Analysis of Feature Categories for Malware Visualization

Ganthan Narayana Samy[1], Pritheega Magalingam[1], Aswami Fadillah Mohd Ariffin[2], Wafa Mohd Khairudin[2], Mohamad Firham Efendy Md Senan[2], Zahri Hj Yunos[2]

[1]Advanced Informatics School, Universiti Teknologi Malaysia (UTM AIS), Malaysia.

[2]Cyber Security Malaysia (CSM), Malaysia.

ganthan.kl@utm.my

***Abstract*—It is important to know which features are more effective for certain visualization types. Furthermore, selecting an appropriate visualization tool plays a key role in descriptive, diagnostic, predictive and prescriptive analytics. Moreover, analyzing the activities of malicious scripts or codes is dependent on the extracted features. In this paper, the authors focused on reviewing and classifying the most common extracted features that have been used for malware visualization based on specified categories. This study examines the features categories and its usefulness for effective malware visualization. Additionally, it focuses on the common extracted features that have been used in the malware visualization domain. Therefore, the conducted literature review finding revealed that the features could be categorized into four main categories, namely, static, dynamic, hybrid, and application metadata. The contribution of this research paper is about feature selection for illustrating which features are effective with which visualization tools for malware visualization.**

***Index Terms*—Features; Malware; Malware Visualization; Visualization Tools.**

I. INTRODUCTION

The features play a significant role in the visualization analytical tool. Majority of the visualization systems are data-driven [1]. Visualizing the extracted features of the software may classify the activities and the behaviors of that software between normal and malicious activities. Selecting the best features to be visualized is not an easy task. Therefore, it is very difficult to decide the number of features or to specify which features to be visualized for the purposes of analytical descriptions, diagnostic, or prediction [2]. Besides that, some features require a clear pre-understanding of malware families, symptoms, unique features, and the diversity of the sample and the existence of a modification in the malicious application [3].

Visualization of data analysis helps to identify patterns, trends, structures of the malware. Visualization is efficient to ensure that the analysis is meaningful and shows the descriptive, diagnostic, predictive and prescriptive analysis effectively. A single graph or picture can potentially describe a year’s worth of malware activities (depending on the type and number of malware), and present patterns, trends, structures, and exceptions. This is easier than scrolling multiple extracted features of audit data with a minimum sense of the underlying events. However, visualization is still a new term in an information security domain [1] specifically for visualizing features to gain intuition about the malware. This is due to common visualization techniques have been designed for use-cases which are not supportive of security-related data that demands visualization techniques fine-tuned for descriptive, diagnostic, predictive and prescriptive analytics. It may not be possible to fully predict how an end user will perceive and interpret visualization due to the varying nature of audience’s cognitive characteristics. However, careful consideration of the user’s needs, cognitive skills, and abilities can determine the appropriate content and design.

Visualization techniques, design process centered on the needs, behaviors, and expectations of security analysts that can influence and impact the usability and practicality of developing the desired visualization techniques. Nevertheless, developing visualization techniques for multivariate data will be hard enough without providing an in-depth understanding of the available types of the visualization tool, applications and data needed of each tool, besides the extensive hands-on experience. Visualization or scientific visualization analyses the data and represent it as information to generate the output of the analysis in the form an image or graphic, to show the physical phenomena or physical quality changing with time and space.

This paper is organized into four sections. Section II describes the literature review related to malware visualization features. Section III further discusses extracting features based on four main categories and followed by conclusion in section IV.

II. LITERATURE REVIEW

A. Definition of Malware and Malware Visualization

Malware, stand for "malicious software," defined as a type of computer program designed to infect a legitimate user's computer and inflict harm on it in multiple ways [4]. Malware can infect computers and devices in several ways and comes in different forms including, viruses, worms, trojans, and spyware. According to [5] there is a countless number of malware reported cases every year which are related to malicious activities, for example stealing users' data by hackers and system damage [6]. Moreover, based on the SophosLabs 2018 malware forecast report has identified Android malware as one of the trends that remains challenging in 2018 besides other threats [7]. Malware visualization is a field that focuses on detecting, classifying and representing malware features in the form of visual cues that can be used to convey more information about a particular malware [8]. Visualization techniques have been applied to view static data, monitor network traffic or manage networks. Furthermore, they are also applied to detect and visualize the behavior of the malware [9]. According to [10] visualization technique is used to differentiate between malware dataset to identify important malware behavior patterns.

B. Malware Visualization Features

Several features can be extracted to visualize the device activates. The success of visualization system depends on the extracted features. It is significant to illustrate the most common and useful features that are used for visualizing the activities or for analyzing the device performance. According to [5] features are classified into four main categories namely, static, dynamic, hybrid and applications' metadata that can be used for the visualization as discussed in the following subsections.

C. Static Features

Static features are extracted from the available features of the software [4][11][12]. Classifications of static features mostly are based on the extraction process as listed below:

i. Portable Executable (PE): Features are extracted from the Dynamic Link Library (DLL) information inside PE stored in Win32 PE binaries [13].

ii. Byte-sequence (n-grams): The byte sequence approach uses the sequences of n bytes extracted from an executable file.

iii. String features: is based on text strings that are encoded in the program files such as printable string information. [14] Stated that string features are the most accurate feature that has a detection rate of 97.43% with a false positive rate of 3.80%.

iv. OpCode (Operational Code): is used as static information to calculate the cosine similarity between two PE executables.

v. Function-based feature extraction techniques [15][16]: the functions are extracted from a binary file and are used to produce various attributes such as function length, which is measured by the number of bytes of code in it and the function length frequency within any file. These attributes are used for analysis. For example, visualizing feature interaction in 3D, the classes are displayed as 3D nodes to show the inheritance relationships between shared classes as connecting edges.

vi. Intent Filter: The intent filter is one of the elements described in the manifest file. It is an abstract information about an operation request, which infers the intentions of the applications. Intent filter in Android such as pick a contact, take a photo, dial a number, web page links, etc. The appropriate action is taken based on the intent filters.

vii. Network Address: An instructed malware is used to contact back the producer and report the victims' activities, status or personal data. Looking for the network address of the IP address in code is important for preferment analysis.

viii. Hardware Components: Applications request combinations of hardware which are needed to function, for example, the camera or GPS. Combinations of requested hardware imply harmfulness of the application, such as, 3G and GPS access imply a malware that reports the location of the user to the attacker.

D. Dynamic (Run-Time) Features

Dynamic features refer to the behavior of the application that interacts with the operating system or network connectivity. There are two main types of dynamic features used in recent works namely are system calls and network traffic besides other dynamic features [4][5][10][11][12].

i. Network traffic: A dynamic feature used by the researchers since most applications tend to connect to the network to send and receive data, and updates, or maliciously leak personal data to attackers. Monitoring network traffic of the devices is useful for visualizing analytic. [5] stated that out of 42 papers for the dynamic feature, 10 papers were based on network traffic monitoring. Consequently, features extracted from network traffic are also useful for visualization analytic.

ii. System calls: Every application demands resources and services from the operating system. For instance, several features from Application Programming Interface (API) calls can be extracted such a sample of rootkits that use inline function hooking. The idea is to execute the files to generate lists of API calls and then calculate the similarity between two API call sequences by using a similarity matrix. Reported 22 out of 42 studied papers were based on system calls [5].

iii. System components: they could be used to extract useful features such as the usage of CPU, memory access, free memory, running processes besides to battery status (for chargeable devices), Bluetooth and Wi-Fi status. The visualizing these features can be useful, especially for knowledgeable persons. The task manager of the devices that run Windows operating system is an example for the visualizing CPU, Memory, Disk space, Wi-Fi, and Ethernet.

iv. User interaction or observing the behavior of the user: Extracting user's interaction with applications is one of the dynamic features that may enable visualization analytic. For example, the response of the users (e.g. pushing a button, zooming, tapping the screen, long pressing, dragging and navigating through the pages) against some applications can evaluate the behaviors of that program. However, these features are limited for some devices only and based on operating system type.

E. Hybrid Features

It is defined as a group of static and dynamic features used for visualization analytic. They are the most comprehensive features since they involve vetting the file's installation as well as analyzing the behavior of that file at runtime.

F. Applications Metadata

The metadata refers to the information users see prior to the download and installation of the applications, such as the application description, the requested permissions, the information regarding developers, package name, installation size, version, application type, contact website, count and application title. These features categorized as non-static and non-dynamic as they have nothing to do with application themselves. As reported by [5], 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 the user's knowledge. They intended in most cases as promoting information for that product. However, in many cases, the intruder software makers intentionally provided such convenient information.

G. Data Visualization Techniques and Classification

Visualization techniques can be applied to security events which is a useful technique for identifying suspicious activities and responding to an incident in a timely manner. Therefore, this technique is used to analyze and classify the nature of malware activities [17]. Visualization techniques are divided into five different classes since most of visualization systems are data-driven. Figure 1 summarizes the classification of the visualization techniques based on the data source.

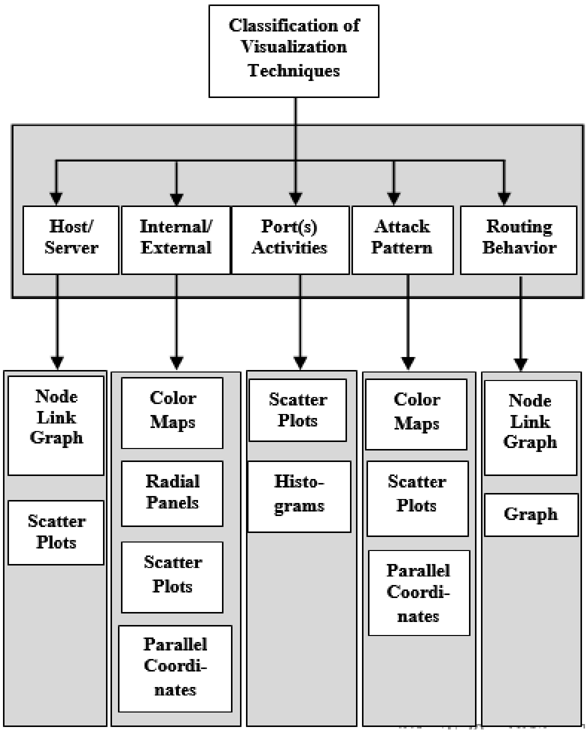

Figure 1: Classification of Visualization Techniques Based on Data Source

III. Discussion

The extracted features based on four main categories, namely static, dynamic, hybrid and applications' metadata mostly transferred to a proper dataset. The datasets have different characteristics as summarized in Table 1 in term of dimension, primary variables or data type, tasks, number of attributes and instances and what type of visualization tools. The dataset could be used for several tasks such as showing the relationship, comparison, distribution or trends. Accordingly, to [18] provide a description of the dataset as illustrated in Table 1.

To conclude the finding of the discussion, irrespective of the dataset characteristics, most reviewed articles agree on these categories, which mean any extracted features will go under four main categories. The most important noticeable point is that some related works use only specific category such as static features to visualize security events while some others use a combination of any mentioned categories. It depends on the objective they intend to achieve. [5] Stated that hybrid features are the most comprehensive features since they involve vetting application installation file as well as analyzing the behavior of the application at runtime. However, the categorized based on the type of features as explained in literature as illustrated in Figure 2. [4] described only 10% of existing work was based on hybrid features, whereas 45% based on static features and 42% of relating works were based on dynamic features respectively and remaining 3% based on applications metadata feature.

Table 1
Description of the Dataset

Attributes	Description	Values/ Range
Dimension	How many dimensions do you have in the dataset?	1, 2, or 3 Dimension or more or Hierarchical
Primary data or data type	What type of data do you have?	Ordinal Interval Continuous Categorical Geographical
Tasks	What does the dataset describe?	Distribution Trends Relationship Comparison
Number of attributes and instances	How many numbers of column and rows do you have in the dataset?	Always presented in numerical values 1, 2, 3,….
Target or visualization	Which visualization tools could be used for such a dataset?	Histogram, pie chart, line chart, etc.

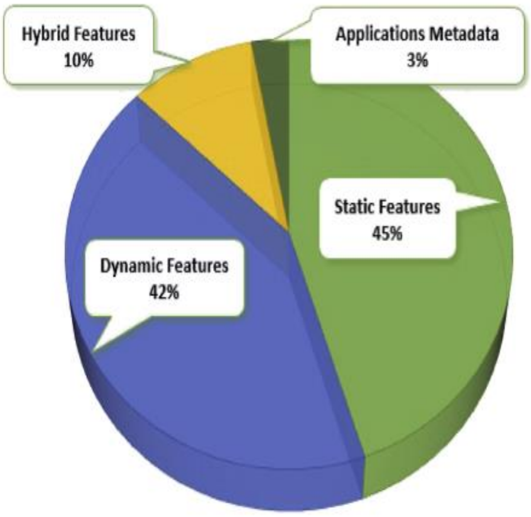

Figure 2: Recent statistical analysis based on type of features

A. The Most Extracted Features for Monitoring Security Events.

The success of visualization system depends on the extracted features. Malware features can be extracted from different resources. In most related work, the extracted features are classified as shown in Figure 3. Furthermore, Table 2 provides clear examples of some extracted features which are classified based on different type of sources.

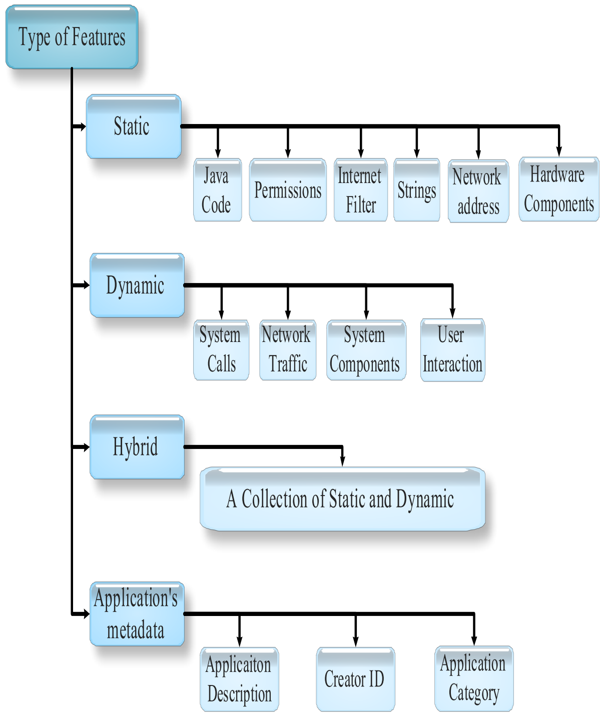

Figure 3: Most common extracted features for analyzing security events

Table 2
Examples of some Extracted Features Classified based on the Sources

Source	Features	References
Network traffic	Packets features: • Tcpdump: • Pcap: Timestamp • Ethernet: SRC MAC, DST MAC • IP: SRC IP, DST IP, type • TCP: SRC port, DST port, Flag • ICMP: type, code • UDP: SRC port, DST port	[19],[20], [21], [22],[23], [24]
CPU	CPU Sessions features: • Process ID • Running file name: Netscape, outlook, winword, explore, explorer, msaccess, powerpnt, excel, acrord32, winzip32 • cpuUser, cpuIdle, cpuSystem, cpuOther • iostat:Reports input/output statistics for CPUs and disks. • 1sof: Outputs a list of all open file descriptors and the processes using them.	[20],[25], [26], [27]
User profiling data or user interaction	API features: • Window titles: whatever is in the title bar of a window appearing on the desktop • The process table: the mechanism that multitasking operating systems use to keep track of the various applications running concurrently • Captures users' interaction with the device (e.g. pushing a button, zooming and navigating through pages).	[20], [28]
Wireless Sensor Networks	• Packet data Features: Message ID • Message type • Destination PAN ID: SRC ID	[29]
Memory	• Free memory • Used memory • memActive, and memMapped • vmstat: Reports memory statistics	[25], [26]
Bluetooth and Wi-Fi status	• WiFi on • WiFi off	[25],[27], [30]
API	• Sequence eventstop: Shows a list of running processes along with process statistics. Information such as memory utilization, runtime, process ID, parent process ID, and so on is shown for each process on the system.	[5]

IV. Conclusion

In summary, selecting an appropriate visualization tool plays a key role in descriptive, diagnostic, predictive and prescriptive analytics which is important to identify which features are more effective for certain visualization types. For instance, visualization tools for predictive task aim to predict the activities and behavior of specific software or for the behavior of a device (running applications). For this type of case, the visualization tool should provide understandable information about the malicious code activities. For example, using the line chart is more understandable than pie or column chart for describing the CPU performance. Whereas using pie or bar chart is more suitable for describing the available size of memory or disk. However, several visualization tools can be used to provide some clear results. Moreover, a visualization tool should be aimed at answering specific questions. Therefore, the visualization tools may incorporate one or multiple features with several visualization tools to visualize the results.

Acknowledgment

The authors would like to thank Universiti Teknologi Malaysia (UTM) and Cyber Security Malaysia (CSM) for supporting this work through the OTR Grant Scheme under Grant vote number R.K130000.7338.4B260.

References

[1] H. Shiravi, A. Shiravi, and A. A. Ghorbani, "A survey of visualization systems for network security", *IEEE Transactions on visualization and computer graphics*, vol. 18, pp. 1313-1329, Aug. 2012.

[2] A. Shabtai, D. Potashnik, Y. Fledel, R.Moskovitch and Y. Elovici, "Monitoring, analysis, and filtering system for purifying network traffic of known and unknown malicious content," *Security and Communication Networks*, vol. 4, 8, pp. 947-965, Aug. 2011.

[3] A.R., Mohd Faizal, A., Nor Badrul, S., Rosli, and F., Ahmad Firdaus, "The rise of malware," *Network and Computer Applications. J.*, vol. 75, pp. 58-76, Nov. 2016.

[4] M. Wagner, A.Rind, N.Thur and W. Aigner, "A knowledge-assisted visual malware analysis system: Design, validation, and reflection of KAMAS," *Computers & Security*, vol. 67, pp. 1-15, June. 2017.

[5] A. Feizollah, N. B. Anuar, R. SallehMarch 201, and A.W.A.Wahab, "A review on feature selection in mobile malware detection," *Digital Investigation*, vol. 13, pp. 22-37, June. 2015.

[6] J. Kim, and J. M. Youn, "Dynamic Analysis Bypassing Malware Detection Method Utilizing Malicious Behavior Visualization and Similarity," In *Conf. International Conference on Multimedia and Ubiquitous Engineering(MUE 2017)*, Seoul, 2017, pp. 560-565.

[7] Sophos Ltd, *SophosLabs 2018 Malware Forecast*. Oxford, UK: Abingdon Science Park, 2017.

[8] S. Z. M. Shaid and M.A. Maarof, "Malware behavior image for malware variant identification," in *Conf. International Symposium on Biometrics and Security Technologies (ISBAST)*, Kuala Lumpur, 2014, pp. 238-243.

[9] K. Han, J.H. Lim and E,G. Im, "Malware analysis method using visualization of binary files," in *Proc. Research in Adaptive and Convergent Systems*, Quebec, 2013, pp. 317-321.

[10] V. Sitalakshmi, and A. Mamoun, " Classification of Malware Using Visualisation of Similarity Matrices," In *Conf. 2017 Cybersecurity and Cyberforensics Conference (CCC)*, London, 2017, pp.3-8.

[11] P. Burnap, R. French, F. Turner and K. Jones, "Malware classification using self organising feature maps and machine activity data," Computers & Security, vol. 73, pp. 399-410, March. 2018.

[12] Z. U. Rehman, N.K Sidra, M. Khan, J. W. Lee, M. "Machine learning-assisted signature and heuristic-based detection of *malwares* in Android devices," *Computers and Electrical Engineering*.,

[13] Z. Zongqu, W. Junfeng and B. Jinrong, "Malware detection method based on the control-flow construct feature of software," *IET Information Security*, vol.8, pp. 18-24, 2014.

[14] T. Dube, R. Raines, G. Peterson, K. Bauer, M. Grimaila and S. Rogers, "Malware target recognition via static heuristics," *Computers & Security. J.*, vol.31, pp. 137-147, Feb. 2012.

[15] T. Ronghua, B. Lynn, I. Rafiqul and V. Steve, "An automated classification system based on the strings of trojan and virus families," in *Conf. 4th International Conference on Malicious and Unwanted Software (MALWARE)*, Quebec, 2009, pp. 23-30.

[16] S. Asaf, M. Robert, E. Yuval and G. Chanan, "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey," *Information Security Technical Report.*, vol.14, pp.16-29, Feb. 2009.

[17] R. Andre, A. Gregio, D. Rafael, C. Santos, "Visualization techniques for malware behavior analysis," SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 801905-9, 2011.In *Proc. Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense*, Florida, 2011, pp. 1-9.

[18] T. Muhammad and Z. Halim, "Employing artificial neural networks for constructing metadata-based model to automatically select an appropriate data visualization technique," *Applied Soft Computing*, vol.49, pp. 365-384, Dec. 2016.

[19] K. Abdullah, C. Lee, G. Conti, and J.A. Copeland, "Visualizing network data for intrusion detection," In *Proc. Sixth Annual IEEE SMC Information Assurance Workshop*, New York, 2005. pp. 100-108.

[20] T. Goldring, " Plots for visualizing user profiling data and network traffic," *In Proc. 2004 ACM workshop on Visualization and data mining for computer security*, Washington DC, 2004, pp.119-123.

[21] E. Corchado, and A. Herrero, "Neural visualization of network traffic data for intrusion detection," *Applied Soft Computing*, vol.11, pp. 2042-2056, March. 2011.

[22] A. Herrero, E. Corchado, M. A. Pellicer, and A. Abraham, "A. MOVIH-IDS: A mobile-visualization hybrid intrusion detection system," *Neurocomputing*, vol. 72, pp. 2775-2784, Aug. 2009.

[23] J. Pearlman, and P. Rheingans, "*Visualizing network security events using compound glyphs from a service-oriented perspective*,". Berlin, Heidelberg: Springer, 2007, pp. 131-146.

[24] F. Mansman, L. Meier and D. A. Keim, "*Visualization of host behavior for network security*," Berlin, Heidelberg: Springer, 2008, pp. 187-202.

[25] D. Gianluca, M. Fabio, S. Andrea and S. Daniele, "MADAM: a multi-level anomaly detector for android malware," In *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*, St. Petersburg, 2012, pp.240-253.

[26] B. Amos, H. Turner, and J. White, "Applying machine learning classifiers to dynamic android malware detection at scale," In *Conf. 9th International wireless communications and mobile computing conference (IWCMC)*, Sardinia, 2013, pp.1666-1671.

[27] H.S. Ham, and M.J. Choi, "Analysis of android malware detection performance using machine learning classifiers," In *Conf. 2013 International Conference on ICT Convergence (ICTC)*, Jeju, 2013, pp. 490-495.

[28] A. Gianazza, A, F. Maggi, A. Fattori, L. Cavallaro, and S. Zanero, "Puppetdroid: A user-centric ui exerciser for automatic dynamic analysis of similar android applications," *arXiv preprint arXiv*, vol. 1402.4826, Feb. 2014.

[29] S. Ravichandran, R.K. Chandrasekar, A. Selcuk Uluagac, and R. Beyah, "A simple visualization and programming framework for wireless sensor networks: PROVIZ," *Ad Hoc Networks*, vol. 53, pp. 1-16, Dec. 2016.

[30] V. Hoffmann, J., Neumann, S., & Holz, T, "Mobile malware detection based on energy fingerprints- A dead end," In I*nternational Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*, Saint Lucia, 2013, pp.348-368.