IJAMEC
e-ISSN: 2147-8228

INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS ELECTRONICS AND COMPUTERS

www.dergipark.org.tr/ijamec

International Open Access

Volume 08
Issue 01

March, 2020

Review Article

Malware Visualization Techniques

Ahmet Efe [a,*], ***Saleh Hussin S. Hussin*** [b]

[a]*Ankara Development Agency, Internal Auditing, Ankara, Turkey*
[b]*Yıldırım Beyazıt University, Computer Engineering, Ankara, Turkey*

ARTICLE INFO

Article history:
Received 13 February 2019
Accepted 13 February 2020

Keywords:
Extracted Features
Malware Classification
Malware Detection Technique
Malware Survey
Visualization Techniques

ABSTRACT

Malware basically means malicious software that can be an intrusive program code or anything that is designed to perform malicious operations on system and executes malicious actions such as clandestine, listening, monitoring, saving, and deleting without the user's knowledge and consent. Malware review and analysis requires an advanced level of programming knowledge, in-depth file systems knowledge, deep code inspection, and reverse engineering capability. New techniques are needed to reduce indirect costs of malware analysis. This paper aims to provide insights into the malware visualization techniques and its applications, most common malware types and the extracted features that used to identify the malware are demonstrated in this study. In this work, Systematic Literature Review (SLR) conducted to investigate the current state of knowledge about Malware detection techniques, data visualization and malware features. An advanced research has been carried out in most relevant digital libraries for potential published articles. 90 preliminary studies (PS) were determined on the basis of inclusion and exclusion criteria. The analytical study is based mainly on the PSs to achieve the goals. The results clarify the importance of visualization techniques and which are the most common malware as well as the most useful features. Several ways to visualize malware to help malware analysts have been suggested.

1. Introduction

The most important features that distinguish malware from other software are that they are secretly operating in the system they work within and are specifically programmed to cause harm. Malware or malicious software, especially in different variants, is transmitted to the systems with the threat vectors such as wannacry, petya, notpetya or Monero (XMR). One of the most important vulnerabilities used by these pests to spread to the systems is known as the EternalBlue[1] (CVE-2017-0144 / MS17-010) exploit, which is introduced by the publication of the NSA tools. In addition, some methods such as WMI interfaces are being used to stifle antivirus software, to make itself permanent as a service and later to spread to other systems. Therefore, malware can migrate to other systems on the network by collecting its credential information on the systems such as infection by mimicatz[2] and using EternalBlue exploit.

Recently, advanced malicious software called APT (Advanced Persistent Threat) which uses many different vectors has become the indispensable tool used by hackers. Such malware often serves commercial and military purposes, such as gathering intelligence and neutralizing target systems. Customers of malicious software cover a wide range from organized crime to state sponsored attacks against e-government sites.

Targeted malware bypasses or disables various security mechanisms to achieve the defined goal. Therefore, the technical information, codes and structures used in the malware are complex. An advanced review and analysis are required to find out what malware does and what kind of function it has. For this reason, malicious code analysis is a very difficult task. Moreover, the fact that inadequate number of experts working in this field makes malware

[1] For further details see: https://github.com/am0nsec/exploit/tree/master/windows/smb/MS17-010-EternalBlue
[2] For further details methodology and techniques see: https://www.hackers-arise.com/post/2018/11/26/metasploit-basics-part-21-post-exploitation-with-mimikatz

* Corresponding author. E-mail address: *icsiacag@gmail.com*
DOI: 10.18100/ijamec.526813

analysis more difficult. Malware review and analysis requires an advanced level of programming knowledge, in-depth file systems knowledge, deep code inspection, and reverse engineering capability. During the examination and analysis, firstly, if the symptoms of infection are detected, it is necessary to reach the infecting agent and examine it in depth. In this respect, the most important factor affecting a successful review and analysis is to collect and retain enough data to determine the ability of the malware.

During the data collection, all platforms are examined, structures related to the harmful code properties are detected, attack vectors are defined, and all kinds of file types, libraries and interactions are examined to determine the identity of the malware. Then, the data obtained must be tested and simulated. Therefore, having a good malware analysis laboratory is very important on the road to success.

While analyzing the file injection malware, structural analysis of the malware should be done first. By structural analysis, it is meant to consider the following data in the analysis:

- Textual expressions that the malware contains in the system before it is run,
- Accessed OS functions,
- File section entropies,
- Whether the content of the building is packaged and
- Hash values.

By examining whether there is an abnormality in the data obtained as a result of the structural analysis, the malware detection process can be contributed. Hackers often pack all the data when creating pests so that they are not detected by antivirus programs. While the malware is running in the victim system, the packaged structure is opened first, and if there is any complicated or encrypted data, it is resolved and the process continues. For this reason, it is important to analyze the malware by working step by step in order to understand the operation of the pest. After the structural analysis of the malicious code structure is performed, step-by-step the malware is run using the sandbox structure or virtual computers. Thus, file and registry activities created by the malware can be examined and network analysis can be made. Then, using the data obtained, behavioral and characteristic information about the pest is obtained.

One of the most important features of the laboratory that will perform malware analysis is that it is capable of performing dynamic and static analysis. For these analyses, in addition to the methods and techniques such as virtualization and configuration, the disassembler, file format analyzer, data carriers, sniffing and packet analyzer, debugger and reverse engineering techniques must be well known.

As mentioned above, malware is referred to as malicious software, malicious code (MC) and Malcode that crashes or destroys normal operations without the knowledge of user [1]. The most known of the malware with many varieties are listed below:

- Viruses (Viruses)
- Trojan horses
- Spyware
- Worms
- Rootkits
- Keyloggers
- Backdoors
- Advertising purposes (Adware)
- Ransomware
- Browser hijacker (Browser hijacker)

Malwares could be classified into various types, like viruses, worms, trojans, spyware, adware, Rootkits and others as listed above [2] [3]. Malware causes the most common incidents ranged from; gathering sensitive

Figure 1. Systematic Literature Review Methodology

information [4], performing malicious activities and gaining access [5], giving a malicious party remote access [6] to the financial loss [7].

Different methods have been deployed in order to detect, identify and classify the malware. According to [2] and [3] the malware detection techniques can be categorized to Signature-Based, behaviour-Based, Analysis-Based, anomaly-Based and visualization-Based methodologies that are being used in different types of products.

Malware visualization is a domain which concentrates on detecting, classifying, and representing malicious software features in the visual signals form which can be utilized to convey more data about specific malicious software [8]. Visualization techniques have been utilized to display static data, monitor network traffic or manage networks. Recently, visualization techniques have been utilized to discover and visualize the behavior of malware [8]. According to [9], there are many data visualization techniques, for instance, area, pie, bar, pizza, lines and dots graphics and volume slicing in 3D to present bi-dimensional images. Malware threat scenarios are rapidly changed in the recent years with the creation of new attacks techniques. In addition to the fact that the severity of malwares on the operations of systems is also increased, the malware detection techniques also have been increased both in quantity and methodology. Therefore, it is important to systematically review the existing malware visualization techniques in order to highlight the most used techniques and the most common and extracted features that are being used by the malware visualization techniques.

2. Systematic Literature Research Method

This Systematic Literature Review (SLR) performed by an electronic literature searching considering all yeas from 2009 through the 2018 in order to cover a wider range of publication years, and followed the approach of [10] for conducting SLR. This search process had four phases. Figure 1 graphically illustrates the phases involved in this SLR and the performed activities of each phase.

This work aimed to systematically answering the following research questions:

RQ1: What are the malware visualization techniques and applications?

RQ2: What are the types of malware and features that are mostly reported and investigated?

The first (RQ1) is motivated by the desire of exploring the malware detection techniques as well as to illustrate the visualization techniques and their applications. whereas, the second (RQ2) is motivated by the desire of exploring the most common type of malware as well as to explore the extracted features that used by the visualization techniques for malware detection, classification and identification.

For an advance search, the key words that covered the research topic are identified based on the key terms taken from the research questions, substitutional spellings and synonyms of the key terms, and research keywords that appeared in the existing literature review. Boolean AND; OR; can be utilized to link the key search terms and substitutional spellings and synonyms of the keywords. The search terms that were utilized to extract data from these digital libraries comprised the following key words: visualization techniques, malware detection technique, malware type, extracted features, detection technique, malicious code detection, malware classification, malware survey. These key words were researched on their own or in conjunction with each other's.

For gathering the most related primary studies (PSs) and to obtain a thorough list of papers in this area, more than two digital libraries was selected to heighten sensitivity [11], [12]. The main academic and scientific digital libraries (Online database) that employed for the systematic literature are IEEE Xplore, ScienceDirect, Scopus, ACM Digital Library, Springer, and Web of Science. These digital libraries were selected because they include peer revised journal papers, and conference proceedings, and we think that these digital libraries comprise an exemplary sample of the literature created in the subject matter as relevant to this research.

2.1. Primary Study Selection

The searching strategy generated a big count of the articles. Therefore, this stage is significant for identifying and evaluating of the first obtained list of PSs articles. In this stage, an inclusion and exclusion criteria are defined. Table 1 exhibits how many outcomes published on different digital libraries and how the outcomes were straitened and chosen in order to obtain the final comprehensive list of related articles.

The first stage was to search in all digital libraries on all articles that are relevant to Malware detection technique, Data Visualization technique, malware classification, malware type, extracted features and malware survey. Digital libraries tools were used to reduce the research outcomes by chosen published year (2009–2018), and type of document needed. The result is presented in row one where 1857 articles have been obtained. The second stage was to include articles that satisfied the following three criteria:

Titles should contain Malware OR/AND malicious software OR/AND the synonym. The result is presented in row two.

Abstracts should contain Malware OR/AND detection, classification or visualization. The result is presented in row three.

Keywords should contain Malware, visualization, security data visualization, malicious software, dynamic

analysis, static analysis, information system security or detection. The result is presented in row four.

The next stage was to exclude articles that are not an English text. The result is presented in row five. The final stage was to exclude articles that are not accessed in full text. The result is presented in row six.

All of these stages are implemented to filter the first obtained list. The filtering process is performed on the title, abstract and keywords. Then, the articles of final comprehensive list of PSs were then inspected by performing an thorough study and by analyzing their contents, and the papers which contained the data considered reasonable for citing in this study were chosen (chosen outcomes 90 papers).

Table 1. Results from digital libraries

Stages	digital libraries				Total
	IEEE	ACM	Springer	Science Direct	
Stage1	1092	463	187	115	1857
Stage2 (1)	821	108	107	109	1145
Stage2 (2)	194	88	92	89	463
Stage2 (3)	73	51	42	66	232
Stage3	62	49	39	59	209
Stage4	42	16	16	15	89

2.2. Information Extraction and Synthesized

Extracting and synthesized information is the final stage in the reviewing protocol, whereby the relevant information from each article that counted in the final comprehensive list of PSs is extracted and synthesized. For this purpose, a Literature Review Table (LRT) with a number of columns is designed which includes not limited to the author, year, title, detection technique, the extracted feature and the method of analyzing or visualizing the result (Appendix A table 1). The table is used and analyzed statistically to deliver the main objectives of this SLR.

3. Findings and discussion

Before reviewing the PSs, we have focused on the basic definitions of malware. Malware is stands for malicious software. However, many PSs considered any code or program running behind the scenes and without the knowledge of the owner (person or entity) is malicious software. According to [13] there are innumerable number of malwares distributed yearly ascending with malicious actions, for example, stealing users information, transmitting unusual messages and making telephone call to special phone numbers that users have no familiarity with and injury or damage various operating systems.

For answering the RQ1, we have reviewed all PSs with focusing on the most common analytics techniques that are applied to detect, classify and identify the malware. Also we have focused on the data visualization techniques used in recent literature and to illustrate the usefulness of each tool.

To answer the RQ2, we have divided it to two parts. At the first part we reviewed all PSs with focusing on the most common type of malware and families. At the second part, we have reviewed the articles information about the common and useful features that are used as a data sources for the visualization techniques.

3.1. Malware Detection Techniques

The existing literature revealed that six major categories of detection techniques are applied to detect, classify and identify the malwares. In this paper, categorization of the techniques that used for malware detection is based on the method of detection as shown in Figure 2. A brief discussion of the literature about detection techniques is provided next.

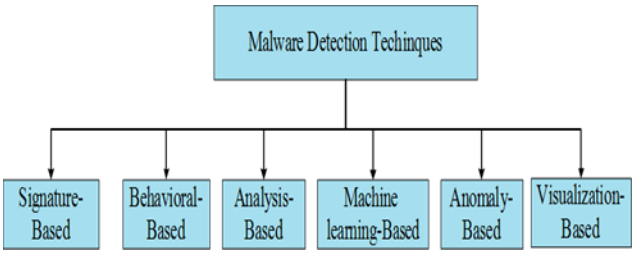

Figure 2. Classification of Malware Detection Techniques

a) Signature-based methods

Signature based techniques utilize the patterns that taken from different malwares to recognize them and are more functional and quicker than any other techniques. These signatures are predominantly taken with specific sensitivity for being special, so those detection techniques that utilize this signature have little error rate. Where this little error rate is the prime cause that most spread commercial antiviruses utilize this technique [2].

These techniques are incapable to discover anonymous malware and as well needs a lot of labor, money, and time to excerpt singular signatures. These are the major drawback of these techniques. Furthermore, insufficiency to defy against the malwares that modify their signs in every infection like metamorphic and polymorphic one is one more drawback. To handle this defiance, research organizations suggest totally new malware detection family.

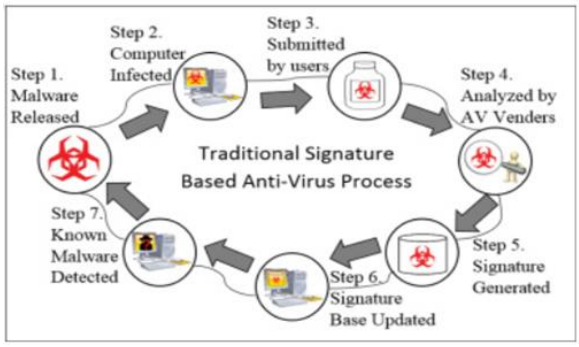

Figure 3. Process of traditional signature-based malware detection. [14]

b) Behavior-based methods

Behavior based methods monitor a program action to infer if it is a malicious or not [2]. For the reason that behavior based methods watch what an executable file make, they are not liable to the weaknesses of signature-based ones. Obviously, a behavior based detector deduces if a program is malicious or not via checking what it makes instead of what it tell. In these techniques, files with the similar conduct are grouped. Consequently, a single behavior signature can pick out diverse samples of malware. Those kinds of techniques assist in discovering malware that continue creating new mutants due to the fact they will constantly utilize the system resources and services in the identical way. The components of a behavior-based detector essentially includes the following: [15]:

Data Collector: To gather dynamic /static data concerning the executable file.

Interpreter: To transforms crude data gathered by data collection component into transitional representation.

Matcher: This component is utilized to contrast this transitional representation with the signatures of the behavior.

The histogram based malicious code detection technology patented by Symantec is one instance of a behavior based identification approach [2].

The major vantage of the behavior based methods is its capability to discover the kind of malwares that are unable to be detected by signature base methods like anonymous and polymorphic malwares. In contrast, the major drawbacks of the behavior based malware detection techniques include the non-availability of promising False Positive Ratio (FPR) and furthermore high amount of scanning time [16].

c) Anomaly-based Detection

This detection technique commonly takes place in two stages–a training (learning) stage and a detection (monitoring) stage. The detector tries to grasp the ordinary behaviour in the training stage. The detector can learn the behaviour of the host or the PUI or a conjunction of both through this stage. A key feature of anomaly-based detection techniques is its capability to discover zero-day attacks, which are the attacks that are formerly anonymous to the malware detector. Its high false alarm average and the complexity in defining the features that should be learned in the training stage are the two essential disadvantages of this techniques [17].

d) Analysis-based Detection

Analysis based malware detection techniques relies on automated malware analyses tools and techniques to differentiate malicious from benign code [18]. There are three types of malware analysis:

Static analysis: comprises analyzing the program without executing it. [19] performed static analysis via excerpting opcode sequences with the assist of a disassembler. The key feature of static analysis is that it more efficient as it is lower costly in terms of exhausting the system resources, however it fails at polymorphic and metamorphic malwares [20]

Dynamic Analysis: checks the program behavior during execution to recognize if or not the executable program is a malware. [16]. The vantage of dynamic analysis is that it precisely analyzes the familiar and anonymous new malware; however this analysis method is extra time exhausting.

Hybrid Analysis: This technique is primarily test the signature specification of any malware code and then adds it with the other behavioral parameters for increase of whole malware analysis. hybrid analysis technique beats the drawbacks of both static and dynamic analysis

e) Machine learning-based Detection

Machine learning-based methods utilize the machine learning classifiers, which learn the attributes of each class automatically, by learning from instance information. To utilize this method to classify executable files as benign or malicious, first construct categorized datasets for training. A diversity of features have been probably efficient in the classification of malware, These features contain API call sequences, n-grams over machine code instructions, and Windows Portable Executable (PE32) header data [21].

Malware classification systems contain two distinguished divisions based totally on the characteristics set. Division one inspects an executable file without running it on the system, and uses static characteristics. Second division monitors the actions of the suspected program whilst permitting their execution in a sandbox surroundings. Network, file, registry, and process actions are tracked and reported. Dynamic characteristics are the outcome of an integrated scenario supplied to a malware sample to be deployed and to execute its malicious tasks. next to extracting the foundation characteristic set, machine learning or data mining tools can be utilized for classification objective[22].

Machine learning based detection approach still developmental in malware detection and have accomplished rising achievement in lab tests, however the

situation utilized do not mirror actual implementations. In general for machine learning, it has been vastly spotted that class imbalance, in which one class extremely predominate the other in one's database, overwhelmingly decreases the accomplishment of the classifier[21].

f) Visualization-based Detection

Visualization technique is developed to accelerate the analysis progress [23]. [24] display that the utilize of visualization technique accelerate the malware detection operation significantly. However, [25], [8], and [26] stated that, visualization methods are utilized to discover and visualize the behavior of the malware so recently. It concentrates on exemplifying malware features in a shape of graphic that could be utilized to carry additional information on a specific malware.

Visualization based methods utilize the static or dynamic (or both) analysis techniques for the purpose of collecting information on a possibly malicious segment of program. Visualization tools utilize these information as main input that creates the goodness of the supplied data paramount to maintaining semantic meaningfulness[27]. Figure 4 shows the general workflow of malware detection using visual analytics methods.

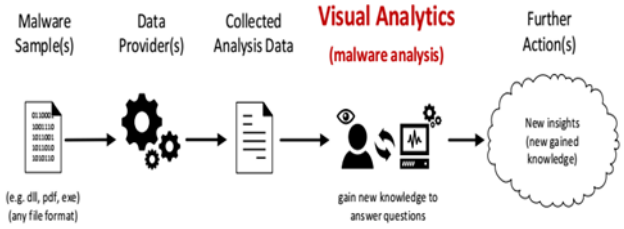

Figure 4. The general workflow of malware detection using visual analytics methods [27]

Visualizations based methods will assist security stuff people with minimal practice to get familiarity of implicit details of a particular portable executable or a binary file.

To illustrate the percentage of usage of each detection technique, a column bars has been drown as shown in **Hata! Başvuru kaynağı bulunamadı.**.

Besides the indication that extracted from the above figure about the usability and effectiveness of visualization techniques. The authors in [28] stated that signature based and behavioral methods that shown in the second and third column bar respectively are not capable to identify the malwares that protected , thus a new technique which can effectively discover this malwares is definitely needed, thus the visualization techniques still the best solution.

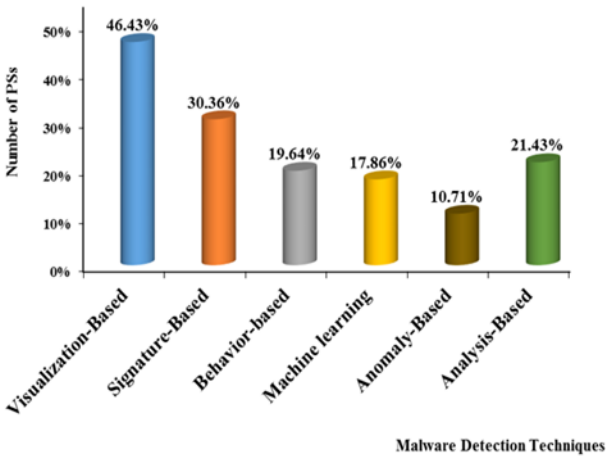

Figure 5. the percentage of usage of each detection technique

3.2. Data Visualization Techniques

Visualization methods can be applied to security events which are a useful technique for characterizing suspected actions and reactions to it simultaneously. Utilizing such technique is aimed to assist analysts to speedily consider and classify the kind of the malware [9]. There exist diverse methods utilized for data visualization , like bar, pie, area, pizza, lines and dots graphics, and volume slicing in 3D to symbolize bi-dimensional images can be utilized to visualize the actions of the malware.

It is important to illustrate the common visualization tools and the usefulness of each one. The most common visualization tools that have been in use for visualizing security events and to serve different security purposes are illustrated below and the usefulness of the tool is presented with simple example(s) on each visualization tool.

Treemap: this technique is utilized to convey the behaviour record into a standardized style. Treemap shows data as a group of nested rectangles [29]. Figure 6 shows an example of treemap of a malware labeled as Adultbrowser.

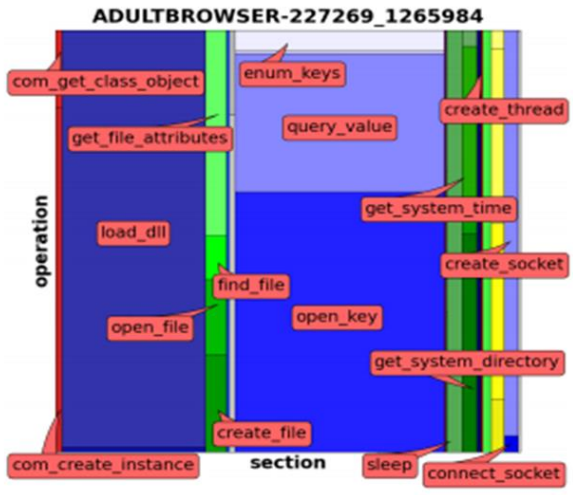

Figure 6. Treemap of Adult browser malware [29].

Thread Graph: the technique of thread graph is utilized for the visualization of the running chronological

behaviour of the malware piece. It could produce a diagram explaining the temporal arrangement of executed system commands and the various threads spawned by a binary. The x-axis symbolizes the time (sequence of performed actions), while the y-axis symbolizes the operation/section of the complete action[29]. Figure 7 shows the Adultbrowser malware sample visualized using the thread graph representation.

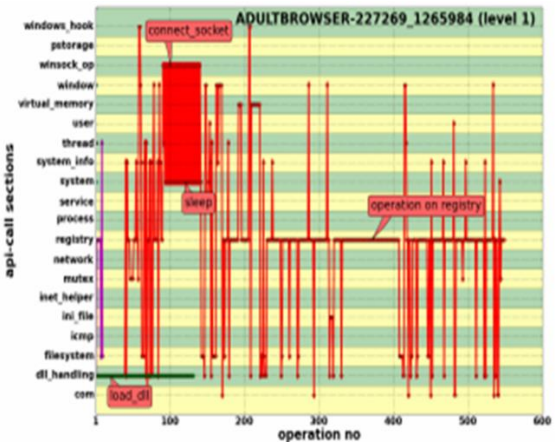

Figure 7. thread graph of Adultbrowser malware [29].

Linked graph: the technique of linked graph is used to visualize data having hierarchical or network relationship. The graph includes a group of nodes and a group of edges, where edges are utilized to connect identical nodes. Different dimensions of data may be displayed by utilizing the position, size, and colour of a node [30]. Figure 8 shows a visualization example of linked graph displays the thorough malicious hostname and determined IPs relationships linked to malware MD5.

Figure 8. Linked graph displays the thorough malicious hostname and determined IPs relationships linked to malware MD5.[31].

Maps: maps are globally recognized and can be utilized as background in graphics that include geographic data. Figure 9 shows a visualization instance of a geographically-located malicious accesses to a set of sensors with markers proportional to the quantity of accesses[9].

Images: Image-based technique Visualizing malware executables as grayscale images. they use visual charts to recognize an image for every malware sample [27]. Some systems visualize the binary information and instantly plot the (raw) byte-code representation or particular entropy values to an image [32]. Figure 10 shows images representing various malware families.

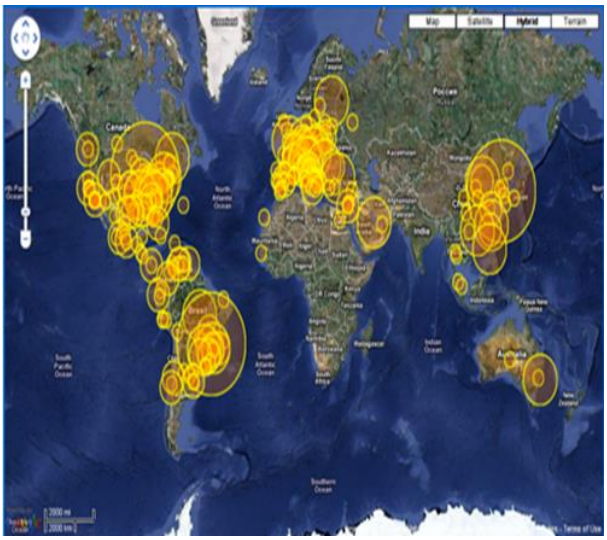

Figure 9. Malware download sources (IP addresses)[9].

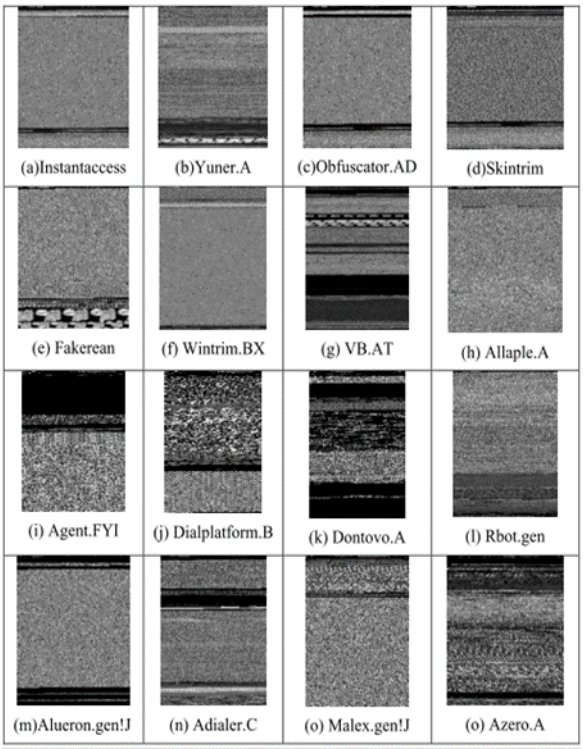

Figure 10. Images represent various malware families [33].

Parallel Coordinates: One of the most useful visualizations to investigate data packets. Useful to represent a multidimensional data, interactive analysis, and parallel coordinate plot [34]. Figure 11 shows a sample of a parallel coordinate graph.

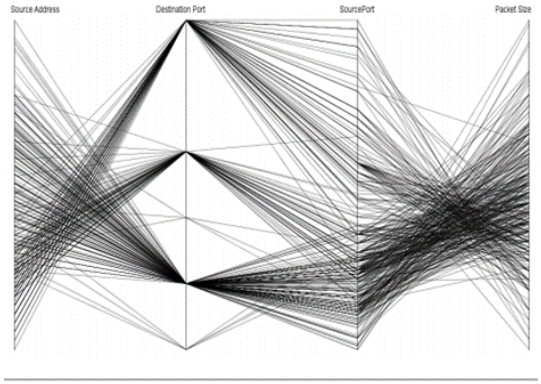

Figure 11. A parallel coordinate graph [34].

- *Histogram:* To display the data distribution (i.e., the frequency of the individual data values).
- *Scatter plot:* Suited for comparison with each other, useful to discover outliers and anomalies, and used for continuous or ordinal data type.
- *Pie Chart:* To visualize distribution of quantities as part of the overall. Pie charts are good for categorical variables data type.
- *3D DNA:* the technique of 3D-DNA is used to visualize the behavioral features and sequences of the processes running on the host. When malware is detected a 3D behavioral sequence chains is generated and the similarity between the behavioral sequence chain and the sequence of a target process is calculated [35]. Figure 12 represents a scene of both 3D DNA structure and behavioural sequence.

3.3. Visualization Analysis in Windows 10 and some Security Tools

Windows 10 and various security tools use visualization techniques to generate graphical representations of log files and for analyzing the security events such as malware analysis. Although the Windows 10 and many other visualization tools are a data driven whereby specific visualization techniques (bar chart, line trend or pie chart) were used for visualizing the events. It is worth to learn from them how they designed the dashboards. For example, windows 10 designed the dashboard of task manager (Resource monitoring) based on the resources to include CPU, Memory, Disk and Network. See Figure 13. In CPU tap, a line graph is produced to monitor the CPU performance at run time. Almost a similar way for monitoring the other resources.

Kaspersky provides almost a similar dashboard as extra tool for monitoring the activates of some resources such as Application control and network monitoring. See Figure 14. Therefore, the idea of windows 10, or Kaspersky could be exploited to build and design dashboards for visualization analysis.

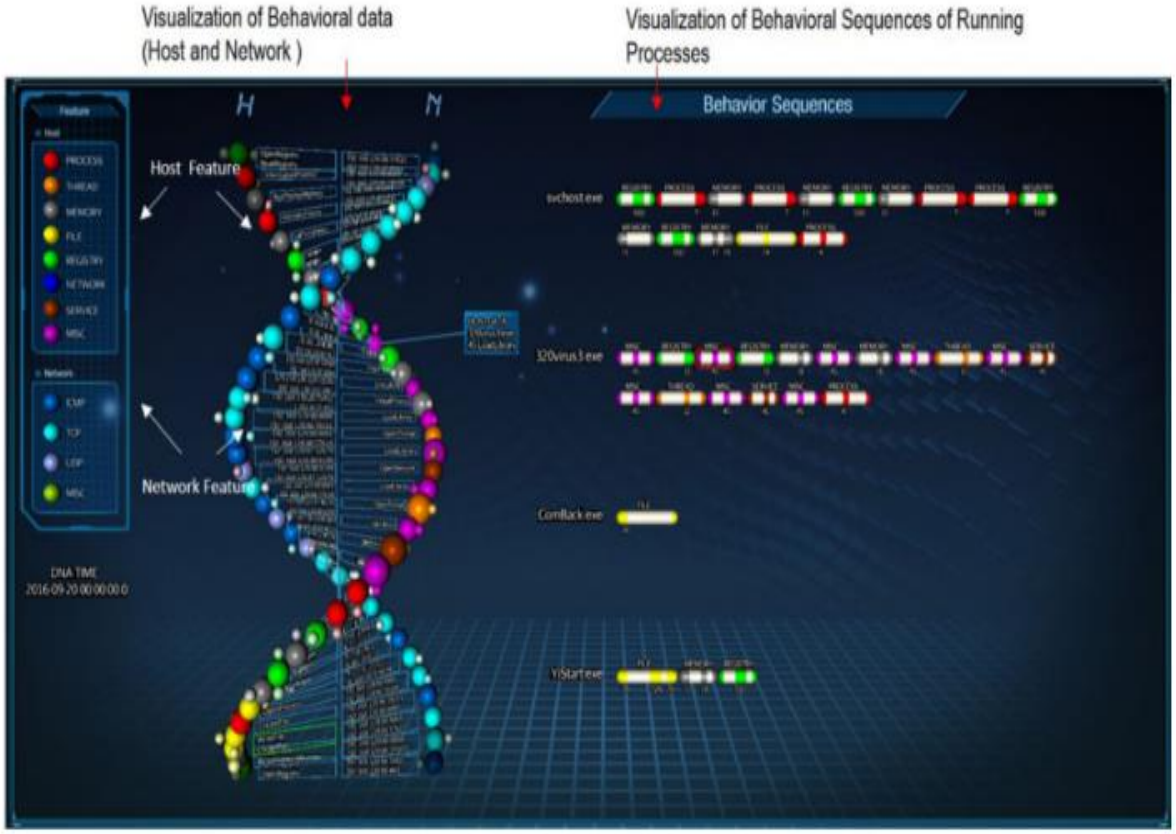

Figure 12. DNA 3D structure and behavioral sequence [35]

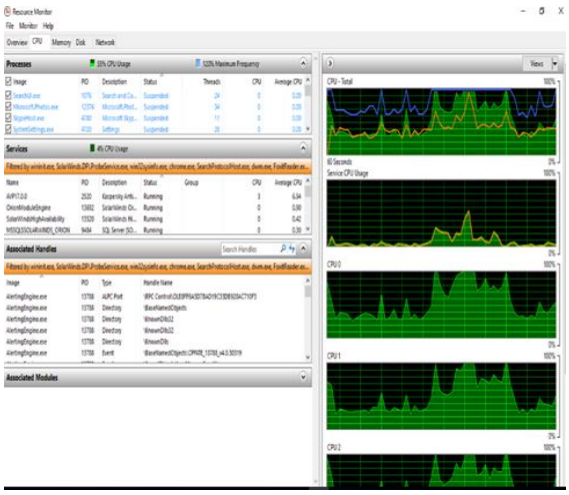

Figure 13. dashboards in windows 10.

Figure 14. dashboards in Kaspersky

Kaspersky Lab presents its new interactive Cyber threats Real-time Map, this visual tool allows users to see what is going on in cybersecurity around the world in real time.

Cyber threats Realtime Map allows users to compare different types of threats and their distribution around the world at any given time. It's pretty apparent that the amount of spam, malware infection rates vary according to the time of the day in any given region [36].

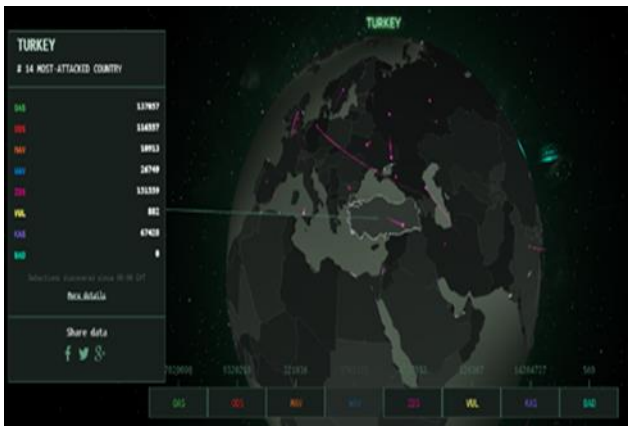

Figure 15. Cyber threats Real-time Map [36].

Thousands of security events can be reviewed simply and quickly by making a graphical image of the data using tools available in the Data Analysis & Visualization Linux (DAVIX), which is a live CD for data analysis and visualization. For instance, Trenton in [37] uses tools such as AfterGlow and Graphviz from DAVIX to visualize the Cisco firewall family (FWSM, ASA, PIX) log data samples, and highlighting areas of potential intrusion.

The following list presents the important tools in DAVIX [34, 38]:

- *AfterGlow*: Tool to convert CSV input to a DOT graph description. It facilitates the process of generating link graphs.
- *Graphviz*: Tool to generate a two-dimensional link graphs.
- *ChartDirector*: Programming library to generate a wide variety of charts.
- *Cytoscape*: Tool for generation and display of two-dimensional link graphs.
- *EtherApe*: Tool for real-time visualization of network traffic.
- *GGobi*: A general-purpose visualization application that can visualize information in a diversity of ways: line charts, bar charts, parallel coordinates.
- *glTail*: Tool for real-time visualization of web server traffic.
- *GNUplot*: Tool for plotting mathematical functions. It generates various types of charts.
- *GUESS*: Tool to display and interaction with two-dimensional link graphs. Has a capability to use a scripting language to process graphs.
- *InetVis*: Tool for real-time visualization of network traffic as a three-dimensional scatter plot.
- *Large Graph Layout – LGL*: Tool for generation of two- and three-dimensional link graphs.
- *Mondrian*: Tool for generation and display of a variety of charts that are linked.
- *MRTG*: Tool for visualization of traffic load on network devices using SNMP queries.
- *NVisionIP*: Tool for Animated two-dimensional scatter plot of ARGUS files.
- *Parvis*: Tool for rendering of data as parallel coordinate display.
- *Ploticus*: Tool for generation of all kinds of charts.
- *R Project*: Tool for statistical analysis that offers a great variety of graphing capabilities.
- *RRDtool*: A tool for graphing time series data.
- *RT Graph 3D*: Tool for real-time 3D visualization of linked graphs.
- *Rumint*: Tool for visualization of real-time and recorded network captures.
- *Scapy*: Tool to capture and manipulation of TCP/IP traffic, and visualization of traceroutes.
- *Shoki Packet Hustler*: Tool for visualization of network

traffic as a three-dimensional scatter plot.

- *Treemap*: Tool for visualization of hierarchical data as treemaps.
- *Tulip*: Visualization tool for linked graphs that supports several layout algorithms.
- *Walrus*: Tool for visualization of hierarchical data as three-dimensional link graphs.
- *FlowTag*: An interactive network trace viewer.
- *Picviz*: Software for transforming the acquired data into a parallel coordinates plot image.

Visualization of data is not always a straightforward process [39]. It is important that the problem or objective is very clear to start with. By other words, visualization technique is mainly guided by the problem statement, the dataset and the tasks that are to be accomplished [40]. Visualization tools must answer the following questions which extracted from [38] and [39]:

When is, the attack happening?

Where in (the network, the memory, disk, CPU, etc.) the attack happening? For example, is the malware affected the memory, the network, the disk, etc.?

What type of attack is happening? If it is Malware try to go deeper and classify or provide some details about it.

As a result, Visualization techniques have many applications include not limited to: view static data [41], monitor network traffic [42], visualization of software security [43], visualization of Cybersecurity data [44], managing networks [45] and recently visualize malware behaviors [46-48].

Overall, malware is a serious issue in private or public sectors. Different techniques have been applied to detect, classify and identify malware. Among several detection techniques, visualization-based technique becomes the most attractive one. Malware visualization is an area that concentrates on discovering, classifying and symbolizing malware features in a shape of visual that could be utilized to transfer more data on a specific malware. Regardless the visualization methods whether if graph, map, etc. most of them have been used to visually detect, classify or identify malware. However, this illustrates with strong evidence the usefulness of visualization techniques not only in detecting malware but also in several other applications.

3.4. Malware Types

As reported by [49] [28] [3], malware has different types. Therefore, this section aims to exploring the most common types of malwares as well as providing brief description for each type in term of capability. Besides that, it aims to illustrate with example the most common malwares in order to provide the reader with brief knowledge on the malware types.

There is no doubt about having many types of malware as confirmed by many PSs [50]. In addition, malware families may shows diverse actions in their lifetime extend from sheer upsetting to highly malicious. However, grouping or categorizing malware types could be done based on functionality, behavior, platform or capability. In contrast to previous related work, here the focus is on the type of malware that mostly investigated and reported in order to provide the reader with brief description and capability on each.

1. *Viruses*
2. *Worms*
3. *Trojan Horses*
4. *Adwares*
5. *Spyware*
6. *Rootkits*
7. *Backdoor*
8. *Ransomware*
9. *Botnets*
10. *Keyloggers*
11. *Phishing Apps*
12. *Malware installation*
13. *Privilege escalation*
14. *JavaScript*
15. *VBScript*
16. *HTML Script*
17. *Macro*
18. *Browser Hijacker*

Based on the PSs, these are the most reported and discussed malwares. However, Trend Micro Encyclopedia web site and some other security projects such as Internet security threat report, Annual Cyber Threat Reports, Open Web Application Security Project (OWASP), Web Application Attack Report (WAAR) and Symantec provide more details and knowledge about malware.

3.5. Features that can be used for an Effective Visualization Analysis

Feature extraction is the key for the success of any visualization system. Based on this idea, it is important to explain the most popular and beneficial features that have been utilized for visualize the actions or to analyze the device performance. According to [13] features can be classified into four kinds include; static, dynamic, hybrid and applications' metadata. Figure 16 shows different kinds of features and subtypes of each kind.

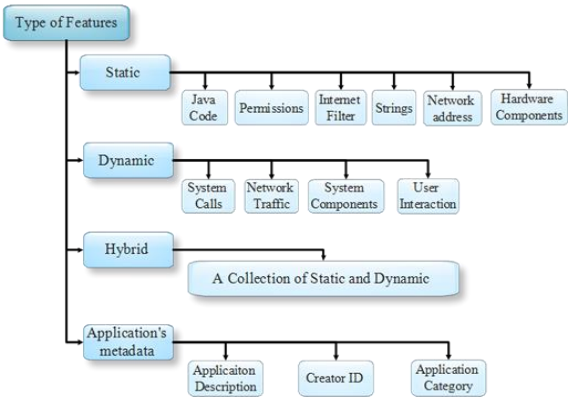

Figure 16. Most common extracted features that used for analyzing security events (Based on [13]).

Next sections discuss each kind of features in detail:

A) Static features:

These features are taken from the software's available feature. Based on the extraction process static features can be classified to:

1. *Portable Executable (PE):* The dynamic link library DDL information inside PE executables stored in Win32 PE binaries are used to produce extracted features. [51]
2. *Byte-sequence (n-grams):* This approach utilizes concatenation of n bytes taken away from an executable program.
3. *String features:* This approach is based on text strings which are encoded in the files like printable string data. [53] and [54] declared that, string features is the best precise feature which accomplished a detection average of 97.43% and a false positive average of 3.80%.
4. *OpCode (operational code):* static data utilized to figure the cosine likeness among two PE executables.
5. *Function-based techniques:* In such technique, functions are taken out from the files that are executable then utilize these functions to yield different features like the length of the function that measured via the size of code in it, and the length frequency of the function in any file.
6. *Intent Filter:* One of the elements that identified by the manifest file is an Intent filter, which is an abstract data around a procedure demanded, that deduce the applications goal. Intent filter in Andorid like chosen a contact, take a photo, dial a number, web pages links, etc. appropriate activity could be taken based on intent filters.
7. *Network Address:* Malware can be utilized to record the victims' effectiveness and situation, or user's private data and transmit it to its creator. Searching for IP address of network in code is significant for accomplishing analysis.
8. *Hardware Components:* the camera or GPS are an example of hardware that can be utilized by a malware to reports the location of the user.

To the best of our knowledge, these are the most common static features that are used to analyze the code without executing programs.

B) Dynamic (run-time) features:

Dynamic features are known as the actions of the application in dealing with operating system or network connectivity [13]. Numerous dynamic features utilized in modern works, these features can be classified to:

1. *Network traffic:* Observing network traffic of the devices is an applicable way for visualization analytic [13]. Network data are summarized by [52] IP addresses, Port numbers incoming/outgoing traffic in Bits per Second and TCP packet data. As example, [53] uses the IP addresses, ports number, and protocols to visualize their activities via bar charts. Mapping Port Activity of Network Traffic
2. *System calls:* [14] reported 22 studied papers were based on system calls. From Application Programming Interface (API) calls, many features can be extracted such as sequenced events, sample of rootkits that use inline function hooking. The concept is to execute files to produce lists of API calls and then calculated the similarity between two API call sequences by using a similarity matrix.
3. *System Components:* system components could be used to extract useful features such as the usage of CPU, memory access, free memory, running processes besides to battery statues (for chargeable devices), Bluetooth and Wi-Fi statues. The visualizing these features could provide useful way especially for knowledgeable persons.
4. *User Interaction:* One of the dynamic features is the user's interaction with applications, this features can be helpful to visualization analytic. For instance, the response of the users to some applications can be utilized to estimate the behaviors of that program. However, such features are restricted for some devices only and operating system based (e.g. pushing a button, zooming, tapping the screen, long pressing, dragging and navigating through pages).

C) Hybrid *features*:

Hybrid features is defined as a combination of static and dynamic features which are utilized with each other for visualization analytic. They are the most comprehensive features, since they involve examining the installation of the file and at runtime analyzing the behavior of that file.

D) *Applications'* metadata:

The metadata is the information users see before the download and installation of the applications, like the applications description, their requested permissions, their rating and information regarding developers, package name, installation size, version, application type, contact website, count and application title. Such features classified as non-static and non-dynamic as they have nothing to do with applications themselves. As stated by [14], a few researchers depend on application's metadata for extracting features. The reason is that, these features may provide implausible information mostly exploits the weakness of user's knowledge. They intended in most cases as promoting information for that produce. However, in many cases the intruder software makers intentionally provided such convenient information.

Feizollah, Anuar [13] statically analyzed the latest outcomes based on the kind of features they utilized and summarized that as shown in **Hata! Başvuru kaynağı bulunamadı.**. Many studies confirmed that dataflow-related Application Program Interfaces (APIs) are the most noteworthy features.

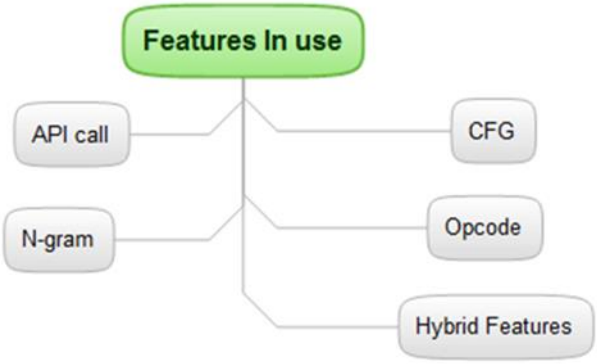

Figure 17: Recent statistical analysis based on type of features [13].

There are some features considered by [2] for malware detection represented shown in Figure 2.

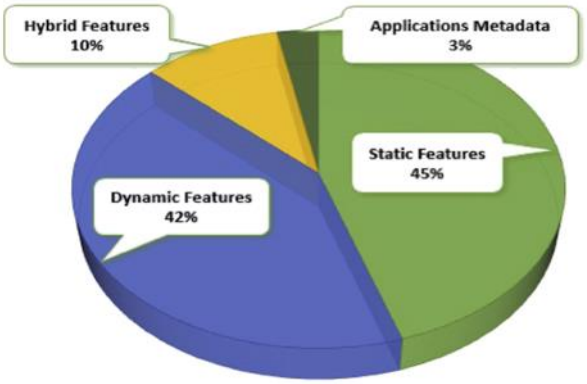

Figure 2. Features in use for malware detection [2]

3.6. Data Sources of the Extracted Features

Based on the review work in [25], the following data sources are the main foundation for extracting useful information to be visualized and used for monitoring malware behaviors.

1. *Host/Server Monitoring:* In this visualization way, the key exhibit is given to the exemplification of hosts and servers. he purpose is to show the running situation of a network via visualizing the number of users, system load, status, and uncommon or unforeseen host or server actions. The capability of the visualization systems of this type in showing a limited number of hosts or servers within the observed network is a tangible matter. Most of the systems of this type are constrained by their combined visualization techniques. For a near real time, test of events and a more reacting system. Server logs, packet traces, and network flows comprise essential data sources for this type of visualizations. Node link graphs, glyphs, and scatter plots are also primary visualization techniques integrated in this class
2. *Internal/External Monitoring:* the visualization systems of this class interested with the interaction of internal hosts with respect to external IPs. The capability of the visualization systems of this class relies on two components; (1) operation that automatically characterize and assess the effect of underlying events, (2) exploratory system that supplies the instrument for an analyst to prove different hypotheses. Common visualization techniques such as Color maps, radial panels, scatter plots, and parallel coordinates are utilized in this class. Packet traces and network flows are also used as the main data sources for visualizations of this class
3. *Port Activity:* Visualizations of this category will aid within the discovery of malicious computer code running within a network. In this category a Scaling techniques is combined within the design of visualizations, due to the quantity of traffic also because large range of possible port numbers and IP addresses. Histograms and scatter plots are the two outstanding visual techniques of this category. While, packet traces and network flows are the most information sources.
4. *Attack Patterns:* Visualizations of this category assist a director in not only the detection of attacks but also the exhibit of multistep attacks. Scatter plots, glyphs, color maps, and parallel coordinates are the widespread visual techniques of this category.
5. *Routing Behaviour:* The major purpose of this visualization category is to recognize the growth of Border Gateway Protocol (BGP) routing patterns over time.

4. Discussion

This section discusses and interprets the results reported in section 3.

4.1. Malware Detection Techniques and visualization Techniques Related to RQ1

In this SLR, based on 90 articles different malware detection techniques have been explored including Signature-Based, Behavior-Based, Analysis-Based, Anomaly-Based and Visualization-Based. The results illustrate that the visualization technique is the most used method. This method is the most common due to the verities of its applications besides the following advantages:

1. It could be easily to automate and utilize the visualization technique to analyze a large number of malware [8].
2. In testing malicious software, visualization-based techniques have proved great usefulness [54].
3. Using visualisation of program execution for finding out and monitor program execution has been utilized in the past with sensible results [23].
4. Visualization techniques not require unpacking or decryption as well as can apply widely used image

processing techniques like textures analysis [55].

In addition, there is different visualization techniques can be used easily by expertise in the field or even who have few knowledge about it. Many visualization techniques such as images, graphs, plots, maps, and others are effective method to detect malware with several visualizing methods. Finally, as illustrated by **Hata! Başvuru kaynağı bulunamadı.** visualization techniques still the best solution among the rest.

4.2. Malware types and features extraction Related to RQ2

Several malwares emerged in almost in all platforms. Categorizing malware could be done based on functionality, behavior, platform or capability. Based on the SLR, most of PSs reported and discussed malwares namely; viruses, worms, Trojan, spyware, adware and rootkits. In addition, Many of PSs confirm that dataflow-related Application Program Interfaces (APIs) are the most noteworthy features.

5. Conclusion

In this work, SLR conducted to investigate the current state of knowledge about Malware detection techniques, data visualization and malware features. 90 primary studies have been identified in accordance with our review protocol and published between 2009 to 2018. The reported results prof that the Malware is a serious issue. Also visualization techniques are recently applied to the field and noticeably considered as the most common technique compared to the others. In addition, among several visualization techniques, graphs and images are the most used visualization techniques. Furthermore, different method has been used to extract information about the malware from different data sources. Extracting features from static and dynamic group are the most useful features. Finally, it is confirmed that dataflow-related APIs are some of the most noteworthy features.

References

[1] Zhang, Y., et al., *A survey of cyber crimes.* Security and Communication Networks, 2012. **5**(4): p. 422-437.

[2] Bazrafshan, Z., et al. A survey on heuristic malware detection techniques. in The 5th Conference on Information and Knowledge Technology. 2013.

[3] La Polla, M., F. Martinelli, and D. Sgandurra, *A Survey on Security for Mobile Devices.* IEEE Communications Surveys & Tutorials, 2013. 15(1): p. 446-471.

[4] Meng, G., et al., Mystique: Evolving Android Malware for Auditing Anti-Malware Tools, in Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. 2016, ACM: Xi'an, China. p. 365-376.

[5] Vemparala, S., et al., Malware Detection Using Dynamic Birthmarks, in Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics. 2016, ACM: New Orleans, Louisiana, USA. p. 41-46.

[6] Dang-Pham, D. and S. Pittayachawan, Comparing intention to avoid malware across contexts in a BYOD-enabled Australian university: A Protection Motivation Theory approach. Computers & Security, 2015. 48: p. 281-297.

[7] Meng, G., et al., Semantic modelling of Android malware for effective malware comprehension, detection, and classification, in Proceedings of the 25th International Symposium on Software Testing and Analysis. 2016, ACM: Saarbrücken, Germany. p. 306-317.

[8] Han, K., J.H. Lim, and E.G. Im, Malware analysis method using visualization of binary files, in Proceedings of the 2013 Research in Adaptive and Convergent Systems. 2013, ACM: Montreal, Quebec, Canada. p. 317-321.

[9] Grégio, A.R.A. and R.D.C. Santos. Visualization techniques for malware behavior analysis. in SPIE Defense, Security, and Sensing. 2011. SPIE.

[10] Kitchenham, B. and S. Charters, Guidelines for performing systematic literature reviews in softwareengineering, Technical Report EBSE-2007-01 Ver. 2.3, School of Computer Science and Mathematics, Keele University

[11] K.K., P., B. N.M.W.M., and D.V. N.K., Systematic review: School health promotion interventions targeting physical activity and nutrition can improve academic performance in primary - and middle school children. Health Education, 2013. 113(5): p. 372-391.

[12] Shea, B.J., et al., Development of AMSTAR: a measurement tool to assess the methodological quality of systematic reviews. BMC Medical Research Methodology, 2007. **7**(1): p. 10.

[13] Feizollah, A., et al., *A review on feature selection in mobile malware detection.* Digital Investigation, 2015. 13: p. 22-37.

[14] Ye, Y., et al., A Survey on Malware Detection Using Data Mining Techniques. ACM Comput. Surv., 2017. 50(3): p. 1-40.

[15] Jacob, G., H. Debar, and E. Filiol, *Behavioral detection of malware: from a survey towards an established taxonomy.* Journal in Computer Virology, 2008. 4(3): p. 251-266.

[16] Elhadi, A., M. Maarof, and A. Hamza Osman, Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph. Vol. 9. 2012. 283-288.

[17] Idika, N. and A. Mathur, *A survey of malware detection techniques*. 2007: Department of Computer Science, Purdue University.

[18] Zolkipli, M.F. and A. Jantan. Malware Behavior Analysis: Learning and Understanding Current Malware Threats. in 2010 Second International Conference on Network Applications, Protocols and Services. 2010.

[19] Rana, H. and M. Stamp, *Hunting for Pirated Software Using Metamorphic Analysis.* Information Security Journal: A Global Perspective, 2014. 23(3): p. 68-85.

[20] Moser, A., C. Kruegel, and E. Kirda. Limits of Static Analysis for Malware Detection. in Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007). 2007.

[21] Markel, Z.A., Machine Learning Based Malware Detection Trident Scholar Report 2015 no. 440 U.S. Naval Academy Annapolis, MD 21402

[22] Pektaş, A. and T. Acarman, *Malware classification based on API calls and behaviour analysis.* IET Information Security, 2018. 12(2): p. 107-117.

[23] Chan Lee, Y., et al. A static and dynamic visual debugger for malware analysis. in 2012 18th Asia-Pacific Conference on Communications (APCC). 2012.

[24] Lee, D., et al. A Study on Malicious Codes Pattern Analysis Using Visualization. in 2011 International Conference on Information Science and Applications. 2011.

[25] Shiravi, H., A. Shiravi, and A.A. Ghorbani, *A survey of visualization systems for network security.* IEEE Transactions on visualization and computer graphics, 2012. 18(8): p. 1313-1329.

[26] Shaid, S.Z.M. and M.A. Maarof. Malware behavior image for malware variant identification. in Biometrics and Security Technologies (ISBAST), 2014 International Symposium on. 2014.

[27] Wagner, M., et al., A Survey of Visualization Systems for Malware Analysis. 2015.

[28] Bazrafshan, Z., et al. A survey on heuristic malware detection techniques. in Information and Knowledge Technology (IKT), 2013 5th Conference on. 2013.

[29] Trinius, P., et al. Visual analysis of malware behavior using

treemaps and thread graphs. in 2009 6th International Workshop on Visualization for Cyber Security. 2009.
[30] Herman, I., G. Melancon, and M.S. Marshall, *Graph visualization and navigation in information visualization: A survey*. IEEE Transactions on Visualization and Computer Graphics, 2000. **6**(1): p. 24-43.
[31] Cheng, j.y. *HpfeedsHoneyGraph - Automated Attack Graph Construction for Hpfeeds Logs*. 2012; Available from: https://www.honeynet.org/node/957.
[32] Han, K., B. Kang, and E.G. Im, *Malware Analysis Using Visualized Image Matrices.* The Scientific World Journal, 2014. p. 15.
[33] Nataraj, L., et al., Malware images: visualization and automatic classification, in Proceedings of the 8th International Symposium on Visualization for Cyber Security. 2011, ACM: Pittsburgh, Pennsylvania, USA. p. 1-7.
[34] Marty, R., *Applied Security Visualization*. 2008: Addison-Wesley Professional.
[35] Kim, H., et al., Improvement of malware detection and classification using API call sequence alignment and visualization. Cluster Computing, 2017.
[36] Kaspersky. *Cyberthreats Map: watch global threats in real time*. 2014 29/03/2018]; Available from: https://cybermap.kaspersky.com/.
[37] Bond, T. *Visualizing Firewall Log Data to Detect Security Incidents*. 2009 29-03-2018]; Available from: https://www.sans.org/reading-room/whitepapers/metrics/security-data-visualization-36387.
[38] Attipoe, A.E., et al., *Visualization Tools for Network Security.* Electronic Imaging, 2016. (1): p. 1-8.
[39] Marty, R., *Applied security visualization*. 2009: Addison-Wesley Upper Saddle River.
[40] Muhammad, T. and Z. Halim, Employing artificial neural networks for constructing metadata-based model to automatically select an appropriate data visualization technique. Applied Soft Computing, 2016. p. 365-384.
[41] Medvedev, G.D., M. Virginijus, and Viktor, *Web Application for Large-Scale Multidimensional Data Visualization.* http://dx.doi.org.ezproxy.psz.utm.my/10.3846/13926292.2011.580381, 2011.
[42] Shabtai, A., et al., Monitoring, analysis, and filtering system for purifying network traffic of known and unknown malicious content. Security and Communication Networks, 2011. 4(8): p. 947-965.
[43] Chen, Y., et al. Multiple sequence alignment and artificial neural networks for malicious software detection. in 2012 8th International Conference on Natural Computation, ICNC 2012. 2012. Chongqing.
[44] Metcalf, L. and W. Casey, Chapter 7 - Visualizing cybersecurity data, in Cybersecurity and Applied Mathematics. 2016, Syngress: Boston. p. 113-134.
[45] Liao, Q., et al., Managing networks through context: Graph visualization and exploration. Computer Networks, 2010. 54(16): p. 2809-2824.
[46] Han, K., B. Kang, and E.G. Im, *Malware analysis using visualized image matrices.* ScientificWorldJournal, 2014: p. 132713.
[47] Han, K.S., et al., *Malware analysis using visualized images and entropy graphs.* International Journal of Information Security, 2015. 14(1): p. 1-14.
[48] Blank, D., A. Henrich, and S. Kufer, Using Summaries to Search and Visualize Distributed Resources Addressing Spatial and Multimedia Features. Datenbank-Spektrum, 2016. (1): p. 67-76.
[49] Idika, N. and A.P. Mathur, *A survey of malware detection techniques.* Purdue University, 2007. .
[50] Somarriba, O., et al., *Detection and Visualization of Android Malware Behavior.* Journal of Electrical and Computer Engineering, 2016.
[51] Zhao, Z., J. Wang, and J. Bai, *Malware detection method based on the control-flow construct feature of software.* IET Information Security IEEE, 2014. 8(1): p. 18-24.
[52] Corchado, E. and Á. Herrero, *Neural visualization of network traffic data for intrusion detection.* Applied Soft Computing, 2011. **11**(2): p. 2042-2056.
[53] Kiran, L., et al. Closing-the-loop in NVisionIP: integrating discovery and search in security visualizations. in IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05). 2005.
[54] Conti, G., et al. Visual Reverse Engineering of Binary and Data Files. in Visualization for Computer Security. 2008. Berlin, Heidelberg: Springer Berlin Heidelberg.
[55] Kancherla, K. and S. Mukkamala. Image visualization based malware detection. in 2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS). 2013.
[56] Ecemiş, A. , Küçüksille, E. U. , Yalçınkaya, M. A. "*Yaygın Görülen Dosya Enjeksiyon Zararlılarının Analizi ve Sistematik Olarak Tespiti*". Niğde Ömer Halisdemir Üniversitesity, Journal of Engineering Sciences, 7/2 2018:478-489. https://doi.org/10.28948/ngumuh.443149