Review on Malware and Malware Detection Using Data Mining Techniques

Mustafa A. Ali
Kerbalaa University
Mustafa99.Ali99@gmail.com

Wesam S. Bhaya
Babylon University
Wesambhaya@gmail.com

Abstract

Malicious software is any type of software or codes which hooks some: private information, data from the computer system, computer operations or(and) merely just to do malicious goals of the author on the computer system, without permission of the computer users. (The short abbreviation of malicious software is Malware). However, the detection of malware has become one of biggest issues in the computer security field because of the current communication infrastructures are vulnerable to penetration from many types of malware infection strategies and attacks. Moreover, malwares are variant and diverse in volume and types and that strictly explode the effectiveness of traditional defense methods like signature approach, which is unable to detect a new malware. However, this vulnerability will lead to a successful computer system penetration (and attack) as well as success of more advanced attacks like distributed denial of service (DDoS) attack. Data mining methods can be used to overcome limitation of signature-based techniques to detect the zero-day malware. This paper provides an overview of malware and malware detection system using modern techniques such as techniques of data mining approach to detect known and unknown malware samples.

Keywords: Computer Security, Malware Classification, Data Mining, Viruses, Malicious Software.

الخلاصة

البرمجيات الخبيثة هي اي نوع من البرمجيات او شفرات برمجية التي هدفها سرقة بعض المعلومات الخاصة او بيانات من نظام الكمبيوتر او عمليات الكمبيوتر او(و) فقط ببساطة لعمل المبتغيات غير المشروعة لصانع البرامجيات الخبيثة على نظام الكمبيوتر، وبدون الرخصة من مستخدمي الكمبيوتر. البرامجيات الخبيثة للمختصر القصير تعرف كملور. ومع ذلك، اكتشاف البرامجيات الخبيثة اصبحت واحدة من اهم المشاكل في مجال امن الكمبيوتر وذلك لان بنية الاتصال الحالية غير حصينه للاختراق من قبل عدة انواع من استراتيجيات الاصابات والهجومات للبرامجيات الخبيثة. فضلا على ذلك، البرامجيات الخبيثة متنوعة ومختلفة في المقدار والنوعيات وهذا يبطل بصورة تامة فعالية طرق الحماية القديمة والتقليدية مثل طريقة التواقيع والتي تكون غير قادرة على اكتشاف البرامجيات الخبيثة الجديدة. من ناحية أخرى، هذا الضعف سوف يودي الى نجاح اختراق (والهجوم) نظام الكمبيوتر بالإضافة الى نجاح هجومات أكثر تطوراً مثل هجوم منع الخدمة الموزع. طرق تنقيب البيانات يمكن ان تستخدم لتغلب على القصور في طريقة التواقيع لاكتشاف البرامجيات الخبيثة غير المعروفة. هذا البحث يقدم نظره عامة عن البرامجيات الخبيثة وانظمة اكتشاف البرامجيات الخبيثة باستخدام التقنيات الحديثة مثل تقنيات طريقة تعدين البيانات لاكتشاف عينات البرامجيات الخبيثة المعروفة وغير المعروفة.

الكلمات المفتاحية: امن الكمبيوتر، تصنيف البرمجيات الخبيثة، تعدين البيانات، الفايروسات، البرمجيات الخبيثة.

1. Introduction

Malicious software is any program that causes harm to a user, system, computer, or network, such as Trojan horses, Worms, Viruses, Rootkits,... and Scareware (Honig 2012). These malwares are not exclusive types, (i.e. a particular malicious software has a characteristics of multiple types of malware at the same time). However, as long as the computer system is constantly evolved with increasing use in all areas of modern life, it has become fundamental to the success of the political, economic, military, and personal objectives. Therefore, it is necessary to protect the computer system from security threats.

The rapid increase in the speed of internet connections and the vulnerability in architecture of the Internet networks, in addition to, the fact that the most computer users are novice, that they have a sophisticated computer with high-speed internet

connections, all that lead to facilitates malware to propagate very rapidly and increase of security threats facing the internet today and further abuse are rises.

Current commercial antivirus vendors cannot offer all the protection for computer system because of zero day malwares, consequently zero day malwares need to analyze by malware analysis techniques to create their signatures. The signatures are styled in such way that they been use to catch the malicious code, this approach is called signatures-based. The signatures-based approach has highly accurate detection ratio but it vulnerable in some situations. Like, if a new threats show up, then the expert analysts should make a combat signature for them in order to detect them in future, and these new threats and signatures are not easy to be detected. In addition, there will be a lot of time period between the new threats creation and the signatures to detect that new threat, therefor, computers that protected by traditional signature-based approach are vulnerable to infect. The system that used to detect malicious intent in program is known as malware detection system and it has two tasks: analysis and detection (Saeed *et al*. 2013). Several of detection techniques, which can be used by anti-virus engine to detect malware will be explain in Section IV.

One of the biggest and main problems outstanding in the antivirus community is to innovate manner to detect unknown and new malware. Data mining approach comes to help into malware detection by using its methods, such as Ripper, Bayesian Classification, Decision Tree (DT), Support Vector Machines (SVMs), etc. Since these methods could be used to design and build classifier that can be used to automatically and accurately distinguishing malicious executables from the being executables without run the malicious code. Data mining (DM) algorithms are trained over a dataset to create detection model or rule set, that is also known as a classifier. To build a classifier, we must separate our datasets into a training dataset and a test dataset by one of standard methodology. Training dataset used by the data mining algorithm to build model that will be used to classify unknown programs as benign or malicious. The accuracy of the model or classifiers is determined by apply the test dataset on that model. If the model classifies malware as a benign (uninfected), it will counted and considered as false negative (FN). As well if the model classifies legal software, as a malware, it will counted and considered as false positive (FP). Furthermore, if the model correctly classifies the infected software as a malware, it is counted and considered as true positive (TP), as well if the model classifies legal software, as a benign, it will counted and considered as false positive (TN).

The main objectives of this review paper is to identify malware type and detection techniques in addition to investigate the data mining techniques and their performance that used to detect Malware. This paper is organized as follows, in addition to introduction, this paper contains five section. Section II describes the classification of malware, followed by section III which includes malware analysis technique. Section IV explains the malware detection technique. Section V includes some of existing work and literature reviews of data mining techniques used to detection malware. Section VI includes conclusion of this survey.

2. Malware Classification

This section gives a brief explains for different types of malicious software. We have said that any software that is created to harm or steal the computer system data or operations is termed as malware. Malware is general term used for any malicious software, and it is generally used to describe all of the viruses, worms, spyware etc. Before indulging into malware detection, it is important to describe the various types

of malware and the things that the malware usually does. The following list presents the common types that most malware falls into:

Malware class	Malware Name	Properties and Feature	Operation	Damage
The contagious threat	Virus	Malicious code usually hides within another seemingly innocuous executable program and that autonomously produces copies of itself, which might even modify copies and inserts them into other executable programs or on a victim machine once introduced to the system.	Viruses cannot transmit themselves to a new machine autonomously, the require human intervention. It is transported via storage devices, peer to peer clients or internet.	Performance degradation, destroying data, denial of service (Uppal et al. 2014).
	Worms	A malware program that replicates itself in order to spread across the entire network of computers without user intervention or authorization and it is stand-alone (Sharp 2013). Deceive novice users through using of the attractive title Email.	Worms spread via communication media such as Email, exploit the computers and network vulnerability by using network or computer resources and worms spread via storage devices.	Consume large amount of systems resources and also degradation network performance (i.e. consume bandwidth).
	Spam-sending Malware	It is malicious software that infects a computer system and then uses these computers to send malware or spam to other computers.	It is installed accidentally by careless users or even through the exploitation of security holes.	Degradation internet speed, Emails issues.
The Masked Threat	Trojan Horse	Trojans mask themselves by appearing to be something legitimate. they hide silently on the infected computer machines, while the computers users continue with their usual activities. If a program just bypasses remote access, it is considered a backdoor. But, if the malware authors work to gild these backdoor capabilities as some other legal program, then it considers Trojan horse(Skoudis 2004).	Trojan horse spreads through user interaction by tricks the victim to downloading or opening an e-mail attachment and installing it, then attacks, often providing a rootkit and attacker run the Trojan from the internet. Note it is not self-replicate.	Allows your PC to be remotely controlled by the attacker with no authentication (Honig 2012). Denial of service attack. Install additional malware or monitor user activity. Trojan does not infect a file, i.e. there is nothing to clean, though the AV scan engine may report the file as "uncleanable".
	Botnet	Remotely controlled autonomous software that permit the remotely access to the computer system by attacker. However, all machines that infected with the particular botnet are controlled by a single command-and-control server. Botnet infrastructures consisting of hundreds, thousands, or even millions of computers hosts that are may all under one control of attackers(Sampat & Powell 2012).	Botnets are usually delivered via infected internet web pages, or download links to malicious websites.	It considers as prime illegal activities on the internet today like DDos attacks, spreads further malware. PC remotely controlled by the operator which may direct infected machines to execute a variety of malicious functions.
	Rootkits	A suite of one or more programs that performs masking techniques for malware and conceal the malicious intent from the antivirus and it usually spreads with other malware,	Rootkits can't propagate by themselves, they can be downloaded from the internet through	It is main function is concealing the existence of malicious activities, taking control of

		like a botnet. Rootkits often replacing OS API routines or install themselves as drivers or kernel modules.	infected websites or by a Trojan.	infected machine and changing the computer's configuration. Rootkit-based botnets generate untold amounts of spam.
The Financial Threat	Spyware	It is a term used for programs, which hacks, collects personal information and monitors the user activity without the user knowledge. Spyware sends that information back to the attacker so the attacker can use the stolen information in some disreputable way (P. Vinod, R. Jaipur, V. Laxmi 2009).They do not harm your computer. Instead, they attack you.	It is assembled as a hidden object of shareware or freeware programs that can be installed on user's computer or it could be delivered by internet web sites by the webmaster. Whenever the user simply visits one of these websites, the user's computer will be infected.	Monitor/ Log the user activity performed on a computer or person's internet behavior. collecting personal information, such as, email address, usernames, passwords, user's files key pressed by user.
	Information-stealing Malware	malicious software that gathers personal information from infected user's computer and commonly sends this information to the attacker. Keyloggers and sniffers are example of this type of malware (Honig 2012).	It infect computers when a user simply visits infected internet web site or it can be installed by another malware.	Information-stealing malware used to gain remotely access to usernames, passwords, files and user financial information (Honig 2012).
	Scareware	Malware designed to scare victims by showing fake security warning on their computers, and urges users to buying useless, commercial version of their software to rid bogus. It generally has a user interface that could be look as a legitimate antivirus AV or other security software. It warns computers users that there is a malware on their computers without scanning the victims' file systems. It differs from crude AV in that it doesn't detects malicious software, while crude AV detection quality is not good enough to apply it in practical. (Kasuya 2009)	It can be installed by the user when downloading bogus security software, opening spam attachments, by visiting a malicious website or even from famous download sites that are sometimes exploited. In fact, in 2012, a fake AV sample called RegGenie is distributed. (Kasuya 2009)	It collects all information stored on your computer (financial details, personal info) which could be sold to other cyber criminals and shows a disturbing popup window frequently that reports an unreasonably high number of infections. Fake AV business earns tremendous revenue.(Stone-gross et al. 2011)
	Adware	It is advertising software that automatically shows up or displays advertisements after it is installed or used. It is usually assembled in add-ons to internet explorer softwares and free software (P. Vinod, R. Jaipur, V. Laxmi 2009).	The most common source of adware software are add-ons, peer-to-peer clients like KaZaa, and free games.	It goals is to sale some things via displays or downloads the advertisements to users of computers and that leads to user's ennui.

List 1. Common types of Malware.

All malicious softwares are sometimes loosely termed as virus and also the commercial anti-malware products are commonly called antivirus. Readers may find other, slightly different, definitions in the literature, as the borderlines between. malware classes are variety of other classes which may overlap and blur the

boundaries between these classes (i.e. classes are a bit fuzzy because modern malware may spans multiple classes) (Szor 2005). For instance, a program might have a spyware that collects personal information and a worm component that sends Email spam. Note that we have explained previously that malware can be classified based on its functionality, but we also can classify malware according to the attacker's goal as targeted or mass. Targeted malware is designed to a specific infrastructure or organization, such as Trojan horse. It is a bigger threat to computer system and networks than mass malware, because it is not general and common therefore the security products possibly won't protect computer system and networks from it. Security products need to a detailed analysis of targeted malware, so they can protect computer system and networks against that malware and they may also can remove these malwares. Targeted malware is generally very sophisticated, and deep analysis will be required (Honig 2012). Mass malware is tailored to infect as many computers as possible, like a one-of-a-kind Scareware, It has primary goal that to be the most common, but it is usually easier to detect and less sophisticated and computers usually protected against it because security software targets it (Honig 2012).

3. Malware Analysis Technique

Malware analysis is necessary to develop effective malware detection technique. It is procedure of analyzing functionality and objectives of a malicious software, so the goal of malware analysis technique is to understand how the specific code of malware works so that defense can be built to face these malwares and protect the network and computer system. There are many approaches have been proposed for malware analysis that achieve the same goal which is how malware works and its effects on the system, but the tools, time and skills required to perform these approaches of analysis are very different. Although problem of detecting and classifying unknown and new software as benign or not has been proven to be generally undecidable, detecting malware with an acceptable correct detecting rate is still feasible (Bai et al. 2014). Traditionally, there are two main analysis approaches to detect malicious software: static analysis approach and dynamic analysis approach.

3.1. Static Analysis Approach

Static analysis approach analyzes programs or executable binaries without executing it. The program is break down during static analysis by using different reverse engineering techniques and tools, so as to rebuild the original source code. This process mostly is conducted manually (Bergeron et al. 2001). Reverse engineering tools such as disassembler, debugger and analyzer are used through static approach with various techniques as signature based detection and heuristic detection to extract interesting information, such as size of code section, characteristic of each section, characteristic of file, data structure, used functions and call graphs. Note that applying data mining and modern artificial intelligence techniques on static features to detect unknown malware has achieved a good results and interesting accuracy while keeping low false positives rates.

Pros

- Static analysis process is safe while program inspecting the structure of program.
- Static analysis has low overhead of execution time.
- Static analysis can gather information about malicious behavior in the program and can use this information for future security technique.

Cons

- Process of extract the source code of malware samples is sometimes complicated.
- In order to label suspicious files as malware or benign software, information security experts need to analyze manually suspicious files and it is a time-consuming task.
- Malware writers well-known the limitations of static approach and that will motivate and guide them create malware sample that can thwart static analysis.
- Analysts must have deep and good understanding of functioning of operating system and also should have a good knowledge of assembly language.

3.2. Dynamic Analysis Approach

Dynamic approach is the process of evaluating and analyzing program behavior by running the program code and monitoring the execution in real time. Note that dynamic analysis approach is significantly effective to malware encryption or compression and also it is less vulnerable to code obfuscating techniques(Gadhiya & Bhavsar 2013). Dynamic malware analysis overcomes the limitations of static malware analysis (i.e., compression and obfuscation issues) because it performs during runtime and malware unpacks itself (Gadhiya & Bhavsar 2013).

Pros

- Large scale of programs can be analyzing automatically via dynamic analysis.
- Dynamic malware analysis can see the actual program behavior and it activity

Cons

- Some malware samples can be activated only under specific condition for example certain date, time or action.
- Malware may not show their actual behavior when they detect to be running within a controlled analysis environment
- There is probability of harming the computer, if the analyst doesn't properly isolate the analysis environment.
- Dynamic analysis usually suffers from incomplete program coverage because it looks on only one execution path.

4. Malware Detection Techniques

Malware detection techniques are used to detect the malicious software and protect the computer system from being infected and other system compromise such as protecting it from potential information loss. The software uses these techniques often called as anti-virus (often abbreviated as AV), and sometimes known as anti-malware software. It can be classified into signature detection, behavior detection specification detection (Idika & Mathur 2007).

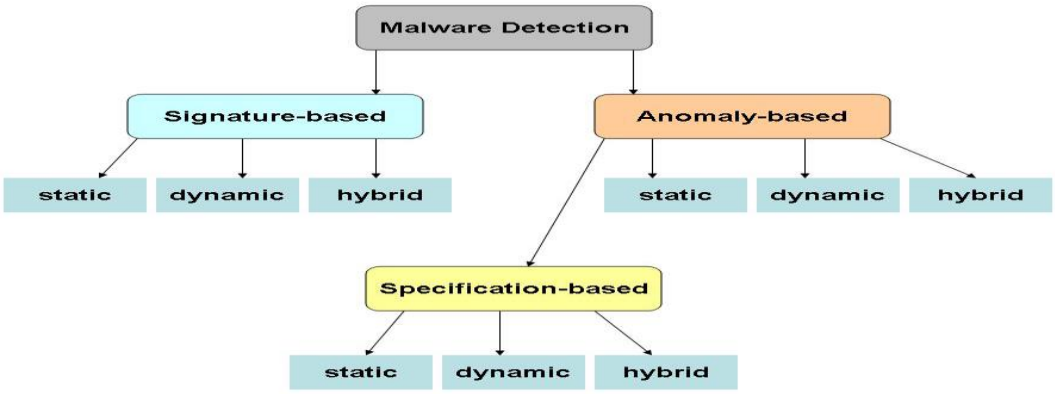

Figure -1: Malware Detection Techniques.

5. Datamining Techniques Used to Detection Malware

Detecting unknown and new malware is a big task today to software security specialists especially that new malwares are generated at average of hundreds every day and form a harmful security threat (Anon 2015; Barossa Community Co-operative Store 2014). So, many of researches have been done in last two decades to detect known and unknown malware using different and various approaches such heuristics, sandbox, data mining algorithms and strategies, and machine learning to reduce the rampant of computer security threats. Machine learning and data mining techniques have been proved to be promised methods that are currently used for the detection of malware as an alternative to the traditional detection methods. The idea of using machine learning and data mining for malware detection is that, they are able to determine the features of a data that is entirely new to their systems or models. This detection is achieved depends on similar sample features that are existing in the model from the training stage. When a set of data with specific characteristics is provided, the model will be capable of determine the class of the new data that entering the model based on the features of these training data set. Researchers discovered good results of applying various data mining techniques to unstructured data such computer machine code, which shows that it is possible to construct accurately and automatically classification system that would be able to distinguish benign computer code from malicious code before they get a chance to run on the system and which therefore could act as an intelligent virus scanner. Data mining algorithms are trained over a particular training dataset, containing samples of both classes, benign and malicious files to build classifiers. A classifier is a detection model, or a rule set which classify a file to a specific class based on its similarity to previous samples of other files. In our case a classifier able to classify a given code as benign or malicious. Data mining have different types of classification techniques that have different characteristic and requirements for example: non-parametric (K nearest neighbor…), mathematical models (neural networks…) and rule based models (decision trees…) …etc. Thus, a dataset prepared for a specific data mining techniques such as a decision tree algorithm might not be appropriate for other data mining algorithm such as K nearest neighbor. Figure -2 shows some of data mining techniques.

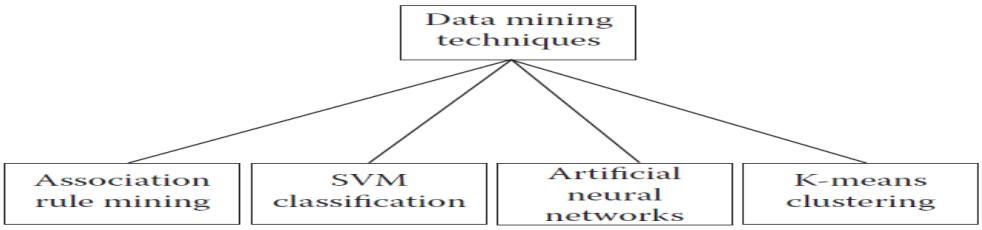

Figure -2: Some of data mining algorithms.

Data preparing is a significant important step in a data mining process (Sung *et al*. 2004). The features are sequences of instructions, n-gram, Opcode n-gram, system call and other features. Various Data mining algorithms like Random Forest, Decision Tree, Association Rule, K nearest neighbor, ... and Naive Bayes have been recommended for classifying and detecting new and unknown files into known malware classes or just determine if file is malicious. Some of literature that use data mining for malware detection are discussed in this section as follow:

Matthew Schultz *et al* (Schultz *et al*. 2001), had the first prominent work using data mining techniques. They introduced data mining models to detect known and unknown malicious executables on Windows OS platform. In the paper, the researchers in the malware detection primarily focused on the static feature extraction (commonly referred to as malware signatures) from executable files and their analysis, and not on dynamic (behavioral) features. They used three different types of statically feature extraction (FE) from the PE files. The first statically feature extraction (FE) was extracted from PE file headers, which were the list of dynamic link library (DLLs) used by the PE file, used function calls in each DLL, and total number of different system calls that used from DLLs. The second feature was the byte sequences n-grams elicitation from a utilities "hexdump" of an PE file. The last feature was string features extracted from the “GNU Strings” program that applied on binaries. The dataset consisted of a total 4266 programs contained 1001 clean program and 3265 malicious program. They used several data mining techniques and algorithms to build models, which were Ripper algorithm, Naive Bayes algorithm, in addition to Multi-Naive Bayes algorithm. Classifiers used to classify PE as malicious or benign programs via a set of features. They applied the Ripper algorithm to the set consist of 244 Windows PE files format. Naive Bayes algorithm, and Multi-Naive Bayes algorithm were applied to the entire PE files collection. Ripper was applied to three different extracted features from the 244 PE files collection, which are (1- List of DLLs, 2- Function calls within each DLL, 3- Total number of different system calls that used from DLLs).

After that, they constructed binary feature vectors for each resource type in the executable based on the presence or absence of that resource. If a given PE used specific DLL, the entry of that DLL in the executable’s vector would be set to one. 229 binary features will be the result from that process, and the second feature (function calls within each DLL) would be encoded in a similar manner as well as third feature (number of system calls for function calls within each DLL), which resulting in 30 integer features. UNIX strings command used as a second technique of feature extraction, which extract all the printable strings in binary file. The researchers inferred, that depend on a highest true-positive rates, thus, the voting naive Bayesian model performed better than all other models. Table 1 showed the accuracy, true-positive rate, and false-positive rate for the researchers' models. However, the curve for the individual naive Bayesian model seemed to master of the voting naive

Bayesian model in generality of Roc Area, showing that the better performance was presently Naive Bayes that used strings as features. However, the researchers are noted, that one had to question the constancy of dynamic link library names, names of function, and strings. For example, one might compile a program by different compiler to generate an executable binaries diverse enough to avoid detection. Softwares were usually obfuscated by Programmers, thus a Dlls or names of used function might not be obtainable. The highest classification performance (provided) over unknown programs yielded by the Naïve Bayes algorithm with an overall accuracy of 97.11%. The researchers implemented a signature detection method as a baseline, and their result showed that all applied models had better results and detection rates for new malware were over double compared with signature detection method.

Table (1) The results that obtained by (Schultz *et al.* 2001).

Method	TP Rate	FP Rate	Accuracy (%)
Signature + hexdump	0.34	0.00	49.31
RIPPER + DLLs used	0.58	0.09	83.61
RIPPER + DLL function used	0.71	0.08	89.36
RIPPER + DLL function counts	0.53	0.05	89.07
Naive Bayes + strings	0.97	0.04	97.11
Voting Naive Bayes + hexdump	0.98	0.06	96.88

Tony Abou-Assaleh *et al* (Abou-Assaleh *et al.* 2004), proposed a model that applied k-nearest neighbor (KNN) classifier with Common N-Gram analysis (CN-G) method to extract and select file features for malicious code detection. Where the idea of this research came from the (CN-G) method generally applied in text classification and natural language processing. By applying manner of one byte at a time of sliding-window on file, the authors gathered Byte n-grams that were overlapping substrings, thus, statistics of substrings of length n and the frequencies of longer substrings were collected. Very frequent N-Gram were produced via N-Gram analysis and it represented signatures. Therefore, n-grams could be used to predict unseen program as malicious or benign program based on features similarity with earlier know sample categories. The features pattern was implied in the selected n-grams. Therefore, virus writers have complex task of writing viruses that can deceives n-gram analyze, although they knew or could access to the detection algorithm. However, class profile generated from the most frequent n-grams with their normalized frequencies which were gathered from training date stage, parameters of the class profiles were the profile length and the n-gram size. Unseen code detected as malicious or benign according to the class that is most similar by use in KNN algorithm with k=1. The researchers' dataset consisted of 40 benign Windows executable and 25 worms taken from infected emails. The researchers' results were average accuracy of 98% with 3-fold cross-validation and accuracy of 100% for the training with some parameter arrangement.

Cumhur Bozagac (Bozagac 2005), takes Schultz (Schultz *et al*. 2001) framework of data mining techniques and applied one of these techniques to figure out effectiveness against new spyware dataset collected in 2005. There was no spyware at that time when Schultz work published. Bozagac nominated Multi-Naïve Bayes algorithm, so he skipped the other different algorithms that Schultz was applied, because according to his thought they were not suitable to detect new spyware. Multi-

Naïve Bayes algorithm is essentially a group of Naïve Bayes algorithms. He used byte sequences in a file as features, that same as the Schultz work [15], but just single one of Naïve Bayes algorithm. His collected dataset contained 926 sample that included 614 executables spyware and 312 benign executables. Hexdump tool was used to extracted byte sequences for each file in the dataset. The number of sequences of byte data can be determined by Naïve Bayes and with default size of window Naïve Bayes took two bytes for frequency analysis. Nevertheless, the user could specify a "size of window" when the algorithm was start run. To evaluate the system he interested in several measures: Detection Rate, Overall Accuracy, True Positives , True Negatives, False Positives, and False Negatives just like Schultz work (Schultz et al. 2001). Naive Bayes algorithm was evaluated, via running the algorithm with and without Trojans for various size of window using 5-fold cross validation as showing in Table 2. He concludes that data mining based heuristic scheme had the potential to be used for detecting new spyware. These best schemes provided an overall accuracy of 91.28% without Trojans and using window size of four as shown in Table (2).

Table (2) The results that obtained by (Bozagac 2005).

	TP	TN	FP	FN	Detection Rate	False Positive Rate	Overall Accuracy
Window Size=2	118	49	14	5	95.93	22.22	89.78
Window Size=4	119	46	17	4	96.75	26.98	88.71
Window Size=2 w/o Trojan	96	46	13	15	86.49	22.03	83.53
Window Size=4 w/o Trojan	99	58	3	12	89.19	**4.92**	**91.28**

He concluded that specific spyware class which was Trojans had a very low detection rate and reason for the high false positive rate, because Trojans had large size compared to other files in the dataset and also it was very complex. Furthermore, he concluded that larger window sizes had better overall accuracy as shown in Table (2).

S. Moskovitch et al (Moskovitch *et al.* 2008), introduced a study that presented a methodology for applying several classifiers to detect of unknown malicious code. They were able to collect large data set that containing more than 30,000 malicious and benign executables, which was the largest test collection currently reported. Their binary code of executables represented by n-grams byte sequence. They implemented several evaluation methods involving eight classifiers and three feature selection methods with investigation on the imbalance problem (i.e. there are large number of sample from one class comparative to other classes) in real life situation, in which the ratio of malware is less than 10% according to recent surveys, but they also considered other percentages in their work. Highest of 95% accuracy can be reached by using training data set that consisted of less than 20% malicious files as their result showed. After extensive and precise experiments to evaluating these classifiers on various number of malware ratio in both the test sets and the training set, best results were achieved when there were similar percentage in both training set and test set. They conclude that it should consider the expected low levels of percentage of existing malicious programs relatively to Benign programs, and the design of training set must be as real-life situation.

Muazzam A. Siddiqui *et al* (Siddiqui 2008), presented Data Mining techniques to detect malwares. Their work was similar to classification techniques and information retrieval with consideration to extract best features and construct classifier that could determine whether the given program as malware or clean programs. Two distinct types of experiments were used. The supervised learning was the first experiment that used a set to train, validate and test, an array of classifiers. They introduce sequential association analysis for feature selection and automatic signature extraction as a second experiment. researchers applied variable length instruction sequence. they collected data set contained 2,775 Windows PE files format, which include 1,330 benign and included 1,444 worms. They addressed and performed detection of crypto, compilers, and common packers first, then they run the process of PE files disassemble. Almost 97% of the sequences were removed by sequence reduction process. Several of data mining algorithms were used such as Random Forest, Decision Tree, and Bagging models. Random forest achieves as1.9% false positive rate on new malware and also it was able to perform as high as 98.4% detection rate, thus can be considered slightly better than the others.

Wang et al (Wang *et al.,* 2009), introduced static analysis method to exploit the information in PE headers for the detection of malware. This work was based on the assumption that there would be difference in the characteristics of PE headers for malware and benign software as they were developed for different purposes. Their detection model included four stages, which were attribute extraction, attribute binarization, attribute elimination, feature selection and classifier training. They performed tests on a dataset that consisted of 9771 executables which included 7863 malicious and 1908 benign executables. The malware samples contained viruses, email worms, Trojans and Backdoors. They collected most of the benign executables from XP OS and Windows 2000 OS in addition to several common user programs that downloaded from well-known internet web site called PChome. PE headers were dumped using a program called DUMPBIN of all the files. Every header in the PE was considered as a potential attribute. Every field in the dataset was converted to binary value in the attribute binarization process. In elimination stage unimportant and redundant attributes were eliminated. All executables files were converted to Boolean vectors according to the residual attributes after the previous elimination stage. Support Vector Machines was used for classify executables as malicious or benign, and the accuracy of classification was calculated by using 5-fold cross validation training method. Their experiment results were without execution feature selection as an overall accuracy, 89.54%, 98.19%, 93.96%, and 84.11% were calculated for backdoors, virus, email worm, and Trojans respectively, after eliminating redundant features the results were 89.93%, 98.23%, 94.07%, and 84.20%, and for backdoors, virus, email worm, and Trojans respectively. although most of modern malware used packer and/or obfuscation techniques, the research hadn't discussed the impact of packing on the executable.

Veeramani and Nitin Rai. (Veeramani & Rai 2012), introduced a framework for malware detection that followed the static analysis approach to analyze and classifying PE executable by mining relevant system call functions (API calls) from malicious executables. The researchers illustrated their application mechanisms and components that involved to make the framework fully automatic for mining API calls. The researchers formed a dataset consist of 210 variety malicious executables from VX Heavens website and 300 benign executables from system32 folder in Windows XP system, where all executables in PE format. In statistical analysis, they considered the proper identification and unpacking of packed malware. After

unpacking the malware executables, IDA Pro tool was used to disassemble the binary file to analyze and extract the Windows API statically. In addition, they used idapython plugin, which facilitates to run the disassembly module automatically for generating 16 tables for each binary executable. Every one of these tables held various information concerning content of binary. All the non-recognizable function names, recognizable API system calls, and the location length of each function ware stored in function table. We extracted the list of API calls using Function table. Microsoft Developer Network (MSDN) Reference is used for matching and in identifying the windows API's. Furthermore, Document Class Wise Frequency feature selection measure (DCFS) was used to get the relevant API calls from the mined API calls to rise the classification and detection accuracy. The aim was that identify a set of API calls that were common used by set of malware likewise identified another set of API calls that were common used by set of benign programs. The researcher used relevant API calls and SVM algorithm to build classifier that could determine whether a given program was benign or malicious. Their experiments were performed on various size of n-gram on SVM classifier. Experiments results were shown in Table (3).

Table (3) Experimental Results for Various Size of N-Grams of (Veeramani & Rai 2012).

Size of n-gram	*Accuracy*
1	97.23 %
2	94.47%
3	93.96 %
4	91.70%

Santos *et al.*, (Santos *et al.* 2013), suggested an hybrid supervised malware classification models that called "OPEM", which could detect unknown malware. It used a set of features extracted from both dynamic and static analysis of malware. Where the Static set of features were frequency of occurrence of operational codes and it extracted without executing the sample while dynamic features were information of the execution trace of an executable. New hybrid representation of executables composed from both static features that extracted by modeling an executable as a sequence of operational codes of a fixed length and calculated their frequencies to generate a vector of frequencies of opcode sequences. In addition, dynamic features that extracted by monitoring system calls, operations and raised exceptions on an execution within an emulated environment to finally generate a vector of binary characteristics representing whether a specific comportment was presented within an executable or not. The approach was then validated over two different data sets: a malware dataset that included 1,000 malicious programs and a benign software dataset that included 1,000 legitimate executables.

They produced opcode-sequence representation for each executable in that dataset for a opcode-sequence with different length. They noted that opcode-sequence with length equal to two generated very high number of features: 144,598 features. Therefore, they used a feature selection method that used Information Gain, to select the top 1,000 features. They extracted the dynamic characteristics for the malware and benign executables by monitoring it in the emulated environment, where the number of features was 63. Researchers, combined these different two dataset features into

one dataset, and thus creating a hybrid static-dynamic dataset. Their result showed that the hybrid approach improved the performance of both approaches when run separately for different learning algorithms such as K-nearest neighbor, Support Vector Machine, Decision Tree, and Bayesian network.

6. Discussion and Conclusion

This paper presented that, data mining technologies have significantly spread, since the beginning of the new century. The developments in information technologies and the exploded amounts of generated data have resulted an increasing need of data mining. Data Mining involves promising means to analyze and uncover hidden knowledge within potentially large amounts of data in addition to predict future behavior. Therefore, it is being used in many applications for security including detecting and classifying malwares as well as for cyber security. On other hand, malware technologies have also exploded. There are several data mining algorithms that can be used to detect and classify malware. As a result, there is now a critical need to develop new DM methodologies and algorithms that are scalable, fast and flexible for detecting and classifying malware as well as transforming raw data into the useful information to secure systems. However, first of all, good data is the primary requirement to better data exploration, because these algorithms are as worthy as the data that has been collected. Next step is to select the most efficient techniques to mine the data. Furthermore, there are characteristics must be considering while choosing the suitable data mining algorithms and methods to be used in a particular purpose. There are obvious differences in the types of fields and problems that are conductive for each algorithm. The best model is often found by trial and error: trying different algorithms and techniques that should applied with caution. Sometimes, in order to obtain the best possible results, the researchers should be compared or even combined data mining techniques. This paper introduced review for Malware Classification, Malware Analysis Technique, Malware Detection Technique. In addition to some existing techniques for detecting and classifying malwares using data mining, where we explain various facts of the detection challenge, such as feature selection methods, file representation, classification algorithms, and the imbalance problem. We show the summary of research that previously discussed in Table (4).

Table (4) Summary of related work on malware detection using data mining techniques.

Title	Author Name and Reference	Year	Malware Types	Techniques (algorithms)	Analysis Method	Features	Result	Description and Discussion
Data mining methods for detection of new malicious executables	Matthew G. Schultz et al, (Schultz et al. 2001).	2001	Malicious executables on Windows OS platform	Ripper algorithm, Naive Bayes algorithm, Multi-Naive Bayes.	Static	String. List of DLLs, function calls within each DLL, total number of different system calls that used from DLLs. N-grams.	The highest overall accuracy of 97.11% yielded by the Naïve Bayes algorithm.	first prominent work done using data mining techniques.
N-gram-based detection of new malicious code	Tony Abou-Assaleh et al, (Abou-Assaleh et al. 2004).	2004	Malicious code such as worms.	k-nearest neighbor (KNN) classifier with k=1.	Static	Common Byte N-Grams	Average accuracy of 98% with 3-fold cross-validation and accuracy of 100% for the training with some parameter arrangement.	Virus writers have complex task of writing viruses that can deceives of n-gram analyze, although they know or can access to the detection algorithm.
Application of Data Mining based Malicious Code Detection Techniques for Detecting new Spyware	Cumhur D. Bozagac, (Bozagac 2005).	2005	Executables Spyware	Multi-Naïve Bayes algorithm	Static	Hex dump tool is used to extracted Byte sequences for each file in the dataset.	These best schemes provide an overall accuracy of 91.28% without Trojans and using window size of four.	Trojans have a very low detection rate and reason for the high false positive rate. Furthermore, he concluded that larger window sizes have better overall accuracy
Unknown malcode detection and the imbalance problem	S. Moskovitch et al, (Moskovitch et al. 2008).	2008	Malicious executable	Artificial Neural Networks, Decision Trees (DT), Naive Bayes, Boosted DT, Boosted Naive Bayes, Support Vector Machines	Static	N-grams byte sequence	Highest of 95% accuracy can be reached by using training data set that consists of less than 20% malicious files.	They were able to collect largest data set that currently reported, with investigation on the imbalance problem where the ratio of malware was less than 10% according to recent surveys, but they also consider other percentages in their work.

Data Mining Methods for Malware Detection	Muazzam A. Siddiqui et al, (Siddiqui 2008).	2008	Malwares	Random Forest, Decision Tree, and Bagging models	Dynamic	Investigation of data mining methods for malware detection	Random forest achieve a 1.9%false positive rate on new malware and also it was able to perform as high as 98.4% detection rate	They addressed and performed detection of crypto, compilers, and common packers first, then they run the process of PE files disassemble
Detecting unknown malicious executables using portable executable headers	Wang et al, (Wang et al. 2009).	2009	Malicious executables viruses, email worms, Trojans and Backdoors.	Support Vector Machines	Static	IDA Pro tool used to disassemble the binary file to analyze and extract the Windows API statically.	After eliminating redundant features the results were 89.93%, 98.23%, 94.07%, and 84.20%, and for backdoors, virus, email worm, and Trojans respectively.	Although most of modern malware use packer and/or obfuscation techniques, the research has not discussed the impact of packing on the executable.
Windows API based Malware Detection and Framework Analysis	Veeramani R and Nitin Rai., (Veeramani & Rai 2012).	2012	Malicious executables malware	Support Vector Machines	Dynamic	Mining relevant system call functions (API calls)	SVM with One n-gram size achieve a 97.23 % detection rate	The researchers make the framework fully automatic for mining API calls
OPEM: A static-dynamic approach for machine-learning-based malware detection	Santos *et al*, (Santos et al. 2013).	2013	Malicious executables malware	K-nearest neighbor, Support Vector Machine, Decision Tree, and Bayesian network.	Hybrid	Set of features extracted from both dynamic and static analysis of malware.	Hybrid approach improved the performance of both approaches when run separately for different learning algorithms	They note that opcode-sequence with length equal to two generated very high number of features. Therefore, they used a feature selection method to select the top features.

References

Abou-Assaleh, T. *et al.*, 2004. N-gram-based detection of new malicious code. *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, 2(1).

Anon, 2015. McAfee Labs Threats Report users are still exposed to. , (February).

Bai, J., Wang, J. & Zou, G., 2014. A malware detection scheme based on mining format information. *Scientific World Journal*, 2014.

Barossa Community Co-operative Store, 2014. Annual Report 2014. , pp.1–28.

Bergeron, J. *et al.*, 2001. Static Detection of Malicious Code in Executable Programs. *Control*, pp.184–189. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.6845&rep=rep1&type=pdf.

Bozagac, C.D., 2005. Application of Data Mining based Malicious Code Detection Techniques for Detecting new Spyware. *Bilkent University*, pp.1–8.

Gadhiya, S. & Bhavsar, K., 2013. Techniques for Malware Analysis. , 3(4), pp.972–975.

Honig, M.S. and A., 2012. *PRACTICAL MALWARE ANALYSIS*,

Idika, N. & Mathur, A., 2007. A survey of malware detection techniques. *Purdue University*. Available at: http://cyberunited.com/wp-content/uploads/2013/03/A-Survey-of-Malware-Detection-Techniques.pdf.

Kasuya, M., 2009. Distinguishing Legitimate and Fake / Crude Antivirus Software. , (c), pp.109–116.

Moskovitch, R. *et al.*, 2008. Unknown malcode detection and the imbalance problem. *Journal in Computer Virology*, 5, pp.295–308.

Vinod, P.; Jaipur, R.; Laxmi, V. and M.G., 2009. Survey on Malware Detection Methods. *Hack in 2009*, p.74. Available at: http://www.security.iitk.ac.in/hack.in/2009/repository/proceedings_hack.in.pdf#page=82.

Saeed, I.A, Selamat, A. & Abuagoub, A.M.A, 2013. A Survey on Malware and Malware Detection Systems. , 67(16), pp.25–31.

Sampat, C.K. & Powell, C., 2012. Red Book. , p.26.

Santos, I. *et al.*, 2013. OPEM: A static-dynamic approach for machine-learning-based malware detection. *Advances in Intelligent Systems and Computing*, 189 AISC, pp.271–280.

Schultz, M.G. *et al.*, 2001. Data mining methods for detection of new malicious executables. *Proceedings 2001 IEEE Symposium on Security and Privacy SP 2001*, 9, pp.38–49. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=924286.

Sharp, R., 2013. An Introduction to Malware Classification of Malware. *Network*, pp.1–28.

Siddiqui, M., 2008. Data mining methods for malware detection. Available at: http://books.google.com/books?hl=en&lr=&id=lZto6RraGOwC&oi=fnd&pg=PR15&dq=DATA+MINING+METHODS+FOR+MALWARE+DETECTION&ots=kPiUFX5bnu&sig=PN3xl8xGeaFGB58tnD0AU3YUF-M.

Skoudis, E., 2004. *Malware: Fighting Malicious Code*, Prentice Hall Professional. Available at: https://books.google.com/books?id=TKEAQmQV7O4C&pgis=1 [Accessed February 27, 2015].

Stone-gross, B. *et al.*, 2011. The Underground Economy of Fake Antivirus Software. *Economics of Information Security and Privacy III*, pp.55–78. Available at: http://www.springerlink.com/index/10.1007/978-1-4614-1981-5.

Sung, A.H. *et al.*, 2004. Static Analyzer of Vicious Executables (SAVE). *Proceedings - Annual Computer Security Applications Conference, ACSAC*, pp.326–334.

Szor, P., 2005. *The Art of Computer Virus Research and Defense*,

Uppal, D., Mehra, V. & Verma, V., 2014. Basic survey on Malware Analysis, Tools and Techniques. *International Journal on Computational Science & Applications*, 4(1), pp.103–112. Available at: http://www.airccse.org/journal/ijcsa/papers/4114ijcsa10.pdf.

Veeramani, R. & Rai, N., 2012. Windows API based Malware Detection and Framework Analysis. *... Conference on Networks and Cyber Security*, 3(3), pp.1–6. Available at: http://www.academia.edu/download/30183099/Conference_Proceedings_book_with_links.pdf#page=45.

Wang, T.Y., Wu, C.H. & Hsieh, C.C., 2009. Detecting unknown malicious executables using portable executable headers. *NCM 2009 - 5th International Joint Conference on INC, IMS, and IDC*, pp.278–284.