on Hybrid Analysis

A. Based on Static Analysis

Schultz et al. [13] used static features of PE metadata. They used three different features – DLL imported and functions referred, strings and byte sequences. They used machine learning algorithm – Naïve Bayes. They tested their method on a data set consisted of 4266 files (3265 malware and 1001 clean files). They achieved classification accuracy of 97.11%.

Tian et al. [14] presented classification method based on function length and frequency of function length. Length of function is measured by number of bytes of code in a function. IDA [15] is used for extracting these values from unpacked code. They used k-fold cross validation for calculating similarity between two samples and achieved 88% average accuracy.

Siddiqui et al. [16] used variable length instructions extracted from the disassembly of worms and clean files and data mining. Before disassembly, malware and clean files are unpacked after packer detection. Feature reduction is done to select only important features before classification. They used Random Forest, Bagging and Decision tree algorithms for classification and achieved 96% accuracy.

Wicherski G. [17] developed a non-cryptographic hash using PE file features and Kolmogorov Complexity for malware clustering. He used PE structural properties -: Image Characteristics, Subsystem, Stack Commit Size and Heap Commit Size along with Virtual Address, Raw Size, Section Characteristics and Kolmogorov Complexity from each section in PE file. Bzip2 compression ratio is used as an approximation of Kolmogorov Complexity [18]. He tested this method on three data sets containing 184538, 90105 and 322 samples collected from mwcollect Alliance Database, Arbor Networks and Windows XP installation respectively. There was 0 false positive for malware files and 7 false positives for files collected from Windows XP installation.

Leder et al [19] used value set analysis (VSA) method for malware classification. VSA is a static analysis technique to track the propagation and changes of values in an executable. File is disassembled to collect value set. Two value sets are compared for classification. They were able to achieve 100 % accuracy.

Yu et al. [20] proposed byte frequency based detecting model (BFBDM) for variant identification. Byte's value ranges from 0-255 so bytes frequency is calculated for each byte in a given file. This results in a 254-dimensional vector space. Euclidean distance and cosine similarity is used for classification.

Kinable et al. [21] used call graph clustering method for malware clustering. Call graphs are generated from binary file using disassembly tools. Before generating the call graphs, file is unpacked and decrypted. They used IDA Pro disassembler tool for extracting functions and assign them symbolic names. They used

graph edit distance (GED) for graph matching and k-medoids and density-based clustering algorithm (DBSCAN) for clustering.

Nataraj et al [22] proposed a classification algorithm method based on image processing techniques. Malware samples are visualized as gray-scale images. They observed that similar malware binaries will appear very similar in layout and texture. A malware is read as vector of 8 bit unsinged integers and converted to a 2D array. They used k-nearest neighbour with Euclidean distance for classification. Proposed method was evaluated on a data set of 9458 samples having 25 different malware families. They achieved 98% classification accuracy.

Raman [23] used static PE file features for classification. Features are selected from metadata of PE file. He selected DebugSize, ImageVersion, IatRVA, ExportSize, ResourceSize, VirtualSize2, and NumberOfSections features from PE metadata. He tested IBk, J48, J48 Graft, PART, Random Forest, and Ridor classification algorithms on data set of 100000 malware and 16000 clean files using the seven features as input to these algorithms. With J48 algorithm he achieved accuracy of 98.5 %.

Rad et al. [24] used histogram of opcodes for malware classification. This histogram represents the distribution of opcodes in malware sample. They used disassembler program to extract opcodes from a malware. Firstly, data base of different versions of a morphed virus is built. For a given sample, histogram of opcodes is generated and average of histograms of different version of virus in virus database is calculated. The classification is achieved using comparison of dissimilarity between histogram of PE file and histogram of a virus family. They used Euclidean form distance metric for calculating dissimilarity. They tested the method on 100 various PE-files and got 100 % accuracy.

Kang et al. [25] presented a major block comparison (MBC) system which identify important parts of binaries to represent a malware family. Binary file is divided into major blocks such as blocks with system calls and blocks with user-defined functions. Similarity of two files is calculated in two phases. In first phase, each major block in one file is compared with major blocks in other file then aggregated similarity is calculated from the results of first phase. System was able to classify malware successfully by analysing only 24% of binary contents.

Kong et al. [26] presented automatic malware classification framework using discriminant distance learning on structural information extracted from malware. Framework extract function call graph along with various types of fine-grained features like system calls made and I/O read/write oprations from a program. For evaluating the similarity of two malware programs, discriminant distance metric learning and pairwise graph matching is used.

Aljamea et al. [27] used static analysis based approach using text based search technique (control flow graph, hashing) and machine learning. They used IDA Pro to disassemble malware. From disassembly, they extracted control flow graph (CFG) and generated cryptographic hash of each block of each function in CFG. After this, they applied text-based search to gain the most frequent basic block from CFG. For classification, they used decision tree (j48) algorithm implemented in WEKA classifier.

Hu et al. [28] presented a framework, called MutantX-S, which takes advantage of instruction format of x86 architecture to represent a program as sequence of opcodes. N-gram features are extracted from binary data. Hashing trick is used to reduce the dimensionality of the extracted feature vectors. Framework is evaluated using a database of 100000 malware samples. It was able to cluster 80% of the samples. But being a static-analysis approach, MutantX-S is not able to handle binary/instruction level obfuscation.

Lakhotia et al. [29] presented VILO, a malware classification system that make use of three components – N-perm feature vector, Term Frequency × Inverse Document Frequency (TFIDF) weighting of features and nearest neighbour search algorithm. N-perm is variation of N-gram. N-perms are extracted from disassembly of an executable. They evaluated their system on four malware family sets by choosing 100 samples from each set. They divided 100 samples into 40 and 60 for training and verification respectively. They achieved 86.44% accuracy for classification.

B. Based on Dynamic Analysis

Biley et al. [30] used malware behaviour in terms of system changes for malware classification. Malware is executed in controlled environment (VM) with Windows XP installed. Malware behaviour report is collected that includes files written, processes created and network activity. For clustering malware, a pairwise single linkage hierarchical algorithm is used with normalized compression distance (NCD) as a distance metric. The limitation of this work is that some malware doesn't show full behaviour in VM.

Bayer et al. [31] presented a clustering method using behaviour profiles. A malware is executed in controlled environment to collect behaviour profiles. These profiles represent malware behaviour in terms of operating system (OS) objects and OS operations. For clustering, they used local sensitive hashing (LSH). They were able to cluster 75,000 samples in less than 3 hours.

Firdausi et al. [32] presented a proof of concept of malware classification based on malware behaviour. Malware is executed in a sandbox environment called Anubis [33]. Sparse vector models are created for each Anubis report. They used 5 different classification algorithms - k-Nearest Neighbors (kNN), Naive Bayes, J48 Decision Tree and Support Vector Machine (SVM). They achieved 93.6% accuracy with J48.

Park et al. [34] proposed a malware classification based on behaviour graphs using maximal common subgraph detection. A behaviour graph is generated by monitoring system calls along with parameter values during execution of a suspicious sample in controlled environment. Similarity measurement of two dynamic system call dependence (DSCD) graphs is computed using maximal common subgraph (MCS). They evaluated proposed method on 300 different samples and were able to classify 95.7 % samples.

Grégio et al. [35] presented a clustering method based on monitoring of data values written in memory by a subset of x86 instructions. A malware sample is executed in controlled environment and data value traces are captured if any instruction from the selected subset of instructions execute. The subset of instructions includes instructions related to arithmetic and logical operations. For comparison of traces they used a two-step algorithm. First, a quick comparison is performed using Jaccard index and selecting k least-frequent bigrams that appear in a trace. After this, full similarity is calculated using longest common subsequence (LCS). They evaluated their method on 16 thousand malware samples and achieved average precision value of .843 for sets and agreement value of .871.

Rafique et al. [36] presented FIRMA, a tool to cluster malware samples based on their network traffic. Network traffic logs are collected by executing malware in controlled environment and a feature vector is extracted from the network traffic. A feature vector contains fid, rid, proto, msg, sport, dport, sip, dip, dsize, endpoint and ptreei. They evaluated the tool on 16,000 samples

and were able to cluster 97.7 % of the samples.

Mohaisen et al. [37] introduced AMAL, an automated behaviour based malware analysis, classification and clustering system. AMAL contains two subsystems, AutoMal and MaLabel. AutoMal collect malware behaviour that contains file system, memory, network and registry modifications. MaLabel uses these artifacts to create features for classifications. It also enables unsupervised learning using multiple clustering algorithms. They evaluated on two sample sets of 4000 and 115000. They achieved 98% accuracy in classification.

Canzanese et al. [38] presented an automatic classification system based on behaviour features extracted from sensors. They selected three distinct types of features - performance monitor, system call and system call sequence - for classification. They used decision trees and random forest algorithms for classification. They evaluated classification system on set of 800 malware and achieved 99% accuracy.

Nari et al [39] present a framework for automated malware classification based on network behaviour of malware. Network flows are extracted from pcap file that contains information about network connections made by malware. A behaviour graph is generated from these network flows. Then features like graph size, root out-degree, average out-degree, maximum out-degree, number of specific nodes are extracted from behaviour graphs. These features are used to classify malware using classification algorithms. They concluded that J48 performs better than other algorithm available in WEKA.

B. Based on Hybrid Analysis

Cesare et at. [40] presented Malwise system that use both static and dynamic analysis techniques. Initially, entropy analysis is performed to determine if given file is packed or not if packed then hidden code is revealed using dynamic analysis. Static analysis is performed on unpacked code for building signatures for control flow graphs in each procedure. CRC64 is used to generate hash of string signature. They evaluated system on 15,000 real malware and were able to classify 86% of the samples in 1.3 seconds.

Santos et al. [41] presented OPEM a hybrid unknown malware detector that use frequency of opcodes and execution trace of executable. Malware is first disassembled to get opcodes and their frequency is calculated using term frequency. For extracting execution tracing, malware is executed inside sandbox. This approach is evaluated on 1000 malicious and 1000 clean files using Decision Tree, K-nearest neighbour, Bayesian network, and Support Vector Machine classification algorithms.

IV. Methodology

Based on the guidelines given by Reed [42], Stapić et al [43] and Webster et. al [44] for writing literature review, We first reviewed doctoral thesis from national and internal universities as these have been reviewed and approved. Online libraries were used for collecting doctoral thesis using keyworks like – "Malware Analysis", "Malware Classification", "Malware Clustering" and "Automatic signature creation for Malware detection". We also reviewed leading journal and conferences papers as these are peer reviewed before publication. For collecting journal and conferences papers, we used online search engines like Google Scholar [45] and Research Gate [46] using the keywords mentioned above.

References

[1] H. J. Highland,"A History of Computer Viruses -The Famous 'Trio'," Computers & Security, Vol. 60, No. 5, pp. 412-415, 1997.

[2] A. Gazet,"Comparative analysis of various ransomware virii", Journal in Computer Virology, Vol. 6, No. 1, pp. 77-90, 2010.

[3] M. Kasuya, K. Kono,"Distinguishing Legitimate and Fake/Crude Antivirus Software", In The Seventh International Conference on Emerging Security Information, Systems and Technologies, Barcelona, Spain, 2013.

[4] McAfee,"IMPACT OF CYBERCRIME AND CYBER ESPIONAGE", Center for Strategic and International Studies, 2013.

[5] S. Gadhiya, K. Bhavsar,"Techniques for Malware Analysis", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, No. 4, pp. 972-975, 2013.

[6] M. Andreas, K. Christopher, K. Engin,"Limits of Static Analysis for Malware Detection", In 23rd Annual Computer Security Applications Conference, Miami Beach, FL, 2007.

[7] R. P. Goldberg,"Survey of virtual machine research", IEEE Computer magazine, pp. 34-45, 1974.

[8] F. Bellard,"QEMU, a Fast and Portable Dynamic Translator", In FREENIX Track of the USENIX Annual Technical Conference, 2005.

[9] M. EGELE, T. SCHOLTE, E. KIRDA and C. KRUEGEL, "A Survey on Automated Dynamic Malware-Analysis Techniques and Tools", ACM Computing Surveys (CSUR), Vol. 44, No. 2, pp. Article No. 6, 2012.

[10] ANUBIS,"Analysis of unknown binaries", [Online]. Available: http://anubis.iseclab.org/. [Accessed 20 08 2013].

[11] ThreatExpert, "ThreatExpert," ThreatExpert, [Online]. Available: http://www.threatexpert.com/. [Accessed 20 8 2013].

[12] COMODO, "COMODO Automated Analysis System," COMODO, [Online]. Available: http://camas.comodo.com/. [Accessed 20 8 2013].

[13] M. G. Schultz, E. Eskin, E. Zadok, S. J. Stolfo,"Data mining methods for detection of new malicious executables", In IEEE Symposium on Security and Privacy, 2001.

[14] R. Tian, L. M. Batten, S. Versteeg,"Function length as a tool for malware classification", In 3rd International Conference on Malicious and Unwanted Software, IEEE, 2008, pp. 69-76.

[15] Hex-rays,"The IDA Pro disassembler and debugger", Hex-rays, [Online]. Available: http://www.hex-rays.com/idapro/. [Accessed 12 2 2014].

[16] M. Siddiqui, M. C. Wang, J. Lee,"Detecting internet worms using data mining techniques", Journal of Systemics, Cybernetics and Informatics, Vol. 6, No. 6, pp. 48-53, 2008.

[17] G. Wicherski,"pehash: A novel approach to fast malware clustering", 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2009.

[18] NIST,"Kolmogorov Complexity", [Online]. Available: http://xlinux.nist.gov/dads//HTML/kolmogorov.html. [Accessed 12 1 2014].

[19] F. Leder, B. Steinbock, P. Martini,"Classification and Detection of Metamorphic Malware using Value Set

Analysis", in 4th International Conference on Malicious and Unwanted Software (MALWARE), 2009.
[20] S. Yu, S. Zhou, L. Liu, R. Yang, J. Luo,"Detecting malware variants by byte frequency", Journal of Networks, Vol. 6, No. 4, pp. 638-645, 2011.
[21] J. Kinable, O. Kostakis,"Malware classification based on call graph clustering", Journal in computer virology, Vol. 7, No. 4, pp. 233-245, 2011.
[22] L. Nataraj, S. Karthikeyan, G. Jacob and B. Manjunath, "Malware images: visualization and automatic classification," in Proceedings of the 8th International Symposium on Visualization for Cyber Security, 2011.
[23] K. Raman,"Selecting features to classify malware", In InfoSec Southwest, 2012.
[24] B. B. Rad, M. Masrom, S. Ibrahim,"Opcodes histogram for classifying metamorphic portable executables malware", In e-Learning and e-Technologies in Education (ICEEE), 2012 International Conference on, 2012.
[25] B. Kang, T. Kim, H. Kwon, Y. Choi, E. G. Im,"Malware classification method via binary content comparison", In Proceedings of the 2012 ACM Research in Applied Computation Symposium, 2012.
[26] D. Kong, G. Yan,"Discriminant Malware Distance Learning on Structural Information for Automated Malware Classification", In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 2013.
[27] M. Aljamea, V. Ghanaei, C. S. Iliopoulos, R. E. Overill, "Static Analysis and Clustering of Malware Applying Text Based Search", The International Conference on Digital Information Processing, E-Business and Cloud Computing (DIPECC2013), pp. 188-193, 2013.
[28] X. Hu, K. . G. Shin, S. Bhatkar, G. Kent,"MutantX-S: Scalable Malware Clustering Based on Static Features", In USENIX Annual Technical Conference, 2013.
[29] A. Lakhotia, A. Walenstein, C. Miles, A. Singh,"VILO: a rapid learning nearest-neighbor classifier for malware triage", Journal of Computer Virology and Hacking Techniques, Vol. 9, No. 3, pp. 109-123, 2013.
[30] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, J. Nazario,"Automated classification and analysis of internet malware", In Recent Advances in Intrusion Detection, Springer Berlin Heidelberg, 2007, pp. 178-197.
[31] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, E. Kirda,"Scalable, behavior-based malware clustering", In Network and Distributed System Security Symposium (NDSS), 2009.
[32] I. Firdausi, C. Lim, A. Erwin, A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection," in 2010 Second International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT), 2010.
[33] ANUBIS, "Analysis of unknown binaries," [Online]. Available: http://anubis.iseclab.org/. [Accessed 20 6 2014].
[34] Y. Park, D. Reeves, V. Mulukutla and B. Sundaravel, "Fast malware classification by automated behavioral graph matching," in Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, 2010.
[35] A. R. A. Grégio, P. L. de Geus, C. Kruegel, G. Vigna, "Tracking memory writes for malware classification and code reuse identification," in Detection of Intrusions and Malware, and Vulnerability Assessment, Springer Berlin Heidelberg, 2013, pp. 134-143.
[36] M. Z. Rafique, J. Caballero, "Firma: Malware clustering and network signature generation with mixed network behaviors," in Research in Attacks, Intrusions, and Defenses, Springer Berlin Heidelberg, 2013, pp. 144-163.
[37] A. Mohaisen, O. Alrawi, M. Larson, "AMAL: Highfidelity, behavior-based automated malware analysis and classification," Verisign Labs, Tech. Rep, 2013.
[38] R. Canzanese, M. Kam, S. Mancoridis, "Toward an automatic, online behavioral malware classification system," in IEEE 7th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), 2013, 2013.
[39] S. Nari, A. A. Ghorbani,"Automated malware classification based on network behavior", In International Conference on Computing, Networking and Communications (ICNC), 2013.
[40] S. Cesare, Y. Xiang, W. Zhou,"Malwise – An Effective and Efficient Classification System for Packed and Polymorphic Malware", IEEE Transactions on Computers, Vol. 62, No. 6, pp. 1193-1206, 2013.
[41] I. Santos, J. Devesa, F. Brezo, J. Nieves, P. G. Bringas, "Opem: A static-dynamic approach for machine-learning-based malware detection", In International Joint Conference CISIS'12-ICEUTE´ 12-SOCO´ 12 Special Sessions, Springer Berlin Heidelberg, 2013, pp. 271-280.
[42] L. E. Reed,"Performing a literature review", In 28th Annual Frontiers in Education Conference, 1998. FIE'98., 1998.
[43] Z. Stapić, E. G. López, A. G. Cabot, L. de Marcos Ortega, V. Strahonja,"Performing systematic literature reviews in software engineering", In Proceedings of 23rd Central European Conference on Information and Intelligent Systems., 2012.
[44] J. Webster, R. T. Watson,"Analyzing the past to prepare for the future: Writing a literature review", Management Information Systems Quarterly, Vol. 26, No. 2, pp. 13-24, 2002.
[45] "Google Scholar", Google, [Online]. Available: http://scholar.google.co.in/. [Accessed 23 3 2014].
[46] "Research Gate", [Online] Available: https://www.researchgate.net. [Accessed 29 5 2014].

Nirmal Singh received his MSc (IT) degree from Mata Gurji College, Fatehgarh Sahib, Punjab, India in 2008. He is currently PhD scholar in Computer Science Dept at DeshBhagat University, MandiGobingarh, Punjab, India. His research mainly focuses on Security threats like malware, exploit detection, analysis and classification.

Sawtantar Singh holds degrees of MCA, M.Phil (Computer Science) and Ph.D. He has authored 15 books, developed course materials for various courses for Distance Education Department, reviewed one book & three research papers for international journals and contributed seven chapters for various books. Besides having delivered guest lectures on several occasions, he has got published 18 research papers at International and National Seminars. He has conducted several workshops chaired sessions at various seminars also.