## PROJECT SYNOPSIS

OF

PERFORMANCE EVALUATION AND DETECTION OF MALWARE

Submitted

***in partial fulfillment for the award of the degree***

## BACHELOR OF TECHNOLOGY

**in**

Computer Science and Engineering



Submitted by: Under the Supervision of :

Aayush Kumar , 2001320100002 Mr. Ashwini Verma

Abhishek Saxena , 2001320100010 Assistant Professor, Anjila Choudhary ,2001320100026 Dept. of CSE Azharuddin Alam ,2001320100041

# Greater Noida Institute of Technology, Greater Noida Dr. A.P.J. Abdul Kalam Technical University, Lucknow

June , 2023

**Table of Contents**

**Description Page No.**

[Abstract 1](#_TOC_250001)

1. Introduction 2
2. Literature Review 11
3. Objective 14
4. Methodology 15

[References 16](#_TOC_250000)

## Abstract

In today’s world, information is one of the most valuable assets, but there is a major threat to it by the evolving second generation sophisticated malware, because it can enter the networks, quietly take the confidential data/information from the computational devices, and can cripple the infrastructures, etc. To detect these malware, time-to-time various techniques are proposed. These methods range from the early day signature-based detection to Machine/Deep Learning techniques. Therefore, to understand the evolution of malware and its detection technique, this paper presents an overview of the evolution of malware and it’s detection techniques. It discusses in details the various type of second generation malware, and the popular detection techniques used to detect it viz. signature matching, heuristic methods, normalization, and machine/deep learning techniques.

## Chapter 1 Introduction

The breakthrough in internet technology and computer networking have made high speed shared internet possible. The effect of this development is the daily increase in the number of computer systems that have become susceptible to malware attacks[1][2] . Malware, short for malicious software, is any software intentionally designed to cause disruption to a computer, server, client, or computer network, leak private information, gain unauthorized access to information or systems, deprive access to information, or which unknowingly interferes with the user's computer security and privacy. Researchers tend to classify malware into one or more sub-types (i.e. computer viruses, worms, Trojan horses, ransomware, spyware, adware, rogue software, wiper and keyloggers). Malware poses serious problems to individuals and businesses on the Internet.

The number of new malware on the internet keep on increasing at an alarming rate even as anti-virus companies are making effort to curtail the trend so as to make the vast number of computer user safe. Malware has evolved over time and is becoming more sophisticated than before. It is now more difficult to detect them. There is therefore the need to invent more efficient techniques that can detect and prevent these attacks. Malware is a malicious program which infringes on the security of a computer system in terms of privacy, reliability, and accessibility of data[3] . This trend has made academicians and industry practitioners to move from the conventional static detection techniques[4][5] to more dynamic, sophisticated and spontaneous methods that applies accumulated malware behaviour to detect malware attacks [6][7][8] .

There are two types of malware:-

1. First Generation Malware
2. 2nd Generation Malware

### First Generation Malware

The first generation, i.e., static malware are generally classified on the basis of their infection strategy as Viruses ,Worms and Trojans. Few other notable first generation malware are root-kits, spyware, crime-ware, adware, etc. (fig. 1). They all exhibit different sort of malicious behavior on the target systems, but design of the malware remain unchanged.

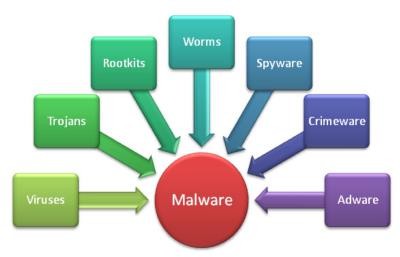


Figure 1: *First Generation Malware*

### Trojan Horse

It is a program that looks harmless and helpful to users like any other authentic software. However, after opening the application, this malware distributes some other malicious codes that corrupt the files and applications installed on the computer, and also steal sensitive information such as password. Unlike computer viruses and worms, Trojans require interaction with users to reproduce themselves. This makes Trojans one of the most destructive and hazardous types of malware because it is mostly discovered after it has affected the computer system [9].

### Virus

Virus as a malware that has a self-replicating nature. It is constructed to modify or stop the functioning of a computer. It multiplies by first infecting one program. It is a kind of malware that can cause serious damage varying from the computer system merely displaying arbitrary errors in making the system experience a Denial of Service (DoS) attack. What distinguishes a virus from a Trojan is the ability of a virus to duplicate itself by attaching itself to other valid software and become a part of them.

### Adware

It is a malware whose only purpose is to show advertisements to the user. They are regarded as one of the least threatening categories of malware. Their intention is to display on the affected computer commercials which the user is likely to be attracted to, it records data from the computer such as browser and search engines histories [10].

### Spyware

It is a kind of self-installing malware that execute without the user’s approval. It is used to gather and track information about the person and the browsing history of a computer system. It is generally packaged together with software that is made available to users at no cost[11]. Spyware is also called rootkit because of the packaging with freeware. Spyware is a code that enables a third party to spy on a host.

### Worm

It is a malware that does not attach itself to other software as it does not need a host software to fasten itself to. This is what differentiates worm from the virus. A worm normally affects its victim through the area of exposures that it can exploit. It employs various means to propagate, and corrupt other computer systems .

### Bot

It is also known as a web robot or botnet. Bots are application software that runs automated tasks over the internet. They belong to a category of malware that allows its principal to gain access to the infected computer system. Bots can propagate through backdoors made available by a virus or worm on the

victim computer.

### Ransomware

It is a subcategory of malware which encrypts the files on the victim’s computer or totally locked you out. It turns your files to unintelligible information and makes them useless and payment is necessitated prior to the decryption and returning of the ransomed files to the owner. They usually infect their victims through Trojan[12] .

### Rootkits

These are a set of software tools used by hackers to get and sustain continuous administrator-level access to a computer system so as to camouflage the changing of files, or activities of the hacker to keep the user in the dark. Rootkits are commonly linked with Trojans, worms, and viruses that obscure their presence and actions from users and other system processes [12].

### Backdoor

It is a class of malware that offers a supplementary stealthy “entrance” to the system for attackers. The backdoor itself does not directly harm the system but it opens the door for attackers to wreak havoc. Due to this characteristic, backdoors are in no way used individually. Ordinarily, a backdoor is antecedent malware attack or other forms of attacks [12].

### Keylogger:

It is also known as keystroke logging. This is a type of surveillance malware that once the computer is infested with it has the ability to record every keystroke make on that system. The recording is saved in a log file which is normally encrypted and sent to a specific receiver. Such information can include passwords, Band Verification Number, ATM card numbers and other confidential information[13].

### 2nd Generation Malware

In the 2nd generation, i.e., dynamic malware after each infection, structure of the malware changes to create a new variant keeping the intent same.On the basis of the mechanism by which either the code or the structure are obfuscated to conceal the signature of the malware, the 2nd generation malware can be further classified as Encrypted, Oligomorphic, Polymorphic and Metamorphic Malware.

### Encrypted Malware

Encryption is the first concealment technique in which malware body consists of a encrypted malicious code, key and encryption/ decryption algorithm (fig. 2). In this, body of the malware are XORed with the generated key to make it difficult to detect. The main objective to create encrypted malware was to evade the static code analysis and traditional signature based detection technique. It infects the system by decrypting itself using decryption algorithm and a key, after that it again encrypt itself by the encryption algorithm and generate a new key for another variant to avoid the detection mechanism.

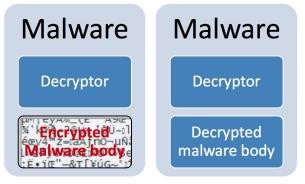


Figure 2:*Encrypted Malware*

### Oligomorphic Malware

Limitation of the encrypted malware, i.e., invariance of the decryptor in the particular malware variants made easy for the anti-malware to detect it by simply finding the signature of the decryptors. Hence, led to the development of various concealment techniques to evade the detection mechanism. In this malware (fig. 3) decryptors are mutated in the malware variant, i.e., it provides set of obfuscated decryptors.

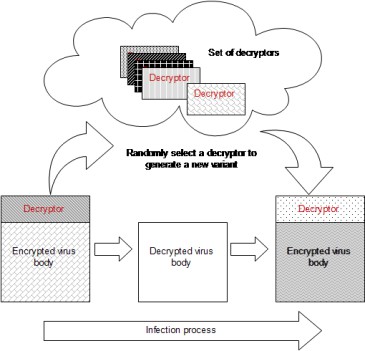


Figure 3:*Oligomorphic Malware*

### Polymorphic Malware

Polymorphic malware are similar to Oligomorphic malware, but it can generate millions of decryptors by mutating the instructions in the variant of the malware [14] to evade the signature matching detection technique. In this malware, mutation engine generates an encryption algorithm and a corresponding decryption algorithm, then malware code and mutation engine both get encrypted to generate a new variant of a particular malware. Figure 4 shows the structure of Figure 4: Polymorphic malware. a polymorphic malware which has two parts, a decryptor and the body of the malware.

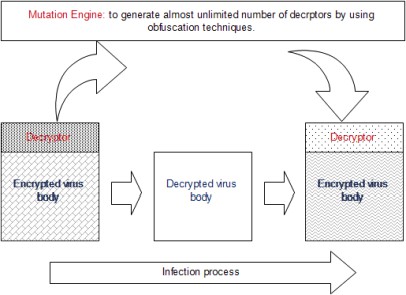


Figure 4: *Polymorphic Malware*

### Metamorphic Malware:

In Metamorphic malware (fig. 5) instead of mutating the decryptors the malware body is mutated itself (i.e., body-polymorphic) to create a new variant without changing its actions to evade the detection [15]. To create the variants similar to polymorphic malware various obfuscation methods viz., dead- code insertion, data modification, control/data flow modification, register renaming, subroutine permutation, equivalent code substitution, etc. are used to get the metamorphic behavior.

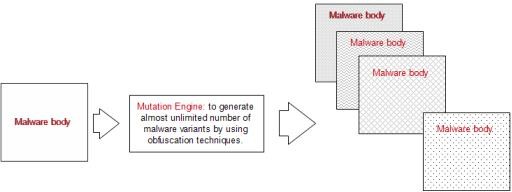


Figure 5: *Metamorphic Malware*

**Project Name: PERFORMANCE EVALUATION AND DETECTION OF MALWARE**

# Project Members:

This project is done in a group of four people. Project members are.

1. Aayush Kumar
2. Abhishek Saxena
3. Anjila Choudhary
4. Azharuddin Alam

## What is Malware Detection ?

Malware detection refers to the process of identifying and recognizing malicious software, commonly known as malware, on a computer system or network. Malware encompasses various types of malicious software, such as viruses, worms, Trojans, ransomware, spyware, adware, and more. The goal of malware detection is to identify and mitigate the harmful effects of malware to protect the integrity, confidentiality, and availability of computer systems and data.

## History of Detection of Malware:

The history of malware detection dates back several decades, starting with the early days of computing. Here's a brief overview of the key milestones and developments in the history of malware detection:

### Signature Based Detection

The traditional signature based detection (fig. 6) is an effective and simple technique to detect the known malware [16]. In this technique after identifying the malware, a unique short sequence/pattern of bytes are extracted to differentiate the malware from the benign programs [17].

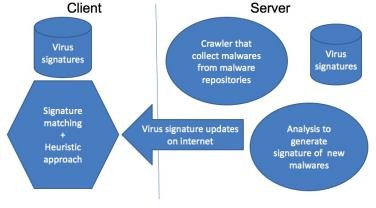


Figure 6: *Traditional detection system*

### Heuristics Based Detection

Between 2000 - 2010 heuristic based detection technique combined with signaturebased detection was a major defense against malware, and heuristic method was one of the promising approach to detect the new or previously unseen malware [18]. In this method, two approaches are used for the identfication. Firstly, in static methods suspicious programs are analyzed to obtain a defined pattern in the program, if any, and if the result crosses the threshold, then the file is said to be infected [19].

### Malware Normalization

By using the sophisticated obfuscation techniques, malware developers have developed automated advanced malware generation toolkits like Zeus, Ultimate Packer for Executable and Mitsfall [20]. These kits can generate thousands of malware in a day which is extremely

hard to detect with signature or heuristic based malware detection techniques. In this method normalize executable/ malware are obtained by removing the obfuscation in the program and then it can be used to increase the detection accuracy of an existing anti- malware (fig. 7) [21].

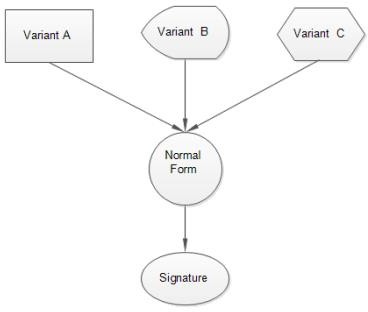


Figure 7: *Malware Normalization*

### Machine/Deep Learning Techniques:

Nowadays, machine learning [22] techniques are widely explored to detect the unknown or previously unseen malware. This technique not only detects the known malware but can also detect the unknown malware by getting knowledge from the previously detected malware. The technique is a two-step process, in the first step feature (e.g., API Calls, N- gram, Strings, Opcodes, Control Flow Graph, etc.) are extracted from the know datasets which plays a vital role, not only to represent the target concept but also to speed-up the learning and classification/detection processes. In the second step, appropriate machine learning techniques viz. Decision Tree [23], Naive Bayes [24], Data Mining [25], Hidden Markov Modes [26], Neural Networks [23], etc. are trained for detection/classification of malware.

## Why Malware should be detected ?

The detection of malware is crucial for several reasons:

1. **Protection of Systems and Data:** Malware can cause significant harm to computer systems, networks, and data. It can corrupt or delete files, disrupt system operations, steal sensitive information, or grant unauthorized access to attackers. Detecting malware helps prevent these malicious activities and safeguards the integrity, confidentiality, and availability of systems and data.
2. **Early Identification of Threats:** Detecting malware allows security professionals to identify and analyze new and emerging threats. By understanding the characteristics and behaviors of malware, they can develop countermeasures and updates to security systems to mitigate the impact of these threats before they spread widely.
3. **Preventing Further Infections:** Malware often spreads by infecting other systems within a network or by propagating through various means, such as email attachments, malicious websites, or removable media. By detecting and isolating malware promptly, further infections can be prevented, limiting the damage and stopping its propagation.
4. **Mitigating Financial Losses:** Malware attacks can lead to significant financial losses for individuals, businesses, and organizations. For example, ransomware attacks can result in extortion demands, while banking trojans can steal financial credentials and cause unauthorized transactions. By detecting malware, organizations can minimize the financial impact and avoid potential legal and regulatory consequences.
5. **Protection of Personal Information:** Malware often targets personal information, such as credit card details, passwords, social security numbers, and other sensitive data. Detecting malware helps protect individuals' privacy by preventing the theft of their personal information and reducing the risk of identity theft.
6. **Maintaining Productivity:** Malware infections can severely disrupt the normal functioning of computer systems, leading to downtime, decreased productivity, and increased IT support costs. By detecting and removing malware, organizations can maintain business continuity and ensure uninterrupted operations.
7. **Compliance with Regulations:** Many industries and jurisdictions have specific regulations and legal requirements related to cybersecurity and the protection of sensitive data. Effective malware detection is essential for complying with these regulations, avoiding penalties, and maintaining the trust of customers and stakeholders.

Overall, malware detection plays a vital role in preventing damage, protecting systems and data, preserving privacy, and maintaining the overall security posture of individuals, businesses, and organizations in the face of the ever-evolving threat landscape.

## Project Scope:

The Main objective is to find the best accuracy of the the malware detection using different machine learning models. Machine learning techniques includes such as Random forest, XGBoost,Decision Tree and many others.

.

## Chapter 2 Literature Review

### Introduction

The first virus was created in 1970 [27] and since then there is a strong contest between the attackers and defenders. This rat-race led to the development of complex malware and its detection techniques. To defend the malware attacks, antimalware groups are developing new techniques. On the other hand, malware developers are adopting new tactics/methods to evade the anti-malware. The complexity of the malware is continuously growing for the military espionage, sophisticated cyber attacks and other crimes, which motivated the academicians and digital investigators to develop the advanced method to combat the threats/attacks from it. To combat the threats/ attacks from the second generation malware, time to time, a number of static and dynamic methods has been proposed by the researchers [28] [29]

[30] [31] [32].

### Survey on the Detection of Windows Desktops Malware

Static and dynamic analysis are the two main approaches applied for the detection of malware [33]. In static analysis, without executing the malware, the codes are analysed to find a malicious pattern by extracting the features/signature such as Ngrams, Application Program Interface (API) sequences, opcodes, data flow, resources, Dynamic-link Library (DLL) usage, etc. Whereas, in the dynamic analysis the program are examined in a runtime environment by monitoring the dynamic behaviours, such as network connections, system calls, resources usage, etc. of the programs. However, in both the approaches selected classifiers are trained with a known dataset to differentiate the benign and malware programs.

In 2001, Schultz et al. [34] using the data mining methods claimed that their framework can detect the new unknown executables twice than the detection rate of traditional signature based techniques. For the classification, they used three different static features viz. Portable Executable (PE), strings and byte sequences. From the data set of 3265 malicious and 1001 benign programs, they extracted the features from DLL residing in the PE files, strings are extracted from the executables that are encoded in program files and the bytes sequences are extracted from the executable file. Finally, n-grams byte sequences were used as input data for the Multinomial Naive Bayes algorithm and obtained the detection rate of 97.76%.

Kolter and Maloof in 2004 [28] uses the techniques of Information Retrieval (text classification) for the detection of unknown malicious executables. They experimented with 1971 benign and 1651

malware programs, selected 255 millions distinct n-grams with variety of inductive methods, including Naive Bayes (NB), Decision Trees (DT) and Support Vector Machines (SVM). Their results from the boosted decision tree outperformed other methods with an area under the Receiver Operating Characteristic (ROC) curve of 0.996. In 2005, Karim et al. constructed a malware phylogeny model using n-perms as a feature to match the possible permuted code [35].

In 2006, O. Henchiri et al. [29] proposed a search method to select generic features for the detection of the computer virus of different families. They used 1512 labeled viruses, 1488 benign program and Iterative Dichotomiser-3 (ID3), j48, Naive Bayes, Sequential minimal optimization (SMO) classifier for the evaluation. Their result outperforms the traditional search methods, and the best detection accuracy obtained was 92.56% by the J48 classifier. They claimed that their approach, which uses a family of non-specific features, performs very well, while existing techniques for detecting the previously unseen viruses perform significantly poor. Their results indicate that around 95% accuracy can be achieved through the use of a training set that contains below 20% malicious file content.

To detect new or previously unseen polymorphic/metamorphic malware, Yanfang Ye et al. [36] in 2008 analysed the Windows API called by Program executables (PE) files to develop a Intelligent Malware Detection System (IMDS) using Objective-Oriented Association (OOA) mining based classification, which is an integrated system consisting of three major modules: PE parser, OOA rule generator, and rule-based classifier.

Siddiqui et al. [37] applied data mining techniques for the extraction of variable length instruction sequence to identify the worms from the benign programs. Their analysis was facilitated by the program control flow information contained in the instruction sequences. From these instruction sequences, they formulated a binary classification problem and built tree based classifiers (Decision Tree, Bagging and Random Forest).

In 2009 Syed Bilal Mehdi et al. [38] advocated the need of sophisticated,efficient, and accurate malware classification techniques that can detect a malware on the first day of its launch, called In- Execution Malware Analysis and Detection (IMAD) that not only identify the zero-day malware without any apriori knowledge but can also detect a malicious process with 90% accuracy while it is executing.

In 2011, Santos et al. pointed out that supervised machine learning is an effective method to detect the unknown malware, but are not efficient because it requires a significant amount of labeled executables for both malware and benign programs. From the analysis they concluded that due to the complexities

of malware, the larger n-gram size yields the higher accuracy, and the proposed feature extraction methods achieves 96.64% accuracy with 4-gram and Support Vector Machine.

In 2013, Santos et al. proposed a method to detect unknown malware, based on the occurrence of opcode sequences, and also described a technique to mine the relevance of each opcode. They experimented with a malware dataset of 13,189 and 13,000 benign executables from different systems and applications viz. text processors, drawing tools, windows games, Internet browsers, etc. In addition, they provided an empirical validation of the method, which is capable of detecting unknown malware, and found that SVM provides an accuracy of 92.92% and 95.90% for one opcode and two opcode sequence length respectively [39].

## Chapter 3 Objectives

The objectives of malware detection can be summarized as follows:

* + 1. **Preparation of Data set:** In preparing a dataset for malware detection, diverse malware samples are collected from public repositories or through analysis. The samples are labeled as malware or benign. Relevant features are extracted, including file attributes, static analysis attributes (headers, opcodes, API calls), dynamic analysis features (system calls, network traffic), and behavior-based features. The dataset is split into training, validation, and testing sets, and preprocessing steps like shuffling and normalization may be applied to ensure data consistency and balance.
    2. **Identify the suitable features:** Suitable features in malware detection can include a combination of static and dynamic attributes. Static features may include file size, file type, header information, opcode frequencies, API calls, and string patterns. Dynamic features can consist of system calls, network traffic, and file activity. Behavior-based features, such as process activity and registry modifications, can also be valuable. Additionally, machine learning-generated features can capture patterns and relationships within the malware samples. Choosing a diverse set of relevant features helps in accurately distinguishing between malware and benign software in the detection process.
    3. **Model prepare, Training and Validation:** In malware detection, the training and validation process involves splitting the labeled dataset into training and validation sets. Features are extracted and normalized, suitable machine learning models are selected, and trained using the training set. The models are evaluated on the validation set, and hyperparameters are tuned to optimize performance. This iterative process is repeated until satisfactory results are achieved. Finally, the selected model is evaluated on a separate testing set to assess its performance on unseen data. This ensures the development of effective malware detection models.
    4. **Accuracy Check:** Accuracy is a commonly used metric to assess the performance of malware detection models. It measures the percentage of correctly classified samples out of the total samples. However, accuracy alone may not provide a complete evaluation due to imbalanced datasets where the number of benign samples significantly outweighs malware samples. In such cases, additional metrics like precision, recall, F1 score, and AUC-ROC (Area Under the Receiver Operating Characteristic Curve) should be considered to account for the trade-off

between correctly identifying malware and minimizing false positives and false negatives.

14

## Chapter 4 Methodology

The performance and evaluation of malware can be a complex task that requires a thorough methodology to ensure accurate analysis and assessment. Here is a general methodology that can be followed:

### Collection:

Obtain malware samples from reliable sources, such as malware repositories or security vendors. Ensure that the samples are representative of different types and variants of malware.

### Preparation:

Set up a controlled environment for analysis. This typically involves using a virtual machine or a dedicated sandbox with isolated network connectivity to prevent the malware from spreading or causing harm.

### Execution:

Execute the malware samples in the controlled environment. Monitor and record its behavior, including any changes to the system, file modifications, network communication, and processes spawned.

### Performance Evaluation:

Assess the performance of the malware based on different metrics. These may include detection rate by antivirus or security solutions, impact on system performance (e.g., CPU and memory usage, network bandwidth), evasion of detection mechanisms, propagation capabilities, or ability to evade sandbox environments.

### Reporting:

Compile the findings and observations from the analysis and evaluation process into a comprehensive report. Document the characteristics, behavior, and classification of each malware sample. Include details about its performance metrics and any notable findings.

# References

1. Sanjay Chakrabortya and Lopamudra Dey. A rule-based probabilistic technique for malware code detection. Multiagent and Grid Systems – An International Journal, IOS Press, 12, 2016, pp. 271–286

271. DOI 10.3233/MGS-160254

1. Y. Zhou, Z. Wang, W. Zhou, and X. Jiang. Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets. in NDSS, vol. 25, no. 4, 2012, pp. 50–52.
2. D. Keragala. Detecting malware and sandbox evasion techniques, SANS Institute InfoSec Reading Room, 2016. URL: https:/[/www.sans.org/r](http://www.sans.org/reading-room/whitepapers/)e[ading-room/whitepapers/](http://www.sans.org/reading-room/whitepapers/) forensics/detecting-malware- sandbox-evasion-techniques-36667.
3. Sharif, M., Yegneswaran, V., Saidi, H., Porras, P., and Lee, W. Eureka: A framework for enabling static malware analysis. In Computer security-ESORICS 2008, pages 481- 500. Springer.
4. Moser, A., Kruegel, C., and Kirda, E. Limits of static analysis for malware detection. In Computer security applications conference, ACSAC 2007. Twenty-third annual, 2007, pages 421-430.
5. Egele, M., Scholte, T., Kirda, E., and Kruegel, C. A survey on automated dynamic malware-analysis techniques and tools. ACM Computing Surveys (CSUR), 2012, 44(2):6.
6. Ahmad, S., Ahmad, S., Xu, S., and Li, B. Next generation malware analysis techniques and tools. In Electronics, Information Technology and Intellectualization: Proceedings of the International Conference EITI 2014, Shenzhen, 16-17 August 2015, page 17. CRC Press.
7. Gorecki, C., Freiling, F. C., Kuhrer, M., and Holz, T. Trumanbox: Improving dynamic malware analysis by emulating the internet. In Stabilization, Safety, and Security of Distributed Systems, Springer, 2011, pages 208-222.
8. Cisco (2015). What is the difference: Viruses, worms, trojans, and bots? Online. <http://www.cisco.com/web/about/security/intelligence/virus-worm-diffs.html>.

[10]. Radu-Stefan Pirscoveanu. Clustering Analysis of Malware Behaviour. Master Thesis Department of Electronic Systems at Aalborg University, 139 pages, 2015.

1. Symantec. What are malware, viruses, spyware, and cookies, and what differentiates them? 2009.Online. [http://www.symantec.com/connect/articles/what-are-malware-viruses-spyware-and-](http://www.symantec.com/connect/articles/what-are-malware-viruses-spyware-and-cookies-and-what-differentiates-them) [cookies-and-what-differentiates-them](http://www.symantec.com/connect/articles/what-are-malware-viruses-spyware-and-cookies-and-what-differentiates-them).
2. Zimba, A., Wang, Z., & Chen, H. Multi-stage crypto ransomware attacks: A new emerging cyber threat to critical infrastructure and industrial control systems. ICT Express, 2018.
3. Kateryna Chumachenko. Machine Learning Methods for Malware Detection and Classification. Bachelor's Thesis in Information Technology. University of Applied Sciences, 93 pages, 2017.
4. Babak Bashari Rad, Maslin Masrom, and Suhaimi Ibrahim. Evolution of computer virus concealment and anti-virus techniques: a short survey. arXiv preprint arXiv:1104.1070, 2011.
5. Babak Bashari Rad, Maslin Masrom, and Suhaimi Ibrahim. Camouflage in malware: from encryption to metamorphism. International Journal of Computer Science and Network Security, 12(8):74–83, 2012.
6. Kent Griffin, Scott Schneider, Xin Hu, and Tzi-Cker Chiueh. Automatic generation of string signatures for malware detection. In Recent advances in intrusion detection, pages 101–120. Springer, 2009.
7. Nhat-Phuong Tran and Myungho Lee. High performance string matching for security applications. In ICT for Smart Society (ICISS), 2013 International Conference on, pages 1–5. IEEE, 2013.
8. David Harley and Andrew Lee. Heuristic analysis–detecting unknown viruses. Technical report, 2007 (Date last accessed 31-May-2018).
9. Kirti Mathur and Saroj Hiranwal. A survey on techniques in detection and analyzing malware executables. International Journal of Advanced Research in Computer Science and Software Engineering, 3(4):422–428, 2013.
10. Ming Xu, Lingfei Wu, Shuhui Qi, Jian Xu, Haiping Zhang, Yizhi Ren, and Ning Zheng. A similarity metric method of obfuscated malware using function-call graph. Journal of Computer Virology and Hacking Techniques, 9(1):35–47, 2013.
11. J-Y Xu, Andrew H Sung, Patrick Chavez, and Srinivas Mukkamala. Polymorphic malicious executable scanner by api sequence analysis. In Hybrid Intelligent Systems, 2004. HIS’04. Fourth International Conference on, pages 378–383. IEEE, 2004.
12. Tom M Mitchell. Machine learning. WCB, McGraw-Hill Boston, MA, 1997.
13. Robert Moskovitch, Yuval Elovici, and Lior Rokach. Detection of unknown computer worms based on behavioral classification of the host. Computational Statistics & Data Analysis, 52(9):4544– 4566, 2008.
14. Mamoun Alazab, Sitalakshmi Venkatraman, Paul Watters, and Moutaz Alazab. Zero-day malware detection based on supervised learning algorithms of api call signatures. In Proceedings of the Ninth Australasian Data Mining Conference-Volume 121, pages 171–182. Australian Computer Society, Inc., 2011.
15. Muazzam Siddiqui, Morgan C Wang, and Joohan Lee. A survey of data mining techniques for malware detection using file features. In Proceedings of the 46th annual southeast regional conference on xx, pages 509–510. ACM, 2008.
16. Gr´egoire Jacob, Herv´e Debar, and Eric Filiol. Behavioral detection of malware: from a survey towards an established taxonomy. Journal in computer Virology, 4(3):251–266, 2008.
17. Peter Szor. The art of computer virus research and defense. Pearson Education, 2005.
18. Jeremy Z Kolter and Marcus A Maloof. Learning to detect malicious executables in the wild. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 470—478. ACM, 2004.
19. Olivier Henchiri and Nathalie Japkowicz. A feature selection and evaluation scheme for computer virus detection. In Data Mining, 2006. ICDM'06. Sixth International Conference on, pages 891—895. IEEE, 2006.
20. Chatchai Liangboonprakong and Ohm Sornil. Classification of malware families based on n- grams sequential pattern features. In Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on, pages 777-782. IEEE, 2013.
21. Zahra Salehi, Ashkan Sami, and Mahboobe Ghiasi. Using feature generation from api calls for malware detection. Computer Fraud U Security, 2014(9):9—18, 2014.
22. Chandrasekar Ravi and R Manoharan. Malware detection using windows api sequence and machine learning. International Journal of Computer Applications, 2012.
23. Ashu Sharma and Sanjay Kumar Sahay. Evolution and Detection of Polymorphic and Metamorphic Malwares: A Survey. International Journal of Computer

Applications, 90(2):7—11, March 2014.

1. Matthew G Schultz, Eleazar Eskin, Erez Zadok, and Salvatore J Stolfo. Data mining methods for detection of new malicious executables. In Security and Privacy, 2001. SUP 2001. Proceedings. 2001 IEEE Symposium on, pages 38—49.IEEE, 2001.
2. Md Enamul Karim, Andrew Walenstein, Arun Lakhotia, and Laxmi Parida. Malware phylogeny generation using permutations of code. Journal in Computer Virology,  2005.
3. Yanfang Ye, Dingding Wang, Tao Li, Dongyi Ye, and Qingshan Jiang. An intelligent pe-malware detection system based on association mining. Journal in computer virology, 4(4):323—334, 2008.
4. Muazzam Siddiqui, Morgan C Wang, and Joohan Lee. Detecting internet worms using data mining techniques. Journal of Systemics, Cybernetics and Informatics, 2008.
5. Syed Bilal Mehdi, Ajay Kumar Tanwani, and Muddassar Farooq. Imad: inexecution malware analysis and detection. In Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pages 1553—1560. ACM,2009.
6. Igor Santos, Felix Brezo, Xabier Ugarte-Pedrero, and Pablo G Bringas. Opcode sequences as representation of executables for data-mining-based unknown malware detection. Information Sciences, 231:64—82, 2013.