Behavioral-based malware clustering and classification

1. Izzat Alsmadi
Texas A&M University, San Antonio
ialsmadi@tamusa.edu
2. Bilal Al-Ahmad
The University of Jordan
b.alahmad@ju.edu.jo
3. Iyad Alazzam
Yarmouk University
eyadh@yu.edu.jo

***Abstract*— Detection of malwares and security attacks is a complex process that can vary in its details, analysis activities, etc. As part of the detection process, malware scanners try to categorize a malware once it is detected under one of the known malware categories (e.g. worms, spywares, viruses, etc.). However, many studies and researches indicate problems with scanners categorizing or identifying a particular malware under different categories. There are different reasons for such challenges where different malware scanners, and sometime the same malware scanner, will categorize the same malware under different categories in different times or instances. In this paper, we evaluated this problem summarizing existing approaches on malware classification.**

Keywords—Malware: Detection, Classification, and Category

I. INTRODUCTION

Different methods are employed in the process of malwares' detection such as: signature/dictionary-, rule- and behavioral-based methods. Signature-based detection method (e.g. using hashes for known files or malwares) is widely used due to its efficiency and robustness in analyzing a large volume of files within a short amount of time. However, there are many obstacles in expanding this method to detect all types of malwares especially in unknown territories, new malwares, or old malwares with slightly different signatures.

Anomaly or behavioral-based detection methods can complement signature-based methods as they are slow but effective in unknown territories.

Collecting attributes to uniquely identify and distinguish malwares is not a trivial process. Attributes to define and distinguish malwares are many, not universal or widely agreed-upon, exist in different artifacts or locations, can be static or dynamic and can be interpreted by different scanners differently. Malware writers sometimes insert garbage calls in order to confuse the analyst with fake API calls or other useless features that can confuse the detection process and impact its decision and accuracy. They may also encrypt or package certain details in which collecting information about such details will be hard or impossible. Those are examples of the challenges to perform data analysis activities in malwares (e.g. clustering, classification, prediction, etc.).

There are many public malware scanners such as AutoShun, PhishLabs, Kaspersky, StopBadware, Sophos, or Netcraft. Testing output from the different malware scanners indicate that frequently they may have different decisions on the same files. It indicates that they employ also different detection techniques or methods as in (Akour et al., 2017).

Figure 1 shows a sample scanning result for a popular Trojan. Although it is a popular Trojan, yet (1) 9 malware scanners indicate this file as "clean", and (2) only 25 of the 67 malware scanners that identify this file as malware, clearly indicate that its type is a Trojan. Malware scanners have to complete scanning systems with usually a large number of files. They have to analyze each subject or suspect file with static and possibly dynamic methods. They have to be also accurate and avoid different cases of false positives and negatives. Those are also other examples of challenges facing malware scanners and all data analysis activities related to malware analysis and detection.

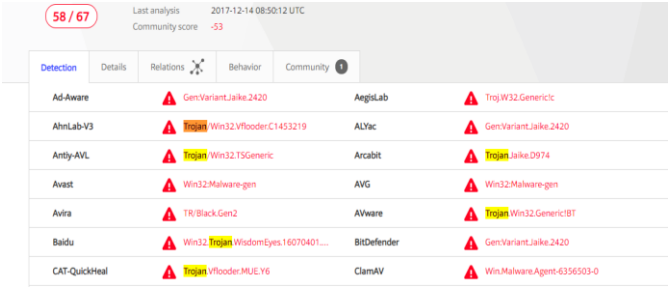

Fig. 1. A sample scanning result for a popular Trojan

Malwares can be classified under different major categories such as: viruses, worms, spywares, ransomwares, etc. Some of the main characteristics that can be used to distinguish those different groups from each other include:

- Payloads: By default, malwares contain a sort of harmful payload to achieve. For example, viruses cause files corruptions or destructions, worms consume machines bandwidths and resources, ransomwares encrypt victims' data, spywares spy on victims' activities, and so on. Some malwares may have different payloads at different stages. Some other large and complex malwares can have several payloads.
- Access or intrusion method: Different malwares have different mechanisms to make their first access to victim machines. Some malwares use other malware types only in their access stage. For example, a worm may use a Trojan method to reach victim machines and control them remotely.
- Propagation: Malwares perform different workflows from the moment of access of targets to the moment of payload deployment. (Saeed et al, 2013) paper shows a table of different categories of malwares and examples of unique attributes related to distinguishing criteria such as: Creation techniques, execution environment, propagation media and negative impacts.

In this paper, our main goal is to use data analysis to group or cluster malwares based on some attributes or behaviors and eventually compare such groups with existing groups related to known malware categories.

The rest of the paper is organized as the following: Section two focuses on research questions, goals and methodologies. In the third section a selection of relevant research publications is discussed. Section 4 focuses on experiments and analysis. Finally, paper is concluded in a small summary or conclusion section.

II. How do malware scanners decide a malware category?

The analysis of the malicious program is important to extract features, which describes the risk and the malware type; there are three types of detection and analysis method to identify the malware categories: (1) static analysis detection technique, (2) dynamic analysis detection, and (3) hybrid analysis detection technique.

First, Static analysis detection technique (Imtithal A.saeed, 2013) (Smita Ranveer, 2015) (Ekta Gandotra, 2014) (Dolly Uppal, 2014) (Balaji Baskaran, 2016) (Saba Arshad, 2017) (P. V. Shijoa, 2015), these studies described the static analysis that is analyzing software malicious and extract the features in the binary code or internal structure of the file without executing, the application is break down by some tools and techniques to rebuild the source code and algorithm of the application that created, it is done during program analyzer and debugger, this type is safe and fast. There are different static analysis techniques: Specific detection (e.g. signature or hash-based detection), and heuristic behavioral detection. Specific detection works by looking for known malware by a specific set of attributes (Imtithal A.saeed, 2013) (Smita Ranveer, 2015) (Dolly Uppal, 2014) (Saba Arshad, 2017), these approaches indicated that known malware have known signatures recorded in a database that can be applied on subject or tested files, but it cannot detect an unknown malware. While Heuristic behavioral detection: It called proactive technique, it is similar to signature based but it does not use searching for signature in code, it search for the instruction that is not appear in the application program (Imtithal A.saeed, 2013) (Dolly Uppal, 2014) they proposed that this process scans for previously unknown malware by looking for known abnormal or suspicious behaviors. Anomaly-based detection depends on monitoring system activities and classifying the subject as either normal or anomalous accordingly.

Heuristic detection technique (Dolly Uppal, 2014) have different types such: (1) File based heuristic analysis file based or file analysis Heuristic system, it analyzes the file completely and check if there is any command in the file can delete or harm other files, it will be considered as malicious. (2) Weight based heuristic analysis, this is the oldest technique, each application have a danger weight or value, and there is a threshold value, if the weight override the threshold value the application will contain a malicious code, (3) Rule based heuristic analysis at this type, the analyzer extract the rules of the application, and match it with the previously defined rules. Such if there is any mismatching then the application contains malware, and (4) Generic signature analysis, this process looks for malware by behaviors of known categories or that are variants of known categories. different behavior for the malware but belongs to the Same category used to discover new variant of malware, for example, statistical-based techniques apply statistical models on system activities such as network connections, bandwidth, memory usage, system calls, etc. which can be usually used by malware. Apparently, false positive cases are common in such scenarios where many “good” applications or system behaviors can be mistaken as malicious activities.

Second, dynamic analysis detection technique (Ekta Gandotra, 2014) (Dolly Uppal, 2014) (KIMBERLY TAM, 2017) (Saba Arshad, 2017) discuss that the dynamic analysis observes the result after executing the program by Analyzing the behavior or the Action of the application but it takes time as the executing time of applications. It interacts with the system while execution in VMware, Simulators and sandbox to find if the executable file is malware or not

Third, hybrid analysis detection technique (Ekta Gandotra, 2014) (Dolly Uppal, 2014) (Balaji Baskaran, 2016) (KIMBERLY TAM, 2017) they proposed that the hybrid analysis is combination of static and dynamic techniques by checking if there is any malware signature in the code. Then observe the code behavior.

In addition, detection can be also classified (Kyoung Han, 2014) into three categories: (1) Host based intrusion detection system, (2) network-based intrusion detection system, and (3) Hybrid intrusion Detection system. Host based intrusion observes and controls the dynamic behavior and the computer system state to check if there is any internal or external Activities that cheats the system policy. Network-based intrusion detection system analyzes all the packets in the network node. Host-based and network-based detection based on the sources of artifacts used in the analysis and detection processes. Hybrid intrusion Detection system: A combination of host- and network-based intrusion detection system is also possible especially with large and complex malware.

Dynamic detection methods run the suspect file in an isolated environment. The research study (Imtithal A.saeed, 2013) proposed isolated environment, sandboxes, where special APIs connect suspect file to the Virtual Machine (VM).From a data science perspective, malware detection process is a typical classification process of two stages: In the first stage, subject file, traffic, hash, etc. will be classified as either a malware or benign. Many malware scanners leave a third category: undecided if the subject fails to be allocated to either one of the malwares/benign categories. For example, many of the newly discovered malware may fall under this (unknown/undecided) category. In the second stage, if the subject is classified as a malware, different categories of malware are available and the process is to allocate this subject to one, or more of those categories.

III. What are the main features in attributes or categories of features that can be used to detect malwares and malware types?

Feature extraction (Mansour Ahmadi, 2016) (Yanfang Ye, 2017) Indicates that the malware detection and classification need to extract some features from (*. byte with Hexadecimal view) and (*. asm file form assembly view). The Portable executable (PE) header describes the basic information and it is rich of information for feature

extraction. It has two essential types: (1) Features from Hexadecimal files, and (2) features extraction from assembly files. There are several features used in Hexadecimal files such: (1) N-gram, (2) metadata, (3) Entropy, (4) image representation, and (5) string. N-gram is a sequence substring with N items, it is used in many fields with characterizing sequence. The study (Mansour Ahmadi, 2016) describes that N-gram means N-byte the malware sample represented as a sequence of hex values and described through n-gram analysis to give information about the malware type. The byte in the binary code contain (2^8) 256 value, by adding special symbol (??) it will become 257 different values, the"??" indicates that there is no mapping for the corresponding byte in the executable file, this value can be discarded. an example the 1 gram feature which represent the byte frequency that described in 256 dimensional vector, in (Meena, 2011) author proposed an approach to detect the Malicious code with n-gram that extracted from Portable execution of malware sample as a feature, (Zhang Fuyong, 2017) proposed classification method based On the similarly of n-grams attribute for malware detection. Also, Metadata it extracts and summarize the basic information of the file, such as file size, address, date modified or date created, (Mansour Ahmadi, 2016) describes that the PE header gives us a meta data information for the executable file. The study (Kun Wang, 2016) proposed a malware detection method depends on metadata of App that extracted from APK file and build vector space for datasets. In addition, the approach in (Mansour Ahmadi, 2016) explained that the entropy technique based on describing the measurement of disorder, it is a numerical measure, the study (Jared Lee, 2015) used the entropy information to interact with metamorphic detection problem, computing Entropy become on the byte level representation of the Malware sample, to measure the disorder in the byte distribution in byte code.

Moreover, image representation (Mansour Ahmadi, 2016) show that the malware sample can be visualized the byte code and explain each byte as gray-level of pixel in the image, this technique used in visual signature By patterns similarity (Kyoung Han, 2014) proposed a method for classification using image representation by converting binary files into images and entropy graph. As in the studies (Mansour Ahmadi, 2016) (Yanfang Ye, 2017), the string transfers the Hex byte file to ASCII string from PE because some strings are not clean enough, so taking the string length better than String and counted as features for malware classification. In terms of the feature's extraction from assembly files, there are several studies used various features such: (1) metadata, (2) symbol, (3) operation code, (4) registers, (5) application program interface, (6) data define instruction, (7) section, and (8) miscellaneous.

Metadata is identical to metadata in hexadecimal file depend on (Mansour Ahmadi, 2016) the technique done by computing and extracting the information about the file as assembly file features, and the size of the binary code file as features but after being disassembled, the two sizes will be different but both of them are recorded. It used symbol to calculate the frequency of the symbols and check the high frequent that used to prevent and avoid malware detection. Also, the operation code it is a representation of machine code, also called instruction syllable and it is a part of machine language instruction that identify the operation to be executed and symbolize assembly instruction which uses 93 common operation code that utilized as features. In addition, there is another approach which assigns a malware sample to a specific family by the frequencies of registers in processors; the registers are small storage in processors with high speed. Application programming interface, many studies focused on detecting malware and their types based on the nature of API calls (Dong-Jie Wu1, 2012) (Zhu, 2013). Also, the study in (Mansour Ahmadi, 2016) identified 794 most frequent APIs in malware that obtained from 500K malicious samples, the type, number and frequency of the different APIs can be also used to uniquely classify the malware. This can vary by the platform (e.g. Windows, Apple, Linux, Android, etc.)

Data define instruction is another important feature, there is no API calls in some malware sample, it contains an operation code like 'db' (defining byte), 'dw' (defining word) and 'dd' (defining double word) that used for packing of malware detection. For each line in the assembly file starts with dot and the assembly name like '.text', '.data', '. bss', '. rdata', '. rsrc, the section feature is used. The executable file has some sections. The study (Mansour Ahmadi, 2016) pointed out that the modification of some sections and generation can be applied on the unknown section names, then counting the common sections and calculating the recorded property of the unknown sections. Miscellaneous is also an important feature that has been used in the literature. It is an assembly code that taken as a keyword and extract the frequency of 95 chosen keywords for feature category, it some part of them indicates the number of blocks or the number of loaded DLL header as in (Mansour Ahmadi, 2016).

Also, Metadata it extracts and summarize the basic information of the file, such as file size, address, date modified or date created, (Mansour Ahmadi, 2016) describes that the PE header gives us a meta data information for the executable file. The study (Kun Wang, 2016) proposed a malware detection method depends on metadata of App that extracted from APK file and build vector space for datasets. In addition, the approach in (Mansour Ahmadi, 2016) explained that the entropy technique based on describing the measurement of disorder, it is a numerical measure, the study (Jared Lee, 2015) used the entropy information to interact with metamorphic detection problem, computing Entropy become on the byte level representation of the Malware sample, to measure the disorder in the byte distribution in byte code.

Moreover, image representation (Mansour Ahmadi, 2016) showed that the malware sample can be visualized the byte code and explain each byte as gray-level of pixel in the image, this technique used in visual signature By patterns similarity, (Kyoung Han, 2014) proposed a method for classification using image representation by converting binary files into images and entropy graph. As in the studies (Mansour Ahmadi, 2016) (Yanfang Ye, 2017), the string transfer the Hex byte file to ASCII string from PE because some strings are not clean enough, so taking the string length better than the string and counted as features for malware classification.

In terms of the features extraction from assembly files, there are several studies used various features such: (1) metadata, (2) symbol, (3) operation code, (4) registers, (5) application program interface, (6) data define instruction, (7) section, and (8) miscellaneous.

Metadata is identical to metadata in hexadecimal file depend on (Mansour Ahmadi, 2016) the technique done by computing and extracting the information about the file as assembly file features, and the size of the binary code file as features but after being disassembled , the two sizes will be different but both of them are recorded.

The study (Mansour Ahmadi, 2016) used symbol to calculate the frequency of the symbols and check the high frequent that used to prevent and avoid malware detection. Also, the operation code it is a representation of machine code, also called instruction syllable and it is a part of machine language instruction that identify the operation to be executed and symbolize assembly instruction which uses 93 common operation code that utilized as features.

In addition, there is another approach which assigns a malware sample to a specific family by the frequencies of registers in processors; the registers are small storage in processors with high speed. Application programming interface, many studies focused on detecting malware and their types based on the nature of API calls (Dong-Jie Wu1, 2012) (Zhu, 2013). Also, the study in (Mansour Ahmadi, 2016) identified 794 most frequent APIs in malware that obtained from 500K malicious samples, the type, number and frequency of the different APIs can be also used to uniquely classify the malware. This can vary by the platform (e.g. Windows, Apple, Linux, Android, etc.). Data define instruction is another important features, there is no API calls in some malware sample, it contains an operation code like ‘db’ (defining byte), ‘dw’ (defining word) and ‘dd’ (defining double word) that used for packing of malware detection.

For each line in the assembly file starts with dot and the assembly name like ‘.text’, ‘.data’, ‘.bss’, ‘.rdata’, ‘.rsrc, the section feature is used The executable file have some sections,. The study (Mansour Ahmadi, 2016) pointed out that the modification of some sections and generation can be applied on the unknown section names, then counting the common sections and calculating the recorded property of the unknown sections. Furthermore, miscellaneous is also an important feature that has been used in the literature (Mansour Ahmadi, 2016). It is an assembly code that taken as a keyword and extract the frequency of 95 chosen keywords for feature category, it some part of them indicates the number of blocks or the number of loaded DLL header.

A. Detection based on API calls

Many research papers focused on evaluating the ability to detect malwares and their types based on the nature of API calls used by the malware such as: (Wu et al., 2012), (Ahmadi et al., 2016), (Peiravian and Zhu, 2013), (Aafer et al., 2013), and (Sami et al.,2010).

(Ahmadi et al., 2016) identified 794 most frequent APIs in malwares are taken into account, which are obtained from approximate 500K malicious samples based on Microsoft Kaggle challenge in 2016. The type, number and frequency of the different APIs can be also used to uniquely classify the malware. This can vary by the platform (e.g. Windows, Apple, Linux, Android, etc.). Authors classified extracted features into: Hex dump-based features and features extracted from disassembled files. Figure 4 shows a sample of their results and some of the important features they found to detect malwares. Figure 5 shows an example of DLL imports for a popular Trojan.

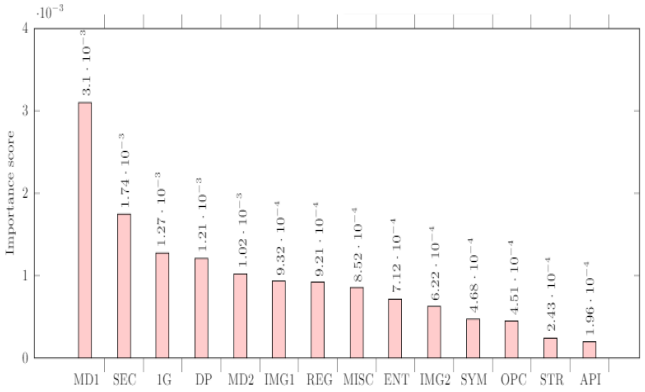

Fig. 2. Importance of features based on mean-decrease impurity

Fig. 3. DLL imports for a popular Trojan

In some cases, the frequency of usage of certain APIs can be an indicator of a malware behavior. Table 1 below shows (in a small dataset of 50 malwares and 150 Benign files) the top APIs used by malwares in comparison of Benign files.

TABLE I. Top APIs used by malwares in comparison of Benign files

API Name	Freq_W_%	Freq Ben%
exitprocess	0.96	0.49
closehandle	0.92	0.76
getmodulefilenamea	0.92	0.43
getmodulehandlea	0.88	0.78
writefile	0.86	0.67
getprocaddress	0.84	0.72
getstartupinfoa	0.82	0.53
createfilea	0.80	0.39
regclosekey	0.76	0.67
rtlunwind	0.76	0.45
virtualalloc	0.76	0.43
getcommandlinea	0.76	0.37

B. Entropy level Detection and Structural Features

Entropy is extracted from *.byte files and calculated at byte level. The sliding window method is adopted with the size of each window as 10,000 bytes According to the Shannon’s formula.

IV. Given a new malware, how can we classify this malware?

There are many learning techniques that applied to classify the malware in the literature (Mansour Ahmadi, 2016) & (George E. Dahl, 2013) such: (1) K-nearest neighbors, (2) Support vector machine, (3) Boosted Trees– XGBoost, (4) Neural Network, and (5) Naive Bayes. First, K-nearest neighbors is one of the most important machine learning technique that used for classification and regression, the basic idea is to predict a point by the neighbors set by measuring the distances, different distance measurements method used for finding the closest neighbors and the most used method is the Euclidean Distance and it is good for the features from the same type, for different type it is better to use Manhattan distance. Second, support vector machine is one of the supervised learning methods for classification, the main idea is to create a hyperplane that separate the classes in some way and to find the hyperplane with the maximum margin, the distance between the support vector and the hyperplane is referred to as margins. Third, Extreme Gradient Boosting (Tianqi Chen, 2016) is based on the model of Gradient Boosting, the XGBoots is a tree ensemble which incorporates a batch of classification and regression trees (CART), used to achieve state of the art results on many machine learning challenges.

The study (Tianqi Chen, 2016) used XGBoosting as a classification method and the analysis of the training set (20,000 samples) and then test the performance against testing set (2,000 unseen samples) shows that XGBoost Model lead to a better accuracy compared to the other models in the research. Fourth, Multi-Layer Perceptron (MLP), it is another supervised learning algorithm based on training dataset by a function and input layer and output layer and a several non-linear hidden layers, the first layer is the input layer made up of a set of neuron which represent the set of features, each neuron from the previous layer in an output value and each value contains two parts, a weighted linear summation and a non-linear activation function. The approach (Joshua Saxe, 2015) introduces a deep neural network based on malware detection system by using an experimental dataset on 400,000 software binaries with a detection rate of 95% and a false positive rate of 0.1%. Fifth, Naive Bayes is probabilistic classifier algorithm based on Bayes theorem, this method based on treating and evaluate the probability of each feature independently and make the prediction based on Bayes theorem, Naive Bayes classifier (Nikola Milosevic, 2017) used to detect malicious android application but the Naïve Bayes have the worst performance in that study.

V. If a malware has common features from different known malware categories, to which category such malware is usually allocated?

Automatic malware categorization plays an essential role in identifying the big size of malware. Different malware detection uses common features in different categories. Some of the malicious files belong to the same family with the same malicious behavior and features, there are some techniques to deal with these files and detect the malicious correctly. In (Jiang Y. Z.-C.-Q.-Y.-Z.-J., 2017) the study proposed a method based on a mixture model clustering ensemble to make an effective malware clustering analysis and system by combining some different features and clustering algorithms.

Another study (Jiang Y. Y.-T.-Y.-Q., 2010) proposed a principled cluster ensemble framework by automatic malware categorization system to group the malware samples into families with a common characteristic, by combining individual clustering solutions. Also, the study (Xin Hu, 2013) improves the approach that used previously in (Jiang Y. Y.-T.-Y.-Q., 2010) by making a combination between static and dynamic features and integrates the results with a similarity metrics. Furthermore, the study (Blake Anderson, 2012) proposed a new technique using kernels for the similarity metric on each special view and multiple kernels learning to get the best classification accuracy by the support vector machine, it used all the information about available execution to perform classification not just by a single data source.

VI. How do malware scanners decide a malware category?

Malware scanners employ one of the following 3 major categories of detection methods: (1) Specific Detection (e.g. signature or hash-based detection): This works by looking for known malwares by a specific set of attributes. Known malwares have known signatures recorded in a database that can be applied on subject or tested files, (2) Generic Detection: This process looks for malwares by behaviors of known categories or that are variants of known categories. For example, statistical-based techniques apply statistical models on system activities such as network connections, bandwidth, memory usage, system calls, etc. which can be usually used by malwares. Apparently, false positive cases are common in such scenarios where many “good” applications or system behaviors can be mistaken as malicious activities and (3) Heuristic, anomaly or behavioral detection: This process scans for previously unknown malwares by looking for known abnormal or suspicious behaviors. Anomaly-based detection depends on monitoring system activities and classifying the subject as either normal or anomalous accordingly.

Detection can be also classified into host-based and network-based detection based on the sources of artifacts used in the analysis and detection processes. Hybrid (host- and network-based) is also possible especially with large and complex malwares.

Dynamic detection methods run the suspect file in an isolated environment. Three types of isolated environments are used: Sandboxes where special APIs connect suspect file to the Virtual Machine (VM). Alternatively, emulation or Virtual Machine Monitors (VMMs) are used to run the suspect file. From a data science perspective, malware detection process is a typical classification process of two stages:

1. In the first stage, subject file, traffic, hash, etc. will be classified as either a malware or benign. Many malware scanners leave a third category: undecided if the subject fails to be allocated to either one of the malwares/benign categories. For example,

many of the newly discovered malwares may fall under this (unknown/undecided) category.

2. In the second stage, if the subject is classified as a malware, different categories of malwares are available and the process is to allocate this subject to one, or more of those categories.

The analysis of the malicious program is important to extract features, which describes the risk and the malware type; there are three types of detection and analysis method to identify the malware categories: (1) static analysis detection technique, (2) dynamic analysis detection, and (3) hybrid analysis detection technique.

First, Static analysis detection technique (Imtithal A.saeed, 2013) (Smita Ranveer, 2015) (Ekta Gandotra, 2014) (Dolly Uppal, 2014) (Balaji Baskaran, 2016) (Saba Arshad, 2017) (P. V. Shijoa, 2015), these studies described the static analysis that is analyzing software malicious and extract the features in the binary code or internal structure of the file without executing, the application is break down by some tools and techniques to rebuild the source code and algorithm of the application that created, it is done during program analyzer and debugger, this type is safe and fast. There are different static analysis techniques: Specific detection (e.g. signature or hash-based detection), and heuristic behavioral detection.

Specific detection works by looking for known malware by a specific set of attributes (Imtithal A.saeed, 2013) (Smita Ranveer, 2015) (Dolly Uppal, 2014) (Saba Arshad, 2017),these approaches indicated that known malware have known signatures recorded in a database that can be applied on subject or tested files, but it cannot detect an unknown malware. While Heuristic behavioral detection: It called proactive technique, it is similar to signature based but it does not use searching for signature in code, it search for the instruction that is not appear in the application program (Imtithal A.saeed, 2013) (Dolly Uppal, 2014) they proposed that this process scans for previously unknown malware by looking for known abnormal or suspicious behaviors. Anomaly-based detection depends on monitoring system activities and classifying the subject as either normal or anomalous accordingly.

Heuristic detection technique (Dolly Uppal, 2014) have different types such: (1) File based heuristic analysis file based or file analysis Heuristic system, it analyzes the file completely and check if there is any command in the file can delete or harm other files, it will be considered as malicious. (2) Weight based heuristic analysis, this is the oldest technique, each application have a danger weight or value, and there is a threshold value, if the weight override the threshold value the application will contain a malicious code, (3) Rule based heuristic analysis at this type, the analyzer extract the rules of the application, and match it with the previously defined rules. Such if there is any mismatching then the application contains malware, and (4) Generic signature analysis, this process looks for malware by behaviors of known categories or that are variants of known categories. different behavior for the malware but belongs to the Same category used to discover new variant of malware, for example, statistical-based techniques apply statistical models on system activities such as network connections, bandwidth, memory usage, system calls, etc. which can be usually used by malware. Apparently, false positive cases are common in such scenarios where many “good” applications or system behaviors can be mistaken as malicious activities.

Second, Dynamic analysis detection technique (Ekta Gandotra, 2014) (Dolly Uppal, 2014) (KIMBERLY TAM, 2017) (Saba Arshad, 2017) discuss that the dynamic analysis observes the result after executing the program by Analyzing the behavior or the Action of the application but it takes time as the executing time of applications. It interacts with the system while execution in VMware, Simulators and sandbox to find if the executable file is malware or not.

Third, Hybrid analysis detection technique (Ekta Gandotra, 2014) (Dolly Uppal, 2014) (Balaji Baskaran, 2016) (KIMBERLY TAM, 2017) they proposed that the hybrid analysis is combination of static and dynamic techniques by checking if there is any malware signature in the code. Then observe the code behavior. In addition, detection can be also classified (Kyoung Han, 2014) into three categories: (1) Host based intrusion detection system, (2) network-based intrusion detection system, and (3) Hybrid intrusion Detection system.

Host based intrusion observes and controls the dynamic behavior and the computer system state to check if there is any internal or external Activities that cheats the system policy. Network-based intrusion detection system analyzes all the packets in the network node. Host-based and network-based detection based on the sources of artifacts used in the analysis and detection processes. Hybrid intrusion Detection system: A combination of host- and network-based intrusion detection system is also possible especially with large and complex malware.

Dynamic detection methods run the suspect file in an isolated environment. The research study (Imtithal A.saeed, 2013) proposed isolated environment, sandboxes, where special APIs connect suspect file to the Virtual Machine (VM).From a data science perspective, malware detection process is a typical classification process of two stages: In the first stage, subject file, traffic, hash, etc. will be classified as either a malware or benign. Many malware scanners leave a third category: undecided if the subject fails to be allocated to either one of the malware/benign categories. For example, many of the newly discovered malware may fall under this (unknown/undecided) category. In the second stage, if the subject is classified as a malware, different categories of malware are available and the process is to allocate this subject to one, or more of those categories.

VII. GIVEN A NEW MALWARE, HOW CAN WE CLASSIFY THIS MALWARE?

There are many learning techniques that applied to classify the malware in the literature (Mansour Ahmadi, 2016) & (George E. Dahl, 2013) such: (1) K-nearest neighbors, (2) Support vector machine, (3) Boosted Trees–XGBoost, (4) Neural Network, and (5) Naive Bayes.

First, K-nearest neighbors is one of the most important machine learning technique that used for classification and regression, the basic idea is to predict a point by the neighbors set by measuring the distances, different distance measurements method used for finding the closest neighbors and the most used method is the Euclidean Distance and it is

good for the features from the same type, for different type it is better to use Manhattan distance.

Second, support vector machine is one of the supervised learning method for classification, the main idea is to create a hyperplane that separate the classes in some way and to find the hyperplane with the maximum margin, the distance between the support vector and the hyperplane is referred to as margins.

Third, Extreme Gradient Boosting (Zhu, 2013) is based on the model of Gradient Boosting, the XGBoots is a tree ensembles which incorporates a batch of classification and regression trees (CART), used to achieve state of the art results on many machine learning challenges . The study (Tianqi Chen, 2016) used XGBoosting as a classification method and the analysis of the training set (20,000 samples) and then test the performance against testing set (2,000 unseen samples) shows that XGBoost Model lead to a better accuracy compared to the other models in the research.

Fourth, Multi-Layer Perceptron (MLP), it is another supervised learning algorithm based on training dataset by a function and input layer and output layer and a several non-linear hidden layers, the first layer is the input layer made up of a set of neuron which represent the set of features, each neuron from the previous layer in an output value and each value contains two parts, a weighted linear summation and a non-linear activation function. The approach (Joshua Saxe, 2015) introduced a deep neural network based on malware detection system by using an experimental dataset on 400,000 software binaries with a detection rate of 95% and a false positive rate of 0.1%. Fifth, Naive Bayes is probabilistic classifier algorithm based on Bayes theorem, this method based on treating and evaluate the probability of each feature independently and make the prediction based on Bayes theorem, Naive Bayes classifier (Nikola Milosevic, 2017) used to detect malicious android application but the Naïve Bayes have the worst performance in that study.

VIII. Dealing with malware detection differences or discrepancies

If malware has common features from different known malware categories, to which category such malware is usually allocated?

Automatic malware categorization plays an essential role in identifying the big size of malware. Different malware detection uses common features in different categories. Some of the malicious files belong to the same family with the same malicious behavior and features, there are some techniques to deal with these files and detect the malicious correctly. In (Jiang Y. Z.-C.-Q.-Y.-Z.-J., 2017), the study proposed a method based on a mixture model clustering ensemble to make an effective malware clustering analysis and system by combining some different features and clustering algorithms.

Another study (Jiang Y. Y.-T.-Y.-Q., 2010) proposed a principled cluster ensemble framework by automatic malware categorization system to group the malware samples into families with a common characteristic, by combining individual clustering solutions. Also, the study (Xin Hu, 2013) improves the approach that used previously in (Jiang Y. Y.-T.-Y.-Q., 2010) by making a combination between static and dynamic features and integrates the results with a similarity metrics. Furthermore, the study (Blake Anderson, 2012) proposed a new technique using kernels for the similarity metric on each special view and multiple kernels learning to get the best classification accuracy by the support vector machine, it used all the information about available execution to perform classification not just by a single data source.

Different approaches and features have used by many researchers in malware detection area to detect malwares. In (Idika et al., 2007) malware detection techniques categorized based on two general main categories: Signature based detection and anomaly-based detection. Signature based detection technique utilizes the known malicious software characteristics in deciding the malicious of the software under inspection. Anomaly based detection uses the software awareness that creates ordinary behavior to detect the maliciousness of the software under examination. Specification based technique is a special type of anomaly-based detection where some rule set and specification of legitimate behavior are influenced in order to detect the maliciousness of the software under inspection.

There are three different approaches are employed in every detection technique namely: dynamic, static and hybrid. Each approach is determined through specifying the information gathering technique in order to detect malicious software. Static approach usually is used to detect the malicious software before execution, whereas the dynamic approach is used to detect the malicious software during software execution (Idika et al., 2007).

Signature-based methods are the utmost popular methods in malware detection (Gutmann, 2007). Signature is like a pattern of an executable file and represents a distinctive feature for every file. Signature based method uses the extracted features from many malwares to classify them and is considered faster and efficient than other approaches. Signatures are extracted primarily with distinctive sensitivity for remaining sole, therefor signature based methods have lesser error rate (Gutmann, 2007). On the other hand, Signature based methods have not the ability to detect unidentified malware variations and need great effort of time and money to extract distinctive signatures. Moreover, incapability to meet malware that mutates its code in every infection like polymorphic and metamorphic is considered extra disadvantage (Gutmann, 2007).

Behavior based methods monitor program behavior to determine whether the software is malicious or not (KALPA, 2011). Behavior based mechanisms have the ability to detect malwares that keep on creating new mutants because always they will use the system services and resources in the same way (Jacob et al., 2008). The main disadvantage of the behavior-based malware detection methods is huge volume of scanning time. The behavior-based detector consists from three main components: Data collector, Interpreter, and matcher. The major improvement of the behavior-based malware detection technique is the capability to determine the type of malware where the variant of malware is polymorphic or unknown (Ahmed et al., 2012).

IX. Conclusion

There are several factors that make the process of malware detection and classification complex. Malwares evolve rapidly and hackers continuously employee new methods to avoid detection or manipulate the analysis and classification activities. The malware categories themselves are not clearly identified in such form that make the process of detection and distinction between the different categories straightforward. In this scope, we conducted this survey paper to raise and answer questions related to malware detection and classification issues. In addition to previous challenges, malware detection should be quick and close to real time detection and eradication. It should also be accurate to minimize false positive and negative cases.

References

Akour, M., Alsmadi, I., & Alazab, M. (2017). The malware detection challenge of accuracy. In 2016 2nd International Conference on Open Source Software Computing, OSSCOM 2016 [07863750] Beirut, Lebanon: IEEE, Institute of Electrical and Electronics Engineers. DOI: 10.1109/OSSCOM.2016.7863676.

Kyoung. Soo Han (2014). Malware analysis using visualized images and entropy graphs. International Journal of Information Security, 14(1), 1-14.

Balaji Baskaran, A. R. (2016). A Study of Android Malware Detection Techniques and Machine Learning. MAICS, 15-23.

Blake Anderson, C. S. (2012). Improving Malware Classification: Bridging the. Proceedings of the 5th ACM workshop on Security and artificial intelligence - AISec 12.

Dolly Uppal, V. (2014). Basic survey on Malware Analysis, Tools and Techniques. International Journal on Computational Science & Applications, 4(1), 103-112.

Dong-Jie Wu1, C.-H. M.-E.-M.-P. (2012). DroidMat: Android Malware Detection through Manifest and API Calls Tracing. Seventh Asia Joint Conference on Information Security.

Ekta Gandotra, D. B. (2014). Malware Analysis and Classification: A Survey. Journal of Information Security, 5(2), 56-64.

George E. Dahl, J. W. (2013). LARGE-SCALE MALWARE CLASSIFICATION USING RANDOM PROJECTIONS AND NEURAL NETWORKS. IEEE International Conference on Acoustics, Speech and Signal Processing.

Hossein Fereidooni, M. C. (2016). ANASTASIA: ANdroid mAlware detection using STatic analySIs of Applications. 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS).

Imtithal A. saeed, A. S. (2013). A Survey on Malware and Malware Detection Systems. International Journal of Computer Applications, 67(16), 25-31.

Jared Lee, T. H. (2015). Compression-based analysis of metamorphic malware. Int. J. Security and Networks, 10(2), 124-136.

Jiang, Y. Y.-T.-Y.-Q. (2010). Automatic Malware Categorization Using Cluster Ensemble. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 10.

Jiang, Y. Z.-C.-Q.-Y.-Z.-J. (2017). Based on Multi-features and Clustering Ensemble Method for Automatic Malware Categorization. IEEE Trustcom/BigDataSE/ICESS.

Joshua Saxe, K. B. (2015). Deep Neural Network Based Malware Detection Using Two-Dimensional Binary. 10th International Conference on Malicious and Unwanted Software (MALWARE).

KIMBERLY TAM, A. F. (2017). The Evolution of Android Malware and Android Analysis Techniques. ACM Computing Surveys, 49(4), 1-41.

Kun Wang, T. S. (2016). Mmda: Metadata based Malware Detection on Android. International Conference on Computational Intelligence and Security.

Mansour Ahmadi, D. U. (2016). Novel Feature Extraction, Selection and Fusion for Effective Malware Family Classification. Proceedings of the Sixth ACM on Conference on Data and Application Security and Privacy - CODASPY 16.

Meena, S. J. (2011). Byte Level n–Gram Analysis for Malware. Communications in Computer and Information Science Computer Networks and Intelligent Computing, 51-59.

Nikola Milosevic aMilosevic, A. D.-K. (2017). Machine learning aided Android malware classification. Computers & Electrical Engineering, 61, 266-274.

P. V. Shijoa, A. S. (2015). Integrated static and dynamic analysis for malware detection. Procedia Computer Science, 46, 804 – 811.

Saba Arshad, A. K. (2017). Android Malware Detection & Protection: A Survey. International Journal of Advanced Computer Science and Applications, 7(2), 2013-2017.

Smita Ranveer, S. H. (2015). Comparative Analysis of Feature Extraction Methods of Malware Detection. International Journal of Computer Applications, 120(5), 1-7.

Tianqi Chen, C. G. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16.

Xin Hu, K. G. (2013). DUET: Integration of Dynamic and Static Analyses for Malware Clustering with Cluster

Ensembles. Proceedings of the 29th Annual Computer Security Applications Conference on - ACSAC 13.

Yanfang Ye, T. L. (2017). A Survey on Malware Detection Using Data Mining Techniques. ACM Computing Surveys, 50(3).

Zhang Fuyong, Z. (2017). Malware Detection and Classification Based on ngrams Attribute Similarity. IEEE International Conference on Computational Science and Engineering.

Zhu, N. P. (2013). Machine Learning for Android Malware Detection Using Permission and API Calls. IEEE 25th International Conference on Tools with Artificial Intelligence.

Imtithal A. Saeed, Ali Selamat, and Ali Abuagoub, A Survey on Malware and Malware Detection Systems, International Journal of Computer Applications (0975 – 8887), Volume 67– No.16, April 2013.

P. Gutmann. "The Commercial Malware Industry.", In DEFCON conference, 2007.

Idika, Nwokedi & Mathur, Aditya. (2007). A survey of malware detection techniques. Purdue University.

KALPA, "Introduction to Malware", "http://securityresearch.in/ index.php/projects/malware_lab/introduction-to-malware/8/", 2011.

G. Jacob, H. Debar, and E. Filiol, "Behavioral detection of malware: from a survey towards an established taxonomy," Journal in Computer Virology, pp. 251–266, 2008.

A. Ahmed, E. Elhadi, M. A. Maarof and A. H. Osman, "Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph Information Assurance and Security Research Group." Journal, A., Sciences, A., & Publications, S., Faculty of Computer Science and Information Systems, 9(3), 283–288,2012