Detecting and Classifying Morphed Malwares: A Survey

Sanjam Singla
Department of Computer Science
PEC University of Technology Chandigarh, India

Ekta Gandotra
Department of Computer Science
PEC University of Technology Chandigarh, India

Divya Bansal
Department of Computer Science
PEC University of Technology Chandigarh, India

Sanjeev Sofat
Department of Computer Science
PEC University of Technology Chandigarh, India

ABSTRACT
In this era, most of the antivirus companies are facing immense difficulty in detecting morphed malwares as they conceal themselves from detection. Malwares use various techniques to camouflage themselves so as to increase their lifetime. These obscure methods cannot completely impede analysis, but it prolongs the process of analysis and detection. This paper presents a review on malware detection systems and the progress made in detecting advanced malwares which will serve as a reference to researchers interested in working on advance malware detection systems.

Keywords
Malware Evolution, Polymorphic, Oligomorphic, Metamorphic, Obfuscation, Decryptor and Encryptor

1. INTRODUCTION
Malware is an infuriating and hostile software program designed to secretly use the system exclusive of user knowledge. Malware authors are often looking for developing the specific code once as morphed malwares of an existing malware [1], are easy to develop and also prevents from developing a new malware from scratch.

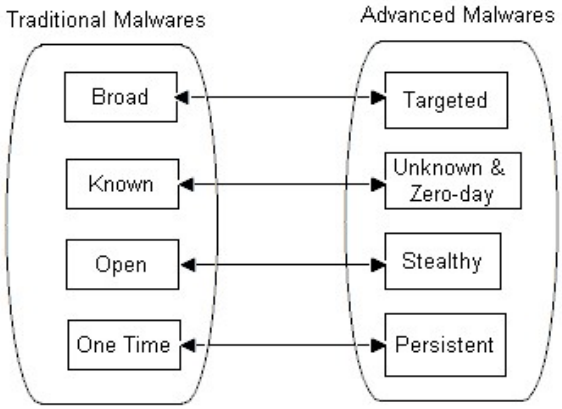

Figure 1. Traditional vs Advanced Malwares [2]

Figure 1 highlights the major differences between traditional and advanced malwares. It is clear from the above figure that new advanced malwares as compared to old traditional malwares are more perilous and difficult to detect as they conceal their presence [2]. According to McAfee 2014 Q3 threat report [3] there is a growth by 76% in appearance of malwares over the past year. Increasing trend in the growth and complexity of malwares makes the job of the antivirus companies much more difficult in detection of these morphed malwares as different generators are used to generate these morphed malwares which mutates after every execution i.e. creating a new malware with the same functionality but with different body structure.

Figure 2 displays evolution period of camouflage techniques in malware and their size depicts the challenges in detecting these malwares.

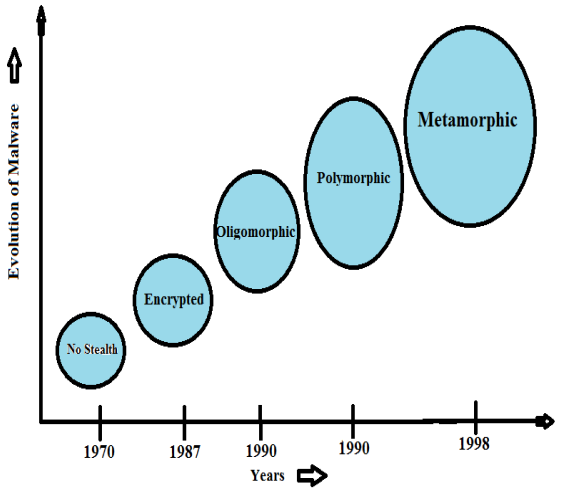

Figure 2. Evolution of Stealth Malwares

This paper presents a brief survey on mechanism of advanced malwares such as polymorphic and metamorphic with their obfuscation and advance detection techniques.

Paper is organized as follows: Section 2 depicts evolution of camouflage malwares followed by their mechanism. Moreover Section 3 explains different concealment techniques used by the malwares to camouflage themselves from detection. Section 4 provides knowledge about various detection techniques and a survey on work done in detecting advanced malwares. Section 5 highlights the discussions with the following gaps in the survey. Finally conclusion is presented in Section 6.

2. CAMOUFLAGE EVOLUTION
Malwares are malevolent software's which are extremely perilous and are a top most security menace to computer systems and the advancement in the malware code by concealing the appearance is a serious challenge for the antivirus companies. This section briefs about advanced morphed malwares with their mechanisms.

A. Encrypted Malwares
The first encrypted virus that came into existence in 1987 was Cascade [5]. The main goal of the malware authors was to

conceal the presence of their malware. For that the first camouflage technique used was encryption. In these viruses body of the viruses is encrypted along with the presence of a decryption module. The body of the malware is encrypted using a key in order to make the detection difficult and moreover these viruses have uniqueness as they use different keys to conceal themselves. But still these viruses can be detected as in these decryption module remains the same which makes the possibility of detection of these malwares by analyzing the decryption module.

B. Oligomorphic Malwares

Oligomorphic virus is known as semi- polymorphic [6]. In this the major difference is that in encrypted virus there is a single decrypted loop whereas in oligomorphic viruses the decryptor loop has different appearance in each new iteration i.e. having the collection of different decryptors. The first oligomorphic virus that appeared in 1990 [7] was Whale, which exhibits properties of a DOS virus.

C. Polymorphic Malwares

Polymorphic malwares are very similar in code encryption [8] as that of earlier versions of malwares. But major difference is that polymorphic viruses contains a million of decryptors that can be produced by modifying instructions in the morphed malware so as to avoid signature based detection.

Mark Washburn wrote the first polymorphic malware named 1260 in 1990 [7]. These polymorphic malwares are generated using obfuscation techniques as explained in section 4. Firstly there is a code decryptor which decrypts the body of the malware and an engine that helps in mutation is attached to the malware body which creates a new decryptor during execution to produce morphed variants.

- *Mechanism*

- A **mutation engine** or a polymorphic engine is a computer program which modifies into another program by using different keys for encryption and producing different decryption modules during execution such that the functionality of the program remains same but the body code is different.
- **Polymorphic Encryptor** encrypts using either same or different key or using simple encryption like "XOR", "AND" or it either use advanced encryption where monoalphabetic substitution tales place.
- **Polymorphic Decryptors** which does not have same ciphering algorithms. Behavior of some of the algorithms is similar to Random Decryption Algorithm and some of them whose decryption is easy are similar to an XOR with a short key.

D. Metamorphic Malware

These malwares show body polymorphism [9] in which instead of using encryption or generating new decryptor, the new body is generated with the help of obfuscation techniques without affecting the overall functionalities. Win95/Regswap [11] is the first metamorphic malware that was created in 1998. These malwares also use concealing or obfuscation techniques to hide their presence and creates new variants [10].

Mechanism

These malwares has the ability to generate new infection each time it is executed such that overall behavior of a malware remains same but the body code is modified. Metamorphic code modifies a particular program by translating its own code into a provisional image, which then rewrites into normal code after editing the image.

- **Entry Point Obfuscation:** (EPO) it is a technique in which the AV scanners are used to prevent the investigation of the files those have been infected. The virus needs to be placed in the flow of program to get activated and gets control. The target executable is patched by the EPO-enabled malware in some place in the centre of the execution process using the call/jump instructions. In this way it receives the control. The AV scanner, that inspects the engine code for entry point is dodged by the EPO.
- **Host Code Mutation:** Transformation to binary code is done with the help of a morphing engine. In this firstly the metamorphic engine has to disassemble the input code and then that engine will generate a new code with same functionality but changing the code body so as to conceal their presence. This kind of behavior varies from virus to virus.
- **Anti-debugging techniques:** Another method of camouflage mostly used by malware software's and packers to make the job of the antivirus companies more difficult and helping a program to protect itself if it is running under a debugger.

3. OBFUSCATION TECHNIQUES

Code obfuscation is basically defined as hiding the presence and making the code difficult to read and understand. In this after execution the code body is modified for new variants while keeping the overall functionality of the malware same. Some code obfuscation techniques [4] are discussed below:

E. Dead Code Insertion

This technique basically inserts dead code in the programming structure of the virus without affecting the code functionality, behavior and metamorphic viruses make use of this obfuscation technique to make the code sufficiently concealed.

F. Instruction Replacement

In this technique instruction are substituted with equivalent instructions which make the detection of these malwares very hard.

G. Register Substituion

In this registers are substituted in different instances of the virus. It keeps the overall functionality same only the programming structure of the virus changes. This method can be used to defeat the string signature detection.

H. Instruction Permutation

In this techniques different combinations of instructions are applied which changes the structure of the code in different iterations.

I. Virtualization Obfuscation

Malware creators use this technique to protect their malware from the process of reverse engineering [8]. In this the code and main logic of the program is virtualized to make the detection more difficult.

J. Code Transposition

In this various conditional and unconditional branches are used for reordering the program but keeping the same flow of execution.

4. DETECTION TECHNIQUES

To combat with the advanced attacks, advanced software's are being developed, assuming that the overall structure of the malware does not modify considerably. As nowadays the Internet is used extensively so the threat from the variant of 2nd generation morphed malwares is escalating everyday. So there is a need that antivirus companies should fight against these malwares before severe damage is done. This section highlights some of the detection techniques:

K. Signature Based Detection

It uses signature as identification and identifies the various families of virus or a single virus using that identification [12]. The main drawback of this is that the frequent updation of signature database is required; to detect malwares else it will not be able to detect new malwares. This type of detection is not suitable for morphed malwares as polymorphic malware uses encryption techniques which make detection using signature impossible and metamorphic malwares are based on code obfuscation techniques, whose detection is also very difficult through signatures.

L. Static Based Detection

In static analysis, the nature of the malwares is analyzed by just looking at the malicious code without executing it. Based on this analysis signatures are created which helps in detection of malwares.

M. Behavior Based Detection

In dynamic analysis the suspicious malware is executed in a virtual environment [13] and its behavior is analyzed. This method requires "templates" of suspicious behavior [14] which are then formed in the form of signature used for detection and analysis of malwares. It would result in less false positives as compare to static based detection and is more reliable.

There are other techniques available for malwares detection like machine learning techniques which are the study of computer algorithms that improved through experiments [15]. Main advantage of these techniques is that it helps in detection of advanced malwares. Many machine learning techniques such as Naive Bayes [16], Decision Tree [17], Data Mining [18], Neural Networks [19] and Hidden Markov Modes [20] have been applied by various researchers for malware classification and detection.

5. EXISTING WORK

A lot of research is done in detection and analysis of malwares. Some popular and recent approaches have been summarized below:

Vinod et al. [21] developed a non signature based approach in order to detect undetectable Metamorphic Malwares. Features selection methods used were GSS, TF-IDF-CF, WET, CPD, TF-IDF, TS. Classification model developed was based on the bi-gram features ranked with the selection technique. They achieved an accuracy of about 99 to 100%. Hira et.al [22] generated signatures using semantic summaries of the morphed malwares. Analysis of control and data flow was done to detect these advanced morphed malwares. Ksenia et al. [23] proposed classification method based on behavioural characteristics. They did their classification by taking into account the number of WinAPI calls made, number of files handled by a malware and finally clustering technique was used. Seyed et al. [24] anticipated that a general malware normalizer called automata structures can be used to store a collection of obfuscation methods which were used for normalizing metamorphic malwares. ADFA or augmented DFA was used for modelling each obfuscation method. This concept traversed the ADFAs so as to search for the occurrence of obfuscated codes in the source code. The presented approach was tested on a large dataset of malwares and clean files, which resulted in achieving high accuracy in detection of metamorphic malwares. Vinod et al. [25] suggested a sophisticated signature extraction approach using multiple sequence alignment (MSA). The proposed method was able to detect most of the variants of a malware with minimum false alarms. Freddy et al. [26] gave an overview about the structural mechanisms of the advanced malwares. Vinod et al. [27] dynamically analysed the API calls made by an executable. After analysis signature of malware class was generated instead of an individual sample. They further used proximity index between different metamorphic generators to determine how similar two generators were. Quinghua et al. [28] proposed a new method for detecting metamorphic malwares through fully automated analysis of executables and by comparing the various system functions and libraries present in the program semantics. Arun et al. [29] tracked metamorphic malwares using the concept of their engine signatures. They used code scoring technique for collecting forensic evidence in order to measure how many code segments were generated by a metamorphic engine.

Christodorescu et al. [30] proposed an interesting aspect using control flow graph for identifying some of obfuscation-deobfuscation techniques for malicious code detection. Identical authors [31] afterwards suggested formalizing templates of specific malicious behavior using instruction semantics and then detecting malwares using a template matching algorithm. PolyUnpack [32] identified the unpack-executing malware by observing the sequence of hidden code in a malware using the combination of dynamic and static analysis. These were identifiable at the time when its static code model was checked against its execution. Kruegel et al. [33] suggested the usage of comparison of binary code and structural analysis. It was based on the control flow. Using graph theory the results of comparison were improved. This technique was computationally expensive.

In their proposed work, authors [34] and [35] detected morphed malware variants using a rewriting engine. Syntactic and semantic structure of variants program was analysed. Krugel et al [36] generated the fingerprint for worms based on their CFG which was rigid against obfuscation techniques. In [37] authors suggested that using code graph can be helpful for analyzing and detecting malware. Furthermore, system call sequence was analysed and a topological graph was produced. Authors in [38] detected variants of a malware using semantic approach which is based on the system call made by the malwares while execution. Authors used Hidden Markov Models (HMMs) to determine the statistical properties of malware variants in [39]. Lee et al. [40] discovered that using some obfuscation techniques, the detection of malwares using HMM can be overcome. Lin et al. [41] proposed a feature extraction technique using n gram and a new list called n-perms was produced using a constant combination.

Tahan et al. [42] used Machine Learning techniques for malware detection, developed and evaluated algorithms and using common segment analysis. Marpaung et al. [43] surveyed evasion and Mitigation techniques and also compared various concealment techniques with the associated features related to attack and detection difficulties. Elhadi et al. [44] proposed a hybrid technique for malware detection. Robin et al. [45] discussed various types of malwares with their mechanism and basic principles of operations. Rehmani et al. [46] analysed

different attack models of the malware to discover the best solution depending on the type of attack. You et al. [4] reviewed obfuscation techniques by analysing encrypted, oligomorphic, polymorphic and metamorphic malwares. Saffaf et al. [47] illustrated the various methods and tools used by antivirus vendors to protect their infrastructure against the malwares attack. Gong et al. [48] suggested adaptive data compression method for Malware detection. Chritodorescu et al. [31] built a detection algorithm which is based on semantics rather than syntactic analysis. Algorithm based on the extracted knowledge from malicious executables behavior, then designed a template and using that template they detected if a program satisfies the behavior. Using this algorithm detection rate of malware variants was very high.

The method described in [49] used a histogram of instruction opcode frequencies to detect morphed malware. Classification of files as malicious or benign was done y comparing the already built histograms of malware samples. Minkowski-form distance was used to find out the similarity between two histograms. Martini et al. [50] used VSA (Value Set Analysis) for detecting metamorphic malware. They executed each malware binary in a virtual environment and tracked the register values for each API (application programming interface) call. Finally the similarity score was computed for classifying the malwares. A framework presented in [51] for polymorphic worm detection was worth mentioning, as it uses byte-pattern-based signatures and graph based classification framework of content based polymorphic worm signatures. Singla et al. [52] proposed a novel approach towards the detection of malwares using static classification in which the feature set used is a combination of Function call frequency and Opcode frequency. This combination provided an accuracy of about 97% for a dataset of 1230 executables containing 800 malwares and 400 cleanwares. Saini et al. [53] used suspicious section count and function call frequency as the features to distinguish malwares from clean ones. They have used machine learning algorithms as available in WEKA library for classification purpose. They achieved an accuracy of more than 98%. Authors in [54] proposed a classification framework which uses integration of both static and dynamic features for distinguishing malwares from clean files. The experimental results, based on a dataset of 998 malwares and 428 cleanwares files provide an accuracy of 99.58% indicating that the hybrid approach enhances the accuracy rate of malware detection and classification over the results obtained when these features are considered separately. Table 1 compares few of the research articles.

6. DISCUSSIONS

After performing the literature review, the following inferences have been made:

1. Though there are several techniques present for advanced morphed malware detection, however most of these techniques are either static or dynamic behavior based.
2. Most of these techniques have been restricted to offline analysis modes only and haven't been validated for real time malware detection.
3. In most of the work, the dataset used in detection of morphed malwares is small in size and processed on raw ASM (Assembly language Source code) files.

After analyzing these gaps, it is evident that there is a need to develop and implement a new technique for detecting morphed malwares which combines the features from both static and dynamic approaches simultaneously along with the machine learning model to test on real time Internet traffic.

7. CONCLUSION

With the advancement in Information Technology, the camouflage in malwares has an exponential growth over the years from simple encryption to complex metamorphic malwares. Today, signature-based malware detection is not sufficiently useful as these advanced malwares uses stealth techniques to hide their presence which are not easily detected by signature based detection and moreover antivirus experts need to have deeper knowledge about these malwares.

This paper discusses the existing mechanisms used by polymorphic and metamorphic malwares in order to evade their detection. It also presents the analysis of existing techniques proposed by various researchers for detecting advanced malwares. This analysis shows that if attention is not paid towards these morphed codes then these codes will become theoretically undetectable virus.

8. REFERENCES

[1] Treadwell S. and Zhou M., 2009. "A Heuristic Approach for Detection of Obfuscated Malware," in Proceedings of the 3rd International Conference on Intelligence and Security Informatics. IEEE, pp. 291–299

[2] Gandotra E., Bansal D. and Sofat S., 2014 "Malware Analysis and classification: A survey," Journal of Information Security, Vol 5, No 2, pp. 56-64, April [Online Available:] http://www.scirp.org/journal/jis http://dx.doi.org/10.4236/jis.2014.52006

[3] Mcafee labs threats report: http://www.mcafee.com/in/resources/reports/rp-quarterlythreat-q3 2014.pdf

[4] You I. and Yim K., 2010 "Malware Obfuscation Techniques: A Brief Survey," Proceedings of International conference on Broadband, Wireless Computing, Communication and Applications, Fukuoka, pp. 297-300

[5] Beaucamps P., 2007 "Advanced Polymorphic Techniques," International Journal of Computer Science, vol. 2, no. 3, pp. 194-205

[6] Aycock J., 2006 "Computer Viruses and Malware," New York, USA: Springer

[7] Szor P., 2005 "The Art of Computer Virus Research and Defence," Addison-Wesley Professional

[8] O'Kane P., Sezer S., and McLaughlin K., 2011 "Obfuscation: The Hidden Malware," Security & Privacy, IEEE, vol. 9, no. 5, pp. 41-47

[9] Rad B.B., Masrom M. and Ibrahim S., 2012 "Camouflage in Malware: From Encryption to Metamorphism," International Journal of Computer Science and Network Security, pp. 74-83

[10] Austin T.H, Filiol E., Josse S. and Stamp M., 2013 "Exploring Hidden Markov Models for Virus Analysis: A Semantic Approach," Proceedings of the 46th Hawaii International Conference on System Sciences, Wailea, HI, USA, pp. 7-10

[11] Ferrie P, Szor P. and Monica S., 2001 "Hunting for Metamorphic," Proceedings of the Virus Bulletin Conference, Czech Republic, Prague, pp. 27-28

[12] Griffin K., Schneider S., Hu X. and Chiueh T., 2009 "Automatic generation of string signatures for malware detection," Proceedings of the 12th International Symposium, RAID, pp. 23- 25

[13] Harley D. and Lee A.,2007 "Heuristic Analysis Detecting Unknown Viruses", [White paper], [Online Available] http://www.eset.Com /us/resources/white-papers/Heuristic Analysis.pdf

[14] Mathur K. and Hiranwal S., 2013 "A Survey on Techniques in Detection and Analyzing Malware Executables," International Journal of Advanced Research in Computer Science and Software Engineering

[15] Mitchell, T. M. "Machine learning", Burr Ridge, IL: McGraw Hill, 1997.

[16] Alazab M. and Venkatraman S., Watters P. and MAlazab Mo., 2011 "Zero-day malware detection based on supervised learning algorithms of Api call signatures," Proceedings of the Ninth Australasian Data Mining Conference, Ballarat, Australia

[17] Moskovitch R., Elovici Y. and Rokach L., 2008 "Detection of unknown computer worms based on behavioural classification of the host," Computational Statistics & Data Analysis

[18] Siddiqui M., Wang M.C. and Lee J., 2008 "A survey of data mining techniques for malware detection using file features," Proceedings of the 46th Annual Southeast Regional Conference, New York, USA, pp. 28-28

[19] Tran N.P. and Lee M., 2013 "High performance string matching for security applications," Proceedings of the International Conference on ICT for Smart Society, Jakarta, pp.13-14-15

[20] Griffin K, Schneider S., Hu X. and Chiueh T., 2009 "Automatic generation of string signatures for malware detection," Proceedings of the 12th International Symposium, RAID, pp. 23- 25

[21] Kuriakose J. and Vinod P., 2014 "Towards the detection of Undetectable Metamorphic malware," SIN'14, Glasgow, Scotland UK

[22] Aggarwal H., Bahler L., Micallef J., Snyder S. and Virodov A., 2013 "Detection of Global, Metamorphic malwares using Control and Data flow Analysis, " IEEE

[23] Tsyganok K., Anikeev M., Tumoyan E. and Babenko L., 2012 "Classification of polymorphic and metamorphic malwares samples based on their behaviour, " SIN

[24] Armoun S.E. and Hashemi S., 2012 "A general paradigm for normalising Metamorphic Malwares," 10th International Conference on Frontiers of Information Technology, IEEE

[25] Vinod P., Laxmi V., Gaur M.S. and Chauhan G., 2012 "MOMENTUM: Metamorphic Malware exploration technique using MSA signatures," International Conference on Innovations in information technology, IEEE

[26] Li X., Loh P.K.K. and Tan F., 2011 "Mechanisms of polymorphic and Metamorphic Viruses," European Intelligence and Security Informatics Conference, IEEE

[27] Vinod P., Laxmi V., Jain H., Golecha Y.K. and Gaur M.S., 2010 "MEDUSA: Metamorphic malware dynamic analysis using signature from API," SIN

[28] Reeves S.D. and Zhang Q, 2005 "MetaAware: Identifying Metamorphic Malware," National Science Foundation (NSF)

[29] Lakhotia A. and Chouchane M.R., 2006 "Using Engine signatures to detect Metamorphic malware," WORM, USA

[30] Christodorescu M. and Jha S., 2003 "Static Analysis of Executables to Detect Malicious Patterns," In Proceedings of the 12th USENIX Security Symposium, pp. 169–186

[31] Christodorescu M., Jha S., Seshia S.A., Song D., and Bryant R.E., 2005 "Semantics-Aware Malware Detection," In Proceedings of IEEE Symposium on Security and Privacy, USA, pp. 32–46

[32] Royal P., Halpin M., Dagon D., Edmonds R., and Lee W., 2006 "PolyUnpack: Automating the Hidden-Code Extraction of Unpack-Executing Malware," In Proceedings of the 22th Annual Computer Security Applications Conference

[33] Kruegel C., Kirda E., Mutz D., Robertson W., and Vigna G., 2005 "Polymorphic Worm Detection Using Structural Information of Executables," In Proceedings of the 8th International Symposium on Recent Advances in Intrusion Detection (RAID), pp. 53–64

[34] Karim M., Walenstein A. and Lakhotia A., 2005 "Malware Phylogeny Generation using Permutations of Code," Journal in Computer Virology, pp. 13–23

[35] Zhang Q. and Reeves S.D., 2007 "MetaAware: Identifying Metamorphic Malware," Computer Security Applications Conference, Annual, pp. 411–420

[36] M. Stamp and W. Wong, "Hunting for Metamorphic Engines," 2006.

[37] Bonfante G., Kaczmarek M. and Marion J., 2009 "Architecture of a Morphological Malware Detector," Computer Virology, pp. 263–270

[38] Kaczmarek M., Bonfante G. and Marion J., 2007 "Control Flow Graphs as Malware Signatures,"

[39] Kruegel C., Kirda E., Mutz D., Robertson W. and Vigna G., 2005 "Polymorphic Worm Detection using Structural Information of Executables," In RAID, Springer, Verlag, pp. 207–226

[40] Lee H. and Jeong K., 2008 "Code Graph for Malware Detection," In International conference on Information Networking, ICOIN, IEEE, pp. 1–5

[41] Lin D. and Stamp, 2011 "Hunting for undetectable metamorphic viruses," In Journal Computer Virology, volume (7), issue (3), pp. 201–214

[42] Tahan G., Rokach L. and Shahar Y., 2012 "Automatic Malware Detection Using Common Segment Analysis and Meta-Features," Journal of Machine Learning Research, pp- 949-979

[43] Marpaung J.A.P, Sain M.and Lee H.J., 2012 "Survey on malware evasion techniques: state of the art and challenges," International Conference of Advanced Communication Technology, pp 19-22

[44] Elhadi A.A.E., Maarof M.A. and Osman A.H, 2012 "Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph," American Journal of Applied Sciences 9

[45] Sharp R., "An Introduction to Malware," Spring 2012 [Online Available]http://orbit.dtu.dk/fedora/objects/orbit:82364/datastreams/file_4918204/content

[46] Rehmani R., Hazarika G.C. and G. Chetia G., 2011 "Malware Threats and Mitigation Strategies: A Survey," Journal of Theoretical and Applied Information Technology, Vol. 29 No.2

[47] Saffaf M.N., "Malware Analysis Bachelors Thesis," Helsinki Metropolis University of Applied Sciences, May 27, 2009

[48] Gong T., Tan X. and Zhu M., 2009 "Malware Detection via Classifying With Compression," The 1st International Conference on Information Science and Engineering, (ICISE)

[49] Rad B.B., Masrom M., and Ibrahim S., 2012 "Opcodes Histogram for Classifying Metamorphic Portable Executables Malware," In ICEEE, pp. 209 – 213

[50] Leder F., Steinbock B., and Martini P., 2009 "Classification and Detection of Metamorphic Malware Using Value Set Analysis," In MALWARE, pp. 39 – 46

[51] Bayoglu B. and Sogukpinar I., 2012 "Graph Based Signature Classes for Detecting Polymorphic Worms via Content Analysis," Computer Network, ISSN 1389-1286, pp. 832–844

[52] Singla S., Gandotra E., Bansal D. & Sofat S., 2015 "A Novel Approach to Malware Detection using Static Classification," International Journal of Computer Science and Information Security (IJCSIS), USA, Vol 13 No.3, ISSN 1947-5500, pp 1-5

[53] Saini, Gandotra E., Bansal D. and Sofat S., 2014 "Classification of PE files using static analysis" SIN'14, Glasgow, Scotland, UK, ACM

[54] Gandotra E., Bansal D. and Sofat S., 2014 "Integrated Framework for Classification of Malwares," SIN'14, Glasgow, Scotland, UK, ACM

Table 1 Comparison of Advance Malware Detection Techniques

S.No	Authors	Dataset	Detection Techniques
1	Jikku Kuriakose, Vinod P [21]	1218 benign executables and 868 NGVCK viruses.	Developed a non signature based approach. Features selection methods used were CPD, WET, TF-IDF, TF-IDF-CF, GSS, TS. Classification model developed was based on the bi-gram features ranked with the selection technique. They claimed an accuracy of about 99 to 100%.
2	Hira aggarwal, Lisa Bahler, Shane Snyder et.al. [22]	76 malware variants from 12 worm/ virus families.	Generated signatures using semantic summaries of the morphed malwares. Analysis of control and data flow was done to detect the advanced morphed malwares.
3	Ksenia Tsyganok, Evgeny Tumoyan, Maxim anikeev, et al. [23]	1080 malware samples gathered by honey pot, Win32 portable executable.	Proposed classification method based on behavioral characteristics. They did their classification by taking into account the number of WinAPI calls made, number of files handled by a malware and finally clustering technique was used to classify them.
4	Babak Bashari Rad, Suhaimi Ibrahim et.al. [49]	122 normal benign files, NGVCK 40 malwares, VCL 10 malwares, Evul.8192 8 malwares	Introduced and examined an opcode statistics based classifier using decision tree.
5	Vinod P, Harshit Jain, et.al [27]	320 viruses from four meta engines	Dynamically analyzed the different API calls made by an executable. After analysis, instead of single sample the signature of entire malware class was generated They further used proximity index between different metamorphic generators to determine how similar two generators were.
6	Quinghua Zhang, Douglas S. Reeves [28]	Variants generated through Vx Heavens website	Proposed a new method for detecting metamorphic malwares through fully automated analysis of executables and by comparing the various system functions and libraries present in the program semantics.
7	Mohamed R.Chouchane, Arun Lakhotia [29]	W32.Evol engine was used to detect engine signatures.	Introduced the concept of engine signatures for detection of metamorphic malware. They used code scoring technique in order to measure how many code segments were generated by a metamorphic engine.