MINAR International Journal of Applied Sciences and Technology
ISSN: 2717-8234
Article type: Research Article

Received: 06/01/2022 **Accepted:** 20/02/2022 **Published:** 01/03/2022

THE EFFECT OF MALWARE'S APIS RELATIONS ON SOFTWARE SECURITY DESIGN

Karam H. THANOON[1]
University of Mosul, Iraq

Basim MAHMOOD[2]
University of Mosul, Iraq

Marwah M. A. DABDAWB[3]
University of Mosul, Iraq

Abstract

Recent years have witnessed a great revolution in web technologies and their applications. Most of these applications are connected to the Internet. One of the most frequent issues in these applications is the security issue. Malware is the main reason behind this issue since they harm users in many different aspects such as damaging files, stealing credentials, operating system malfunctioning, etc. Therefore, many companies around the world develop antiviruses software aiming to mitigate the security issue. Most of the known viruses can access users' computers or web accounts through some APIs. Therefore, antivirus companies try to update the API databases of their software periodically. This paper suggests a method for investigating the relations among different kinds of malware in terms of the API they used. Then, it provides recommendations about this malware and its APIs. The method followed in this work is based on concepts inspired by network science. The malware and its APIs are modeled as a network with nodes and edges. The results show interesting facts about the investigated malware that are of interest for software security architects and give the relations between various malware which call the same API function, depending on that malicious software behavior can be detected by antivirus or anti-malware engine.

Keywords: *cyber security, Malware analysis, Software Security, APIs call, Network Science*.

http://dx.doi.org/10.47832/2717-8234.10.14

[1] karamhatim@uomosul.edu.iq, https://orcid.org/0000-0002-7224-1474
[2] bmahmood@uomosul.edu.iq, https://orcid.org/0000-0001-5070-1882
[3] marwa_marwan21@uomosul.edu.iq, https://orcid.org/0000-0003-0981-277X

Introduction

1.1 Overview

Cyber security includes one of the most important topics, which is software security against different types of malware. The hacking was started as a kind of fun in victim machines but now with the great revolution in the Internet services and smart devices propagation, hackers move to finical profits or valuable assets. Currently, hacking has become popular and has become a phenomenon. All malware is based on API system call for windows functions, therefore, if analysts can recognize the API function and the executable file call, it may enable to expect whether this file was benign or malicious. On the other hand, hackers work hard to add ambiguity to their malware by adding some unused and non-useful API system calls that made malware analysis process more difficult due to the fact that all antivirus software depends on malware analysis to update their databases frequently for all users who installed these antiviruses on their computers (D'Angelo et al. 2021).

On the other hand, network science is a multidisciplinary field and one of the modernist methods for investigating and analyzing complex phenomena. It formalizes a problem as a Graph (V, E) with V nodes and E edges connecting them. This approach enables researchers to dig deeply into the relations among data objects. This is important since the traditional analysis may not be able to show the hidden knowledge within datasets. Network science has become popular in many different fields and is used to create networks for a variety of applications such as biological networks (Hammadi et al. 2021), citation networks (Mohammed et al. 2020), road networks (Hasan and Mahmood 2021), power-grid networks (Sun 2005), social networks (Mahmood et al., 2018), Software Security (Younis and Mahmood 2020, Mahmood 2021), to mention a few.

According to the aforementioned, this work uses concepts inspired by complex networks in the modeling and analysis of this work.

1.2 Literature Review

In 2021, Kim et al. proposed work for expression malware characteristics depending on API system call sequence to detect and classify different types of malware. DLL Injection, Downloader, Key Logger, and Anti debugging were the four malicious behaviors expressed. Their work gives a high-efficiency rate in detection using static analysis. Another study performed in (D'Angelo et al. 2021) presented a malware classification method using API call sequence. They found that the process becomes more complex when malware developers begin to add useless API system calls to evade the detection process and affect the malware analysis process. Also, the results obtained were better than the results achieved from the Markov chain.

The authors in (Schofield et al. 2021) proposed malware classification depending on API system call sequence. Machine learning techniques were used to detect and classify new malware that were considered traditional methods and fail to manipulate with. The researchers used a one-dimension conventional neural network with 5385 samples as a dataset. Their results gave a high accuracy rate with 3D visualization. This helped humans to understand API call stream construction. The study of (Han et al. 2019) presents a detection tool based on correlation with static and dynamic malware characteristics. Their approach focused on the correlation between static and dynamic API call sequences, which was more important to gain strong results. The proposed approach provides a high accuracy rate in performing detection and classification.

Finally, the work of (Gupta et al. 2018) used API call sequence to characterize malware by capturing the behavior of malicious software. The authors analyzed five classes of malware using 2000 samples. The fuzzy hash algorithm was used to generate malware signatures. The proposed algorithm provided efficient performance for malware classification that was based on API calls.

1.3 Research Problem and Contribution

According to the literature, there is a lack in suggesting methods that study the relations among different kinds of viruses through APIs. Hence, the contribution of this work is in providing a network model that is able to enrich security architects with useful knowledge about viruses and API. The model can also be used to provide knowledge about the further dangerous APIs considering different types of viruses.

The rest of this paper is organized as follows: Section 2 describes the research method in terms of the data collection process, network creation, and network measurements used in this work. Section 3 presents the visualization and analysis of the proposed model as well as discussions about the model. Finally, this work is concluded in Section 4.

2. Research Method

2.1 Dataset Collection

The data set was collected in two ways then merged to perform malware datasets depending on the API system call that they called. The first way in dataset collection done by downloading multi malware family with 10 samples for each family, then moved to the malware analysis stage where each malware sample was analyzed using a tool called (pestudio) which has the ability to analyze malware with different families and gives various information about each one, one of the most important information gained was all API call for this malware which stored temporarily in a text file format named as same as malware sample name, this step was repeated until all samples analyzed using this manner, the last step in dataset collection was to create a dataset file using Microsoft excel, his rows represent malware samples and columns represent API call functions name, each malware call API name will be marked with 1, therefore, if it doesn't call this API will be marked with 0. All these steps are done in an automated manner using MATLAB programming language. The second way was by searching and downloading malware datasets from a website with many samples and performing some preprocessing steps on it to be ready for usage. Finally, two files were merged and constructed our malware dataset which visualization will be done on it.

2.2 Network Creation

After completing and verifying the dataset, the output files were used to create the network, and perform the visualization and analysis processes. The first file (node file) was used to create network nodes (malware). The second file was used to create the edges among nodes (APIs). The strategy of creating an edge between two malware (viruses) is that when two viruses have APIs in common, an edge is created between them. If more than one API is associated between two malware, the weight of the edge is increased by one. Therefore, the more APIs in common between two malware, the more weight gained between both.

Now, Gephi software was used to visualize the network. After that, network measurements (as discussed in the next section) were calculated aiming to evaluate the network. Figure 1 depicts the basic visualization of the malware network. The visualization shows the 9 types of malware that are included in the dataset. Each type is encoded with different node color. Also, the edges among the nodes are dense representing the APIs that are associated with different types of malware.

Moreover, the visualization shows several edges with high weight values. This case reflects that some malware has several APIs in common with other malware types. In-network point of view, this phenomenon reflects some facts that will be discussed later in the results section.

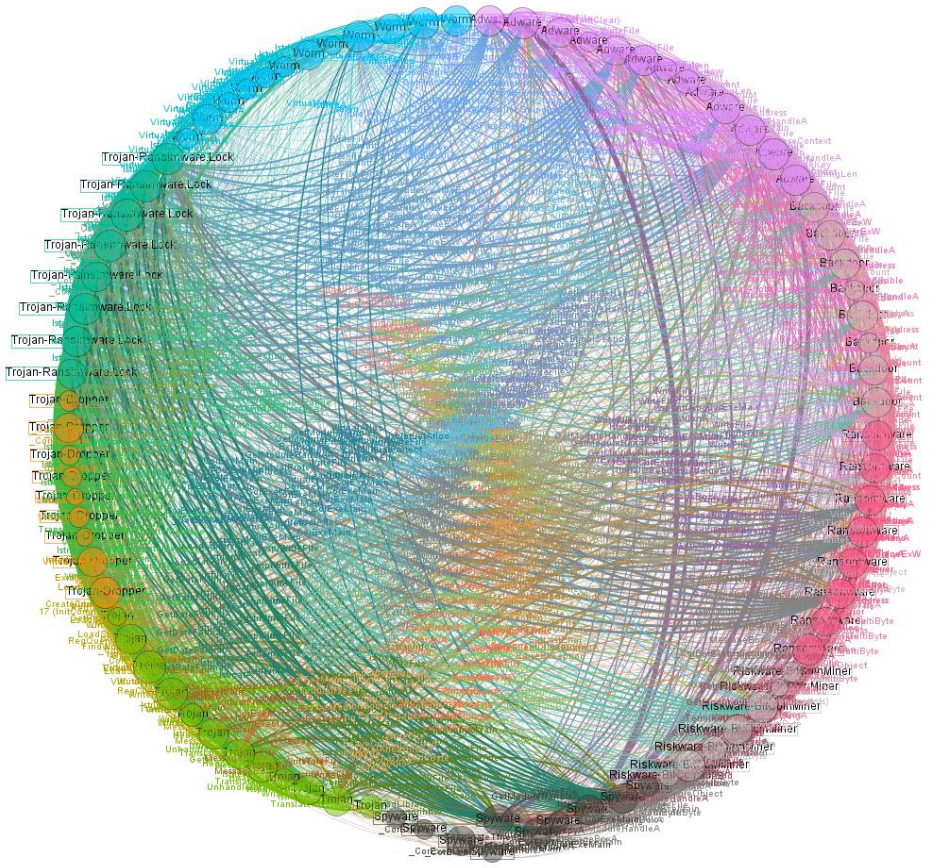

Figure 1: Basic visualization of the malware network.

2.3 Network Measurements

In this work, many network measurements were used to evaluate the network and its nodes as follows:

Network-level measurements: These measurements were utilized to evaluate the whole network structure. The *Diameter* of a network reflects the farthest distance between network pairs. It enables to measure the distances among network nodes. The *Degree Distribution* was also used to measure how the degree of network nodes are distributed. *The average clustering coefficient*, this measurement reflects the average tendency of networks nodes to cluster in groups. *Density* reflects the ratio of the number of actual edges to the number of potential edges within the network. *Average path length*, this metric shows the average length between network pairs. Moreover, a *community* in a network represents a group of nodes that are highly connected and have features in common.

Nodes Level measurements: This kind of measurement reflects facts about network nodes. *Node degree* shows the number of connections that a node has to other network nodes. *Betweenness centrality* reflects how well-positioned a node is in network structure. In other words, it is the number of times that a node appears in the shortest paths of network pairs. *Edge weight* represents how well-connected two nodes are, it also reflects the strength of a relation between two nodes in the network.

3. Results and Discussions

3.1 Results and Network Structural Features

This section presents the obtained results of the malware network generated in this work. Table 1 presents the characteristics of the network shown in Figure 1. According to the table, the network shows noticeable characteristics. The average degree is considered extremely high because most of the malware types have a lot of APIs in common, which leads to having a high average degree of nodes in the network. This fact is also reflected in

the network when observing the density and the average clustering coefficient. Moreover, the diameter and the average path length reflect the low distances between network nodes that are due to the high average degree. Furthermore, based on the basic visualization shown in Figure 1, the network shows 9 different communities representing the 9 malware types in the dataset. Each community is encoded in a particular color.

Table 1: Characteristics of the basic malware network.

Nodes	Edges	Average Degree	Diameter	Density	Average Clustering Coefficient	Average Path Length
87	1895	43.563	3	0.507	0.952	1.233

According to the aforementioned indicators, the structural features of the network show that the relations among different/same malware types are strong. This led to moving to the next step, which is investigating the most effective edges within the network. However, most of the malware in the network is connected, which is clear when observing the measurements presented in Table 1. Therefore, we decided to filter the weights among the malware. Before, performing the edges filtering process, it was needed to extract the distribution of the edges in the malware network aiming to accurately filter the edges and extract the most influential ones. Figure 2 shows the weight values of network edges. It can be observed that the values of the weights are ranged between 1.0 to 192.0. According to the figure, it can also be observed that the weights follow a power-law distribution. This distribution has an important feature, which is the "*Pareto Rule*". It states that approximately 80% of the observations come from the effect of the other 20%. The rule also reflected the "*Elite Theory*" in sociology that states; some people in a community hold the most relationship power and affect the majority of that community. Based on that, the elite weights in the malware network were extracted. The strategy of extracting the elite weights is based on the Pareto rule, which means the malware network keeps the highest 20% of the weights (elite weights) and discards the other weights. The elite weights reflect the strongest relationships in the network. Figure 3 demonstrates the malware network after using only the elite weights.

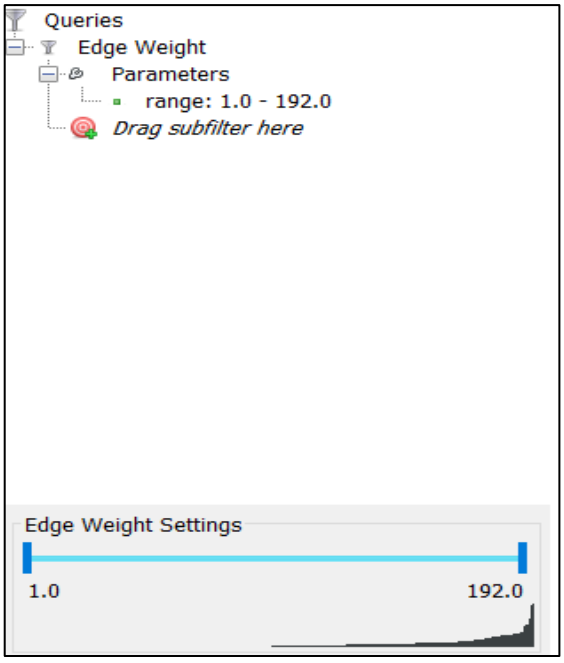

Figure 2: Malware network edge weight settings.

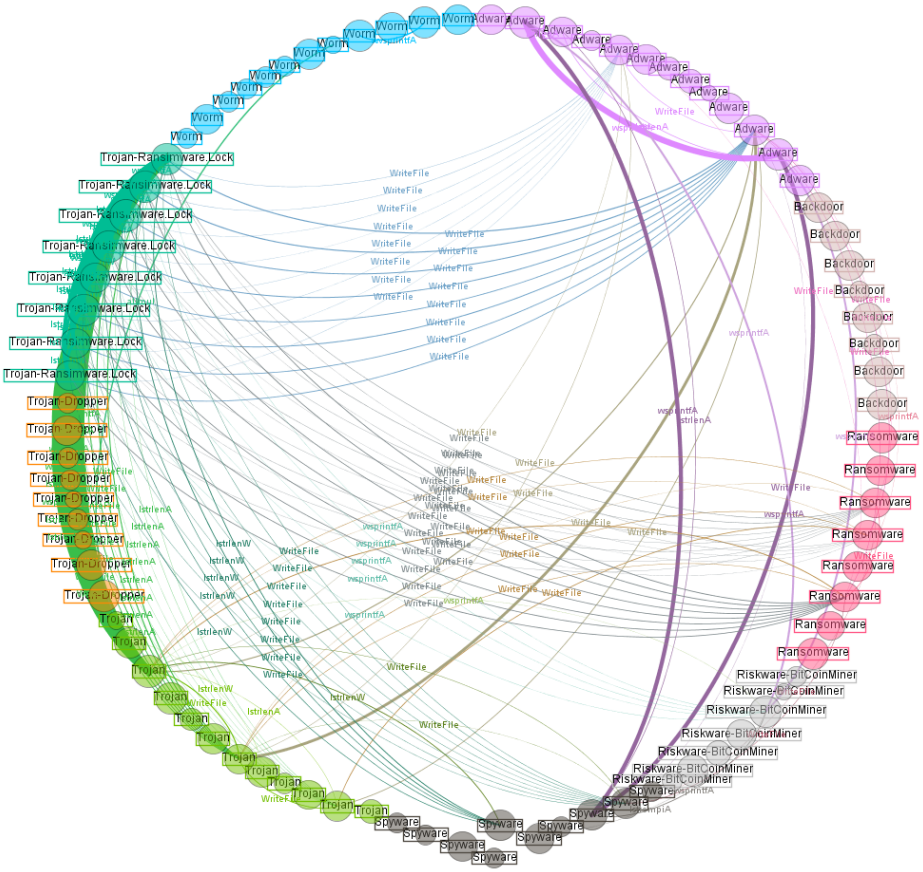

Figure 3: Malware network with only the Elite weights.

The characteristics of the malware network with elite weights are presented in Table 2. It is clear that these characteristics are different from the basic malware network, which is due to eliminating 80% of the lowest weights. It can be noticed that the average degree, density, average clustering coefficient, and average path length were significantly reduced. However, the diameter was slightly decreased due to performing the weight filtering process on the network. Moreover, the number of edges was noticeably decreased due to the same previous reason. It is important to mention that the number of edges does not relate to the weights because the weight represents the strength of relation, while the edges are an expression of whether a relation exists between two nodes.

Table 2: Characteristics of the highest-weight-based malware network.

Nodes	Edges	Average Degree	Diameter	Density	Average Clustering Coefficient	Average Path Length
87	149	3.425	4	0.04	0.878	1.661

Another investigation and analysis were performed using the other network measurements. Table 3 ranked the risk level of malware types in the network based on the betweenness centrality measurements. It should be mentioned this investigation was performed using the filtered network. The betweenness centrality was averaged for all the subtypes of the malware types in the filtered network.

Table 3: Ranking malware types according to their level of betweenness centrality.

RANK	MALWARE TYPE	AVERAGE BETWEENNESS	AVERAGE DEGREE
1ST	Riskware-BitcoinMiner	0.01961	62.0
2ND	Trojan.Ransomware.lock	0.01335	61.0
3RD	Spyware	0.00179	58.0
4TH	Adware	0.00124	60.0
5TH	Trojan	0.00122	58.3
6TH	Ransomeware	0.00004	56.0
7TH	Backdoor	0.00000	56.0
8TH	Trojan Dropper	0.00000	57.3
9TH	Worm	0.00000	57.3

The visualization of the filtered malware network where node size reflects the level of betweenness centrality is shown in Figure 4. In this figure, the types of malware are encoded in color nodes. Also, the figure shows how strong the relation (weight) is between two different malware. The strength of the relation (more APIs in common) is encoded by the thickness of the edge.

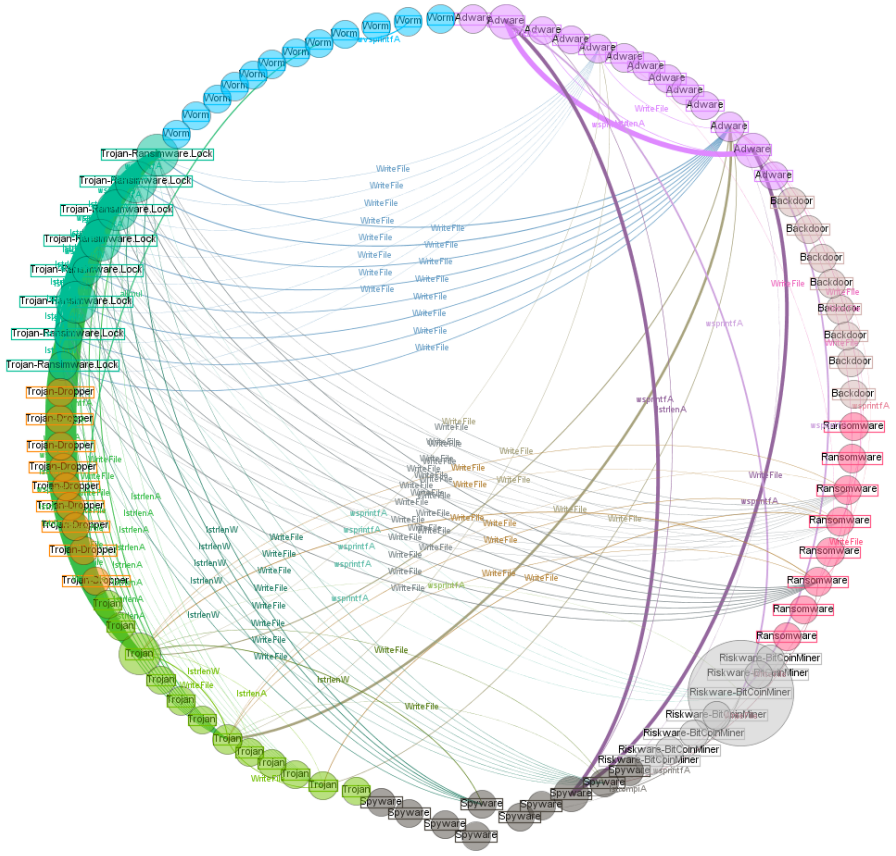

Figure 4: Betweenness-based visualization of the filtered malware network.

3.2 Discussions and Security-Related Analysis

Based on Figures 3 and 4, the malware uses APIs system calls such as *write file*, *istrlena*, *istrlenw*, and *wsprinf*. In particular, the analysis process for malware and extracting API system call for visualization gives an efficient result about the most API call used by malware with all connections between malware families that use this function call. Depending on that and because all the dataset analysis was malicious software and doesn't contain benign software, all edges in the generated network represent the most important

system call by all malware in the dataset. Hence, if antivirus software finds any program that calls these most used APIs function, it can be immediately considered malicious software. From another point of view, this work can evade the situation when hackers add some API to their software in order to add some ambiguity to the structure of the malware. Therefore, the use of complex networks concepts for visualizing the network based on the betweenness and degree centralities of nodes was efficient to introduce a new method for malware analysis.

According to Table 3, malware types or families were sorted according to their level of betweenness centrality, this table ensures the fact that most malicious software was used symmetric or sometimes the same API system call which led to higher betweenness centrality, all above results verified that as example Ransomware nowadays used more than adware.

4. Conclusions

Static malware analysis with dynamic malware analysis still extracts information from malware and creates a dataset that all antiviruses depend on it. However, the false-positive ratio with false-negative ratio is still not small and there exists malware that works on victims' computers without their knowledge because some antiviruses cannot recognize it using the information gained from malware analysis. In this work, a malware network was created representing all relations between API system calls, which are called by various malware families. This work can be considered as a starting point to analyzing malware using network science concepts. The results of this work show the efficiency of network centrality measurements in detecting some important phenomena during malware analysis. This work can be extended to include more data aiming to have a dataset of the most influential malware that security architects should give more attention to. This suggestion is kept for future works.

References

D'Angelo, G., Ficco, M., & Palmieri, F. (2021). Association rule-based malware classification using common subsequences of API calls. *Applied Soft Computing*, *105*, 107234.

Hammadi, D. S., Mahmood, B., & Dabdawb, M. M. (2021). Approaches on Modelling Genes Interactions: A Review. *Technium BioChemMed*, *2*(4), 38-52.

HASAN, A. S., & MAHMOOD, B. (2021). MINAR International Journal of Applied Sciences and Technology.

Gupta, S., Sharma, H., & Kaur, S. (2016, December). Malware characterization using windows API call sequences. In *International Conference on Security, Privacy, and Applied Cryptography Engineering* (pp. 271-280). Springer, Cham.

Han, W., Xue, J., Wang, Y., Huang, L., Kong, Z., & Mao, L. (2019). MalDAE: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics. *Computers & Security*, *83*, 208-233.

Kim, J., Lee, S., & Youn, J. (2021). Expression of malware characteristics using API sequence. *Journal of Smart Technology Applications*, *2*(1), 1-8.

Mahmood, B., Younis, Z., & Hadeed, W. (2018). Analyzing Iraqi Social Settings After ISIS: Individual Interactions in Social Networks. *American Behavioral Scientist*, *62*(3), 300-319.

Mahmood, B. (2021). Prioritizing CWE/SANS and OWASP Vulnerabilities: A Network-Based Model. *International Journal of Computing and Digital Systems*, *10*(1), 361-372.

Sun, K. (2005, August). Complex networks theory: A new method of research in power grid. In *2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific* (pp. 1-6). IEEE.

Schofield, M., Alicioglu, G., Binaco, R., Turner, P., Thatcher, C., Lam, A., & Sun, B. (2021, January). Convolutional Neural Network for Malware Classification Based on API Call Sequence. In *proceedings of 2021 the 14th International Conference on Network Security & Applications. Computer Science & Information Technology (CS & IT). Zurich, Switzerland*.

Younis, Z. K., & Mahmood, B. (2020, February). Towards the Impact of Security Vunnerabilities in Software Design: A Complex Network-Based Approach. In *2020 6th International Engineering Conference "Sustainable Technology and Development"(IEC)* (pp. 157-162). IEEE.