Ransomware Behavior Attack Construction via Graph Theory Approach

Muhammad Safwan Rosli[1], Raihana Syahirah Abdullah[2]*
Warusia Yassin[3], Faizal M.A[4], Wan Nur Fatihah Wan Mohd Zaki[5]
Centre of Advanced Computing Technology, Fakulti Teknologi Maklumat dan Komunikasi,
Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia[1, 2, 3, 4, 5]

***Abstract*—Ransomware has becoming a current trend of cyberattack where its reputation among malware that cause a massive amount recovery in terms of cost and time for ransomware victims. Previous studies and solutions have showed that when it comes to malware detection, malware behavior need to be prioritized and analyzed in order to recognize malware attack pattern. Although the current state-of-art solutions and frameworks used dynamic analysis approach such as machine learning that provide more impact rather than static approach, but there is not any approachable way in representing the analysis especially a detection that relies on malware behavior. Therefore, this paper proposed a graph theory approach which is analysis of the ransomware behavior that can be visualized into graph-based pattern. An experiment has been conducted with ten ransomware samples for malware analysis and verified using VirusTotal. Then, file system among features were selected in the experiment as a medium to understand the behavior of ransomware using data capturing tools. After that, the result of the analysis was visualized in a graph pattern based on Neo4j which is graph database tool. By using graph as a base, the discussion has been made to recognize each type of ransomware that acts differently in the file system and analyze which node that have the most impact during analysis part.**

***Keywords*—Ransomware; behavior analysis; graph theory; file activity system; Neo4j**

I. Introduction

Information security is one of the critical issues that has been addressed in order to maintain the operation of the system constantly [1]. Today, cybercriminal tend to target vulnerable users and communities such as company of business, government sectors and critical infrastructure for example healthcare. The attacks can cause high severity and impact in most cases that even small fraction of time influencing detection and prevention need to be very concern and critical. With the intrusion and attacks, the attackers can gain access of confidential data from the victims or injecting various malware inside victim's machine [2]. With new challenges such as sophisticated malware that has been rampaging in our network, traditional conventional solution like signature-based detection that relies on malware attack pattern does not give higher impact and less efficient in preventing malware attacks [3].

Thus, a few solutions, techniques and approaches have been developed using sandboxes with features that capable to filter and distinguish between benign and malicious files. However, as the solutions which used multiple of sandboxes with virtual machines grow larger, they also consume huge amount of resources such as RAM, machine storage which are time consuming [3]. So, to mitigate the concern issues, researchers need to come with different approaches and solutions to defend against current and future threats and also to understand the behavior of the malware attacks and their interactions with victim's machine [4].

The main problem remain persists yet and it still needs to keep on update where the researchers need to understand the malware behavior whether it is in network traffic or file activity system in the form of statistical and dynamic. This research also stressed out the problem in visualizing malware behavior since the data can be represented in an easy way to be understand such as in the form of graph instead a typical data form such as comma-separated value (CSV). Therefore, the main objective of this research is to study multiple sets of ransomware that will be selected into testing environment. With the result of the data from the experiment, the data were translated and visualized into graph form by using graph database tools in assisting the research development.

In summary, this paper makes the following contributions:

- The research shows an alternative approach and analysis behavior of ransomware by constructing several ransomware samples using file system activity as feature selection.
- The research analyses the ransomware using Process Monitor to capture data log and classifies behavior of ransomware as the log produces multiple attack vector of ransomware.

II. Ransomware at Glance

When it comes to an attack that causes colossal impact, ransomware is one of the malwares that shows high severity in cybersecurity threats. An individual as well as big corporations that heavily depend on network would be facing this risk and need to come up with mitigation strategies. Currently, the popular use of machine learning in many sectors has inspired malware researcher to use the approach as ransomware detection system that helps to increase detection rates [5]. Furthermore, the increases of attacks from new ransomware families shows that attackers improving themselves with numerous cunning and sophisticated features such as encryption mechanisms or propagation of worm [6]. Hence, the motivation of this research is to study anomaly behavior of ransomware and analyze the behavior based on ransomware distinct features. Generally, ransomware will act aggressively

*Corresponding Author.

by starting to encrypt the files of victim personal computer and then delivering a ransom note with a set of instructions for payment usually by using popular cryptocurrency which is Bitcoin [7].

Like other malwares, ransomware also has its own lifecycle that can be seen in Fig. 1. Several actions are needed to make ransomware attack successful when it infects the computer [8]. The chronological attack starts when the victim downloading suspicious link in email attachment or accidently drive-by download from suspicious website. This will lead into next steps which is the victim executes ransomware file since typical ransomware are executable file format. When the ransomware has been executed, the malware will try to establish the connection in Command & Control (C&C) so that the attacker can gain encryption key of victim to bargain with the victim.

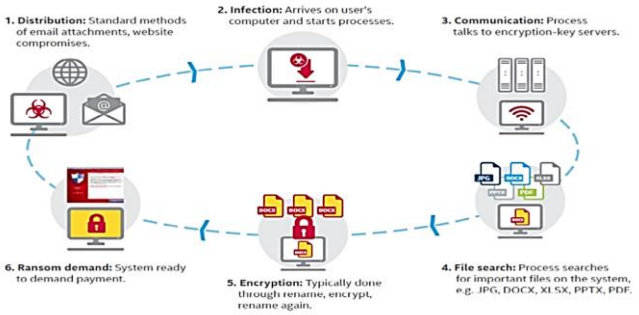
1. Distribution: Standard methods of email attachments, website compromises.
2. Infection: Arrives on user's computer and starts processes.
3. Communication: Process talks to encryption-key servers.
JPG
DOCX
XLSX
PPTX
PDF
DOCX
4. File search: Process searches for important files on the system, e.g. JPG, DOCX, XLSX, PPTX, PDF.
DOCX
DOCX
DOCX
5. Encryption: Typically done through rename, encrypt, rename again.
6. Ransom demand: System ready to demand payment.

Fig. 1. Ransomware Lifecycle [4].

Next, the ransomware will search for related file with specific extension such as pdf, docx, xlsx, pptx, and jpg. Then, encryption will be done by renaming the file, encrypting the file, and then renaming it again. This steps will show most of the encrypted file with unique extension based on the type of ransomware. After the file in the directory has been encrypted, the ransomware will display a ransom note including the instruction and step by step to pay the ransom, mostly using Bitcoin transaction in The Onion Ring (TOR) protocol.

The author in [9] has stated three Indicators of Compromise or IoCs. These IoCs has been analyzed based on the result of ransomware behavior detection. The first indicator that has been identified is file changes in file system. By encrypting the files, ransomware changes the property of the files such as changing file extension and name of files. A second indicator which is file entropy can observe the randomness of file in file system. This is important for ransomware behavior detection since the encryption of ransomware causes high entropy which triggers the detection threshold and the system will detect it is as an attack. The third indicator is canary files which is a fake file that is implemented with real files. These files can set an alarm for a system if ransomware tries to encrypt the files thus, an early detection can be achieved.

The author in [10] classified ransomware as polymorphic and metamorphic also predicted threats for future ransomware that will be used by the attacker such as polymorphic blending of traffic or sandbox evasion technique. Therefore, most researchers are now focusing on Machine Learning (ML) techniques since it is capable to analyze ransomware behavior pattern thoroughly compared to other static analysis which depends on signature-based pattern in ransomware detection [11]. With constant changing of ransomware behavior, the researchers need to be alerted and further improved for current solution or framework.

III. Definition of Graph

A graph can be denoted as G = (V, E) which consists of vertices, V and edges, E. Element of vertices in graph is called nodes where entities, variable have properties, where element edges are called relation or connectivity. Vertices, V and Edges, E can be denoted as $V = \{v_i\}$ and $E = \{k_i\}$ such i = 1,2…,n and k = 1,2,…,m respectively. The value of k can be referred to the edge between v_i and v_j. There are multiple types of graphs depending on the degree of nodes and edges in graph structures such as undirected graph, directed graph, mixed graph, multigraph and weighted graph.

The basic of undirected graph is the two nodes which are connected to each other with identical edge such edge (a, b) is equal to edge (b, a). The total number of edges in this graph can be denoted as $n(n\text{-}1)/2$ without a loop in the graph [12]. A directed graph consists of nodes and the edges which are identical and can be denoted as G = (V, E). These nodes in the graph can be a set of multiple paired with edges in a form of line or arrow [12]. Three elements which can be seen in a directed graph are directed edge, in-degree and out-degree.

In addition, mixed graph consist of directed and undirected edges can be denoted as G = (V, E, A). Usually, ordered pair and unordered pair is called s arc and edge of a graph respectively [12]. Naturally, a multigraph is undirected type of graph with multiples edges. Consequently, the multiple edges also can be connected to multiple vertices or nodes that cause a loop or cycle. Lastly, a weighted graph will have weight assigned on its edges which represent quantities such as time, distance, force or monetary values. Network graph is one of the examples that can be referred to weight a graph as the graph contains measuring cost to an edge of network [12]. Unweighted graph can also be referred to a graph without weightage [12].

IV. Graph Theory

Graph theory is categorized as discrete mathematics in field of mathematics and offers visual representation using graph based on the given networks. Most graph theory combine with other analytical tools, several algorithms or frameworks in order to represent the analysis that has been done. Basically, a graph provides illustrative design to show relationship among the entities [12]. These entities are called node whereas vertices are relationship between the nodes. Multiple structures of graph are possible given information data in order to identify which graph has the most influential based on ranking of nodes. Most of graph theory analysis starts with graph key elements in order to understand the graph itself.

One of the key elements is cardinality that refers to the size of the set or number of elements in the set. In graph theory, vertex cardinality refers to the size or numbers of nodes or vertices. Thus, cardinality of the nodes denoted as $n = |V|$, where V refers to the number of nodes in the graph. Whereas, cardinality of the edges denoted as $m = |E|$, where E refers to the number of edges in the graph. Furthermore, centrality identifies the degree, distance and vertices of the node, then they rank its importance in the network [13].

Additionally, adjacency matrix is the edge between two nodes that are adjacent where the matrix represents adjacency relationship. Adjacency matrix present as A = $[a_{ij}]$ such i,j = 1,2,..n is define as square 0-1 matrix of size n x m which consists of binary value of 1 and 0 where [14]:

$$a_{ij} = \begin{cases}1 \\ 0\end{cases}, if\ nodes\ v_i\ and\ v_j\ are\ adjacent \qquad (1)$$

Besides, incident matrix is a relationship between the nodes and edges in graph that can be define as $B = [b_{ij}]$ such i = 1,2,..n and j = 1,2,..m of size n x m where [15]:

$$a_{ij} = \begin{cases}1 \\ 0\end{cases}, if\ nodes\ v_i\ and\ v_j\ are\ adjacent \qquad (2)$$

Degree of matrix also refers maximum values of each nodes in the graph is denote as $D = [d_{ij}]$ such i = 1,2,…n and j = 1,2,…m where [16]:

$$d_{ij} = \sum_{j=1}^{n} a_{ij} \qquad (3)$$

Finally, clique is structure of a graph that consists of nodes with characteristics and has strong connection to each other's. Maximum clique is based on the number of nodes will denote as $\omega(G)$.

By using graph analysis, the interactive data analytics will become more flexible and easier to be used since the graph represent node and edges can be spotted [17]. Neo4j is a graph database that is highly praised as front-end and back-end media where social graphs represent real information. In addition, graph database also offers graph analytics, allowing method such as prediction and minimum spanning tree to become more popular in graph-based approach [18].

Graphs can represent network state transitions leading to attack goals, attacker exploitation steps related by preconditions and post conditions, intrusion alert sequences, logical dependencies for attack goals, or host attack reachability. Attack graphs have also been implemented with the relational model [19]. It also can become a useful tool for security and risk analysis since it can represent relationship between multiple node as shown in Fig. 2. However, the process of graph analysis can become tedious especially if the data is big and having a large node [20].

Computer network also can be represented by the graph analysis by defining entities such as IP address, hostnames to nodes and activities such as connection between the nodes to edges [21]. Since there are numbers of demand in graph analysis, three factors need to be considered which are graph data model, memory management and the algorithm of graph analytics [22].

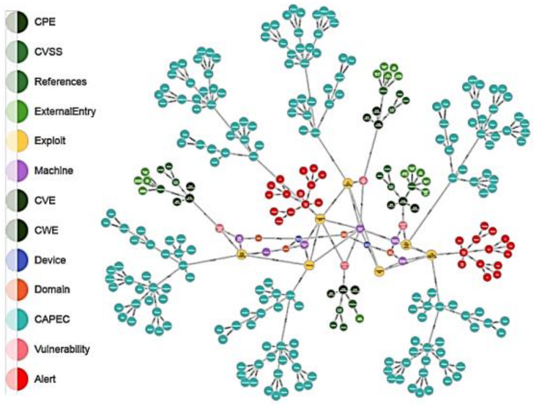

Fig. 2. Graph Model using Graph Database Tool [20].

V. Related Works

One of the disadvantages of static analysis approach is difficulty in extracting information with code obfuscation techniques. A file such as binary is also difficult to dissemble from time to time. Hence, dynamic analysis approach provides controlled environment which runtime information can be captured for analysis, since it does not affect extraction from complex technique [23]. Graph theory is a dynamic analysis approach, which helps in studying the relationship between the entities including the distance of each nodes and degree of nodes within the networks [24]. Furthermore, graph-based approach for detecting malware is an active area of research for malware targeted at PCs as well as mobiles [25]. Thus, a brief set of previous works that have been done by previous researchers, which applied graph theory approach in malware behavior, detection and graph techniques.

In malware detection, [1] proposed graph-based algorithmic technique using System-call Dependency Graphs (ScDG). The system calls that have been generated in ScDG will form a set of graphs called Group Relation Graphs (GrG). Based on the degree and vertices generated by the graphs, the author was able to investigate malware behavior in each system call in graph and enhance the accuracy of detection in their detection model.

To prevent malicious attacks in IoT devices, [26] has proposed behavior-based deep learning framework (BDLF), utilizing Stacked AutoEncoders (SAEs) and machine learning algorithm to obtain high level representation of malware behavior graphs. The framework uses Control Flow Graph (CFG) generated from the malware samples to analyze degree centrality, the size of graph, radius and shortest path of each node. This also includes the investigation of malware behavior similarities and the differences between IoT malware and android malware. Another framework called Together proposed by [27] is capable to generate massive graph provided by android malware samples and network files such as IP addresses and domains. The framework uses multiple heterogeneous graphs for network information to correlate each sub-threat network using Page Ranking algorithm to label malicious nodes in graph.

In addition, machine language instruction, OpCodes also benefit the application of graph theory in malware detection. [28] proposed a malware detection method in executable files by leverage graph using Power Iteration method embedded in graph properties such as eigenvectors and eigenvalues. Then, classification technique was used to classify each vector as malicious or benign. On the other hand, [29] proposed an android malware detection method using weighted probability graph of Dalvik Opcode. The author has presented important steps, which are Opcode sequences will be constructed into directed graph. Then, graph pruning will be executed as to reduce complexity of the graph structure while preserving critical information. Next, the author extracted and analyzed the similarity of features based on centrality and distance of graph in each malware samples using Manhattan Distance.

Furthermore, directed acyclic graph or DAG also can be used for malware classification. [30] used DAG technique as the graph inherits the properties of classification techniques in feature collection for different malware samples. In behavior-based detection method, [31] proposed graph-mining approach with malware behavior information such as system calls will be represented in Quantitative Data Flow Graph (QDFG). Then, the pattern of graph will be used for machine learning classifier to match between malicious or benign software.

The author in [32] proposed an approach for botnet detection where the method extracted malware samples and network traffic from darknet big data in order to investigate malware types and properties. The author utilized graph theorical of maximum spanning tree by implementing modified version of Kruskal's algorithm. Cyberattacks comes from many vectors. Consequently, [4] addressed this issue using multimodal graph approach to identify possible sources or vectors such as actors, actions and means of cyber-attack. Then, the centrality of the graph will be measured in order to identify the highest value or most influential nodes in multimodal graph. Clustering technique, also one of the machine learning technique, has been widely used by researcher when it comes to malware. The author in [33] proposed malware clustering evaluation model using undirected topological graph to construct all malware samples guided by antivirus label information named Malware Relation Graph.

The research found several gaps in literatures, which are related to research problems. The researches in [1], [26], [27], [29], [30], [31], [32], and [33] have been identified with the absence of specific malware used and specific malware behaviour in literature which referred to the insufficient knowledge of malware that has been used during experiment, since each malware comes with massive amount of variant with various amount of behaviour. [28] comes with insufficient presentation of graph that is related to graph and not presented in details thus, the graph has lack of understanding. Although static analysis of graph comes with a complete analysis of an attack [4], a graph takes better advantage in dynamic analysis as the malware behaviour observation can be done in real-time with controlled environment.

VI. Methodology

To understand the behaviour of the ransomware, the research simulates the environment of the attacks by doing an experiment. This experiment consists of activities such as gathering the ransomware samples, selecting the tools to capture the behaviour of the ransomware and visualizing experiment result by using graph database tools for further discussion.

A. *Design of Experiment*

Fig. 3 shows a testbed environment where the experiment has been conducted along with collecting ransomware samples. The testbed environment is needed where the ransomware behavior will be captured using analysis tools in each Client and Server virtual machine. The initial of the testbed is to create the two VMs in main desktop using VMware Workstation 14 Pro then, the DHCP server of desktop will creating a set range of IPs for each ransomware samples running in each test. The operating system in each VMs are using Windows 7 and 4 GB of RAM. Table I shows the specifications of client and server virtual environment:

Two effective network analysis tools have been selected for the experiment and analysis stage where the tools are open source and have been used for multiple time for researcher to capture the data or even to analyze the data since analyzing the data are its main function. Process Monitor or known as ProcMon is a monitoring tool that shows a real-time environment of file system activity, registry and process activity. These logs are arranged into several columns that make them easier to read such as process ID, the operation of process, directory path and result of the process. ProcMon also can be used in monitor and record malware activity since it provides filtering function. While, Wireshark is used to capture the flow of network traffic and analyze the packet that has been captured through network interface card. Packet that has been captured will be presented into multiple types of information such as time, source, destination, protocol, length and the info of each packet frame. Finally, the research leverages a graph database tool to visualize the result from the experiment.

As for the datasets, the process of capturing starts with accessing a GitHub where most of ransomware samples are given for research purpose. Then, selected ransomware will be downloaded into executable format since the experiment is based on Windows operating system. Finally, the samples will be verified by the dynamic malware analysis sandbox, which called VirusTotal to authenticate the MD5 checksum as shown in Table II.

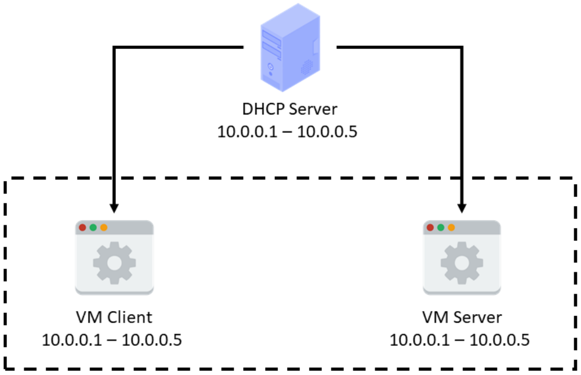

Fig. 3. Testbed Environment

TABLE. I. VIRTUAL ENVIRONMENT SPECIFICATION IN CLIENT AND SERVER

	Client	Server
Host OS	Windows 10 Pro	
Software	VMware Workstation 14 Pro	
Virtual OS	Windows 7 Ultimate SP1 64bit	
Virtual Memory	4 GB	
Virtual Processors	1 core per processor	
Virtual Hard disk	60 GB	
Virtual Network Adapter	VLAN	
Virtual Software	Wireshark v2.6.1, Procmon v3.5	

TABLE. II. RANSOMWARE DATASET PROPERTIES

Ransomware	MD5	File Size	File Type
Badrabbit	fbbdc39af1139aebba4da004475e8839	431.54 KB	Win32 EXE
Cerber	8b6bc16fd137c09a08b02bbe1bb7d670	604.5 KB	Win32 EXE
GoldenEye	e3b7d39be5e821b59636d0fe7c2944cc	254.5 KB	Win32 EXE
Jigsaw	2773e3dc59472296cb0024ba7715a64e	283.5 KB	Win32 EXE
Mamba	409d80bb94645fbc4a1fa61c07806883	2.3 MB	Win32 EXE
Mischa	8a241cfcc23dc740e1fadc7f2df3965e	878.5 KB	Win32 EXE
Rensenware	60335edf459643a87168da8ed74c2b60	96.5 KB	Win32 EXE
Satana	46bfd4f1d581d7c0121d2b19a005d3df	49.67 KB	Win32 EXE
TeslaCrypt	6e080aa085293bb9fbdcc9015337d309	257.5 KB	Win32 EXE
WannaCry	84c82835a5d21bbcf75a61706d8ab549	3.35 MB	Win32 EXE

B. Analysis Process

The flow of the experiment in Fig. 4 starts with using capturing tools that are used to capture dataset in data capturing process. Both tools are started to capture the network traffic and normal process before the execution of the malware samples. After the malware has been executed within certain period, the data analysis process starts to analyses the data from both results based on the tools, data network traffic from the Wireshark tool and resulting huge set number of PCAP (packet capture) files whereas data file activity system from Process Monitor tool resulting massive data log from the testing environment. Each of the data has been analyzed by using filter that provides by the tools to reduce the workload of analyzing both raw data information.

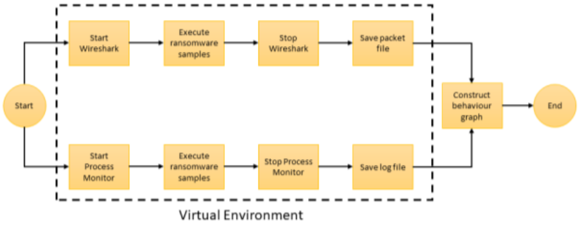

Fig. 4. Analysis Process in Flowchart

VII. ANALYSIS RESULT AND DISCUSSION

Visualizing and constructing the graph is quite challenging since node and edges need to be clarified before represented in the form of graph based on the result of experiment. Thus, a graph database tool is needed to assist in visualizing and representing the data. Neo4j is a property-graph type model, which uses node and edges concepts [34]. Multiple or single directed edge is used in Neo4j to define relationship between these nodes which means the nodes can possibly have multiple relationship with other nodes as well. A basic graph in Neo4j model consists on several elements, which are nodes, relationship, properties and labels. The nodes are described as the main element that connected to other nodes using relationship. The node and edges have properties that can be stored as key-value whereas label is described as roles to define types of node in the graph [35].

Additionally, the same concept of Relational Database Management System (RDBMS) is applied to graph database such to construct the graph, node and edges and need to be declared much like primary key and foreign key in RDBMS. Compare to other graph database tools, Neo4j provides its own database syntax called Cypher Query Language (CQL) which comparable to SQL that has been optimized for query in graph database so multiple variation of graph and complex conceptual connection can be visualized and expressed respectively [34] [36]. Therefore, by using Neo4j, the analysis result will be visualized into main graph, consists of multiple nodes that each node represents set of data analysis from the experiment.

Based on the overall graph model in Fig. 5, there are four types of nodes called Ransomware, FileOpen, FileCreate and FileExecutableUnderMalwareProcessTree. Ransomware node represent each sample has been used in the experiment. FileCreate and FileCreate nodes represent one or many types of DLL that have been accessed by each sample during the experiment and various sample files that have been infected by these ransomwares, respectively whereas FileExecutable UnderMalwareProcessTree node representing executable process that has been created during the experiment.

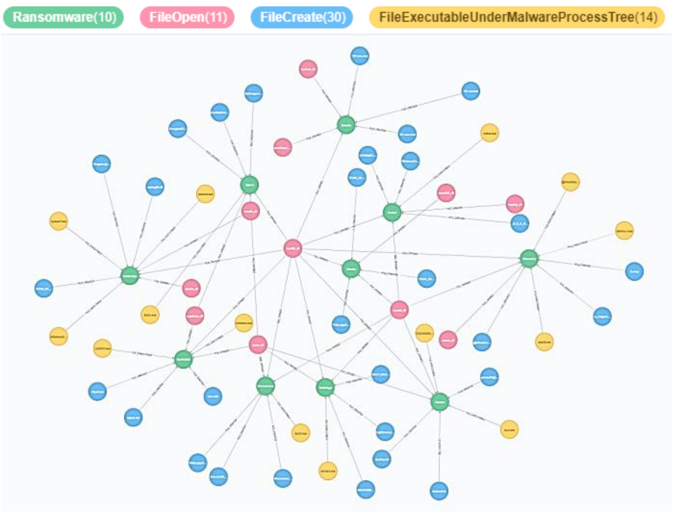

Fig. 5. Overall Graph Model.

There are several nodes shared the same behavior which are accessed the dynamic link library (DLL) during the execution of the ransomware. Other ransomware samples have their own distinguish DLL and the file that have been created in the file directory. All ransomware samples also create their own file executable process during the experiment. Below is the analysis of subgraph among ransomware behavior and its relationship between each file.

A. Badrabbit Ransomware

BadRabbit ransomware capable in deceiving victims into clicking it by creating false notification about Flash player update since it can hide using fake Adobe Flash. Then, the ransomware restarts the system after the attack entered the filesystem. During the process of execution, the ransomware will prompt UAC or User Access Control in order to obtain privilege since certain of the files need to have user permission. After that, it creates several malicious files in Windows directory such as infpub.dat which that responsible for modifying bootloader and encrypting the files.

BadRabbit ransomware subgraph contains multiple sets of nodes and edges that connected to each other as shown in Fig. 6. These set of nodes are labeled based on DLL files and has been accessed by this ransomware or the ransomware created the file or process in file system activity. From the analysis result, cryptbase.dll is among of highlighted DLL that has been accessed by this ransomware. This is because; the DLL is the Base cryptographic API DLL that was introduced in Windows NT 4.0 to provide services that enables developers to secure Windows-based applications using cryptography. Thus, the ransomware leverages the process to use its function to do malicious activity. Furthermore, it creates other file such as infpub.dat and cscc.dat, which are the main module of the malware to execute other process. Also, it creates other process under malware process tress, which are rundll32.exe and schtasks.exe. This process disguises as a normal behavior since it is created in Windows directory.

B. Cerber Ransomware

CRBR Encryptor or Cerber is among of ransomware that capable to encrypt the files even though the victims do not connect to the Internet. Like other ransomwares, file extension also has been renamed by the ransomware, namely, “.ba99”,”.98a0", ".a37b" and ".a563". However, the result of the experiment shows Cerber renamed the extension files as “.bdfa” due to variation version of ransomware.

In Fig. 7, the subgraph shows multiple nodes and edges connected to the main node of Cerber ransomware. Based on the data results, rsaenh.dll is among DLL files that have been highlighted in this subgraph because the function of this DLL is to implement 128-bit encryption of cryptographic service provider (CSP). Therefore, the ransomware leverages the process to use its function to do malicious activity. Likewise, it creates other process under malware process tress, which is mshta.exe. This process disguises as a normal behavior since it is created in Windows directory. Another DLL files that have been access are imm32.dll and cryptsp.dll. Among the files that have been captured are WindowsCodecs.dll, 1dTbfrlajT.bdfa and _R_E_A_D___T_H_I_S___SG08K_.txt. Also, it creates other executable process under malware process tree called mshta.exe.

C. GoldenEye Ransomware

GoldenEye is a type of ransomware that need to obtain administrative permission to proceed the encryption of the files. The unique behavior pattern of this ransomware is it capable to change the Master Boot Record (MBR) with custom boot loader. Then, the computer automatically reboot itself, showing a fake check disk while it performs encryption activity in the background process thus, recovering the data is practically impossible.

Based on the data result shown in Fig. 8, the ransomware creates additional malware called msimg32.dll in Windows filesystem, which is a Trojan dropper. Another typical behavior from this ransomware is it creates a file such as ransom note and ‘x4jBy3PY’ extension file, which is from the file that has been encrypted by the ransomware. Similarly, it creates other process under malware process tress, which are xwizard.exe, typeperf.exe and InfDefaultInstall.exe. These processes disguise as a normal behavior since they are created in Windows directory. Besides, other DLL files that have been access by this ransomware are wow64.dll and cryptsp.dll. Likewise, the files have been captured are Penguins.jpg.x4jBy3PY and YOUR_FILES_ARE_ENCRYPTED.TXT.

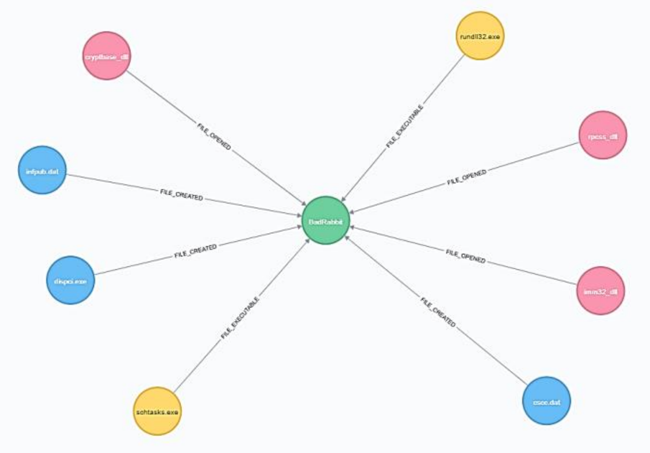

Fig. 6. Subgraph of Badrabbit Ransomware.

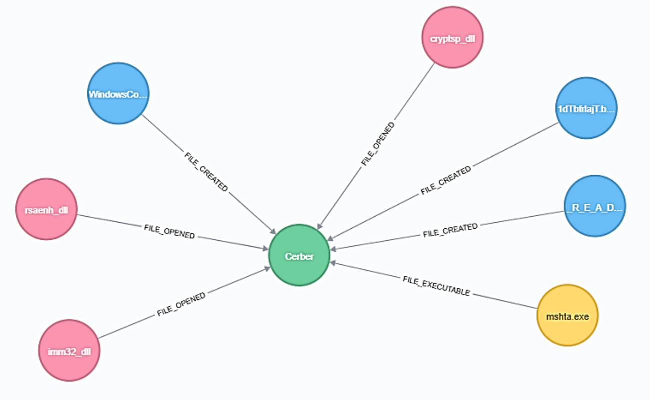

Fig. 7. Subgraph of Cerber Ransomware.

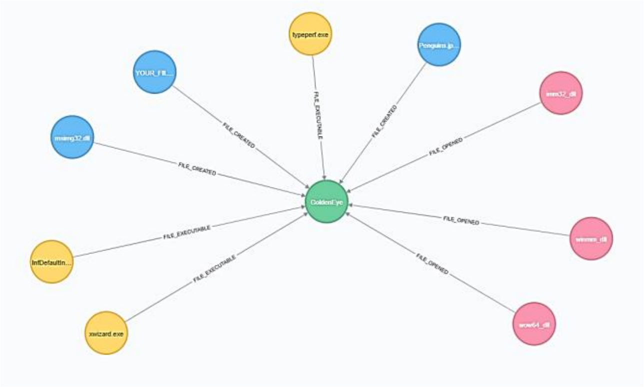

Fig. 8. Graph of GoldenEye Ransomware.

D. Jigsaw Ransomware

One of the distinct key features in identifying Jigsaw ransomware is the ransomware displays a ransom note featuring a character name Billy from movie called Saw. Based on the observation during experiment, the ransomware permanently deletes files from file directory if the ransom has not been done in specific time given.

One of the highlighted DLL files in Fig. 9 is cryptbased.dll. This DLL is the Base cryptographic API DLL that is introduced in Windows NT 4.0 to provide services that enables developers to secure Windows-based applications using cryptography. It also creates other file such as ransom note and ‘fun’ extension file, which is from the file that has been encrypted by the ransomware. Also, it creates other process under malware process tress, which are drpbx.exe. This process disguises as a normal behavior since it was created in Windows directory. Another DLL files that have been accessed are benign files such as imm32.dll and rpcss.dll. Also, there are few files are created by Jigsaw which are Hydrangeas.jpg.fun, RacWmiDatabase.sdf.fun and EncryptedFileList.txt.

E. Mamba Ransomware

HDDCryptor or known as Mamba is a type of ransomware that targets network sharing devices such as network printers, disk drives or network ports using SMB or Server Message Block. Like GoldenEye, the ransomware also requires a permission from administrator to change MBR or Master Boot Record.

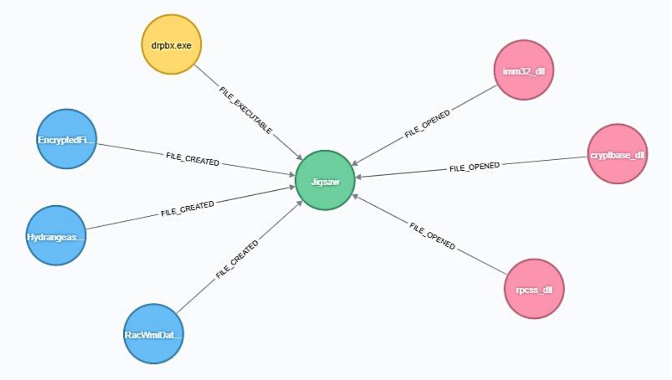

Fig. 9. Graph of Jigsaw Ransomware.

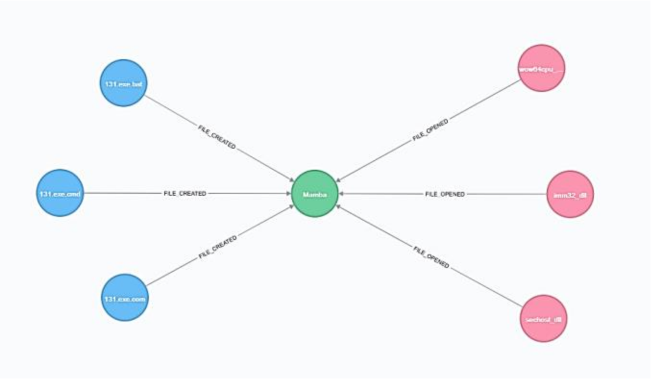

Fig. 10. Graph of Mamba Ransomware.

The subgraph of Mamba ransomware in Fig. 10 shows multiple nodes and edges connecting to main node. The highlighted nodes are DLL files that have been accessed by this ransomware called wow64cpu.dll. The DLL files are used to switch the processor from 32-bit to 64-bit mode when the application needs to be run in 64-bit format. Most of the applications that are using the Wow64 subsystem are created in SysWOW64 directory. Although it does not have an encryption in the experiment, it does have file activity processes, which are bat file, cmd file and com file in the same directory of the malware. Another DLL that has been accessed by this ransomware are imm32.dll and sechost.dll.

F. Mischa Ransomware

Mischa ransomware considered as a successor of Petya ransomware by its creators and it has become highly dangerous when it comes to sophisticated behavior. Different from Petya, the ransomware starts its behavior by scanning the system that has anti-virus software. Then the ransomware starts the encryption process while creating two ransom notes called YOUR_FILES_ARE_ENCRYPTED.HTML and YOUR_ FILES_ARE_ENCRYPTED.TXT to every folder in file directory.

The subgraph in Fig. 11 shows Mischa ransomware with multiple nodes and edges connecting to the main node. The highlighted nodes are DLL files that have been accessed by this ransomware called rsaenh.dll, which implements 128-bit encryption of cryptographic service provider (CSP). It also creates another file such as ransom note in text file and html file and ‘6NRS’ extension file image, which is from the file that has been encrypted by the ransomware. Based on our experiment, there is no process that has been created under this malware process tree.

G. Rensenware Ransomware

Rensenware is created with non-malicious intent and it is accidentally distributed in network that targets Windows OS users. However, if the ransomware finds several files that cannot be encrypted, it will crash itself and cannot be executed. The unique behavior of this ransomware is the victims are required to play certain game called “Touhou 12: Undefined Object”. The victim needs to achieve 200 million in “Lunatic” difficulty. Another interesting behavior is the ransomware does not delete the encryption key because it does not have one.

The highlighted nodes as show in Fig. 12 are files that have been created by this ransomware which each files extension that has been encrypted by this ransomware are renamed as "RENSENWARE". Likewise, it creates other process under malware process tress, which is dw20.exe. This process disguises as a normal behavior since it was created in Windows directory.

H. Satana Ransomware

Satana or Satan ransomware is among of ransomware that operates as RaaS or Ransomware-as-a Service platform. Since it serves as a service, an attacker can implement various constraints or multiple behavior patterns based on functionality implementation. It relies on AES encryption module to encrypt victim's data and demand for ransom using ransom note.

The highlighted nodes are files that has been created by this ransomware which it creates ransom note called "!satana!.txt" after encrypting files in each folder as shown in Fig. 13. The unique behavior of this ransomware is the encrypted file is always have this pattern which is "<email_address>__<original_name of file>". Also, it creates other process under malware process trees, which are qxyi.exe and VSSADMIN.exe. This process disguises as a normal behavior since it was created in Windows directory.

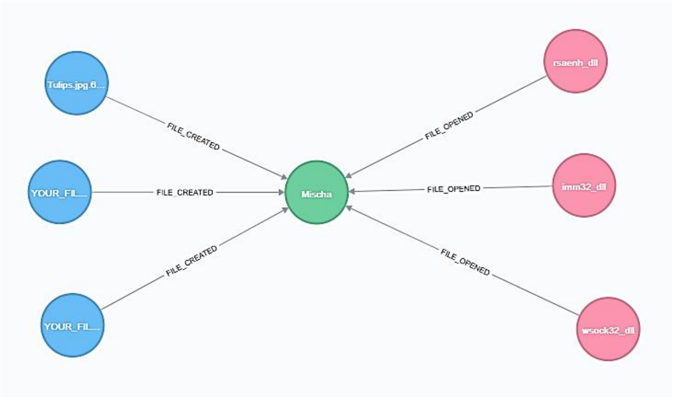

Fig. 11. Graph of Mischa Ransomware.

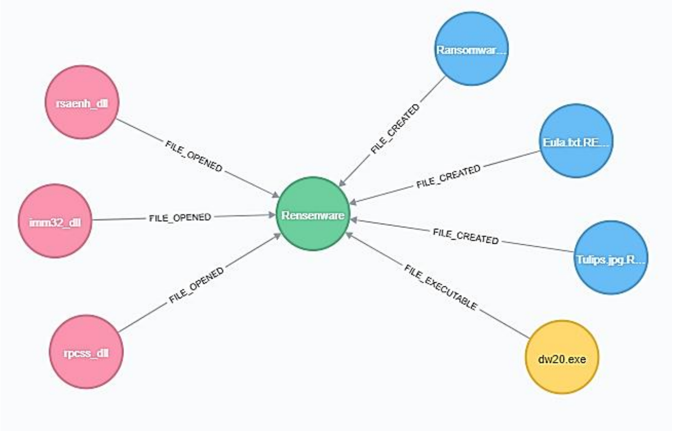

Fig. 12. Graph of Rensenware Ransowmare.

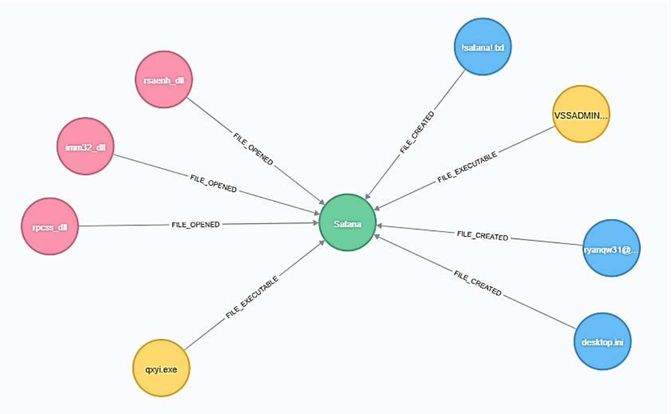

Fig. 13. Graph of Satana Ransomware.

I. TeslaCrypt Ransomware

TeslaCrypt ransomware behaves like other typical ransomware which it leaves ransom note called "HELP_RESTORE_FILES.txt" in each directory after encryption activity has been done. The unique behavior of this ransomware is it specifically target video game data such as data save or game settings in game file directory though, other variant targets different types of game files. Another similar pattern is it uses AES encryption for encrypting data files.

The nodes in Fig. 14 shows DLL that have been accessed by this ransomware are imm32.dll, rpcss.dll and rsaenh.dll whereas the nodes that have been created are Lighthouse.jpg.ecc, RECOVERY_KEY.TXT and HELP_ RESTORE_FILES.txt. Also, it creates other process under malware process trees, which is envtact.exe. This process disguises as a normal behavior since it was created in Windows directory.

J. WannaCry Ransomware

WannaCry or Wana Crypt0r is not something new when the ransomware spread the attack in May 2017 causing chaos around the world which giving awareness about how dangerous of ransomware. The ransomware uses RSA-2048 that is impossible to decrypt thus victims are required to pay ransom in Bitcoin based on the ransom note.

The highlighted nodes shown in Fig. 15 are FileCreate which is "@WanaDecryptor@.exe". The function of this process is to show timers in ransom note and display the instruction of payment based on the language of operating system. It also creates another file such as 'wnry' extension, which consists of language, and normal file that is from the file that has been encrypted by the ransomware. Also, it creates other process under malware process trees, which is taskdl.exe and taskhsvc.exe. This process disguises as a normal behavior since it is created in Windows directory. Other files that have been captured during the experiment are Ransomware. WannaCry\b.wnry and m_bulgarian.wnry.

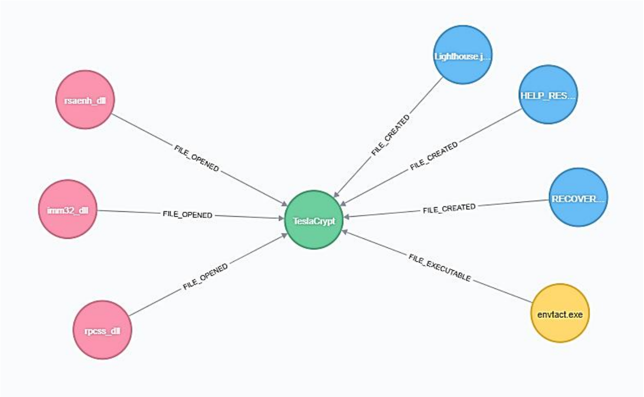

Fig. 14. Graph of TeslaCrypt Ransomware.

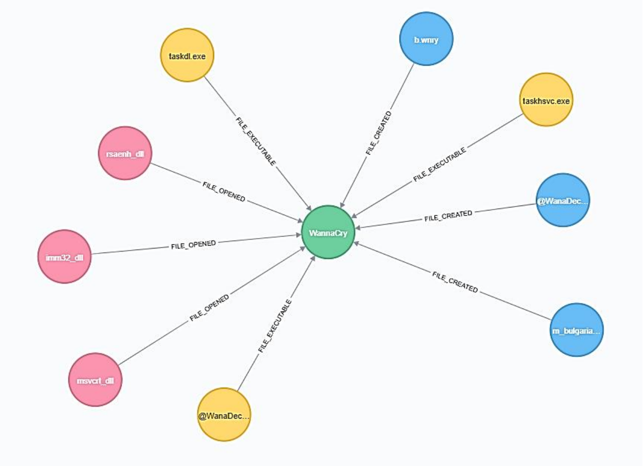

Fig. 15. Graph of WannaCry Ransomware.

VIII. Conclusion and Future Works

In conclusion, this research paper proposes an alternative method in representing the data of ransomware behavior using Neo4j graph database tools. Based on the research experiment, the data are analyzed the by classifying the ransomware behavior using analysis tools based on the log and network provided. Therefore, this research contributes in developing the level of awareness and knowledge of correlation between ransomware and graph theory approach besides providing different method in malware detection field community.

Graph theory analysis provides a new approach in malware detection. With the graph analysis, researchers can find significant relation of malware behavior, as the data from the experiment are represented as vertices and edges. The graph analysis also provides good representative based on the result from the experiment and gives better understanding of ransomware behavior in file system.

Currently, the limitation of this research is the experiment developed in offline environment, which does not project similar behavior of ransomware in online environment. Few ransomware samples also outdated or obsolete which means the old behavior does not reflect the latest ransomware with more sophisticated behavior. Thus, future research is to provide bigger scope by using multiple types of data from multiple sources such as memory and registry file. To improve the quality of the research, the next approach must have online environment features in order to capture live ransomware behavior from the experiment. The ransomware samples also have to increase as different behavior can be analyzed in comparative analysis. Moreover, with the current result of the experiment, it is a need to expand the research towards detection scheme as the analysis approach can take advantage in this scope.

Acknowledgment

This work has been supported under Universiti Teknikal Malysia Melaka research grant GLUAR/APNIC/2018/FTMK-CACT/A00018. Thank you to Research Group of Information Security Forensics and Computer Networking, Center for Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi (FTMK), Universiti Teknikal Malaysia Melaka.

References

[1] A. Mpanti, S.D. Nikolopoulos, and I. Polenakis, “A Graph-based Model for Malicious Software Detection Exploiting Domination Relations between System-call Groups,” Proceedings of the 19th International Conference on Computer Systems and Technologies, pp. 20-26, September 2018.

[2] C. Birkinshaw, E. Rouka, and V.G Vassilakis, “Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks,” Journal of Network and Computer Applications 2019, vol. 136, pp. 71–85.

[3] C.H. Lin, H.K. Pao, and J.W. Liao, “Efficient dynamic malware analysis using virtual time control mechanics,” Computers and Security 2018, vol. 73, pp. 359–373.

[4] N. Ghose, L. Lazos, J. Rozenblit, and R. Breiger, “Multimodal graph analysis of cyber attacks,” Spring Simulation Conference (SpringSim), pp. 1-12, April 2019.

[5] R. Agrawal, J.W. Stokes, K. Selvaraj, and M. Marinescu, “Attention in Recurrent Neural Networks for Ransomware Detection,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3222-3226, May, 2019.

[6] M. Akbanov, V.G. Vassilakis, and M.D. Logothetis, “Ransomware detection and mitigation using software-defined networking: The case of WannaCry,” Computers & Electrical Engineering 2019, vol. 76, pp. 111-121.

[7] G. Hull, H. John, and B, Arief, “Ransomware deployment methods and analysis: views from a predictive model and human responses,” Crime Science 2019, 8(1), p. 2.

[8] B.A.S. Al-rimy, M.A. Maarof, and S.Z.M Shaid, “Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions,” Computers and Security 2018, vol. 74, pp.144–166.

[9] C. Chew, and V. Kumar, “Behaviour Based Ransomware Detection,” Proceedings of 34th International Conference, vol. 58, pp. 127-136, March 2019.

[10] N.K Popli, and A. Girdhar, “Behavioural Analysis of Recent Ransomwares and Prediction of Future Attacks by Polymorphic and Metamorphic Ransomware,” Advances in Intelligent Systems and Computing 2019, vol. 799, pp. 65–80.

[11] S.H. Kok, A. Abdullah, N.Z Jhanjhi, and M. Supramaniam, “Ransomware,Threat and Detection Techniques : A Review,” International Journal of Computer Science and Network Security 2019, 19(2), pp.136–146.

[12] T. Sangkaran, A. Abdullah, N. JhanJhi, and M. Supramaniam, “Survey on Isomorphic Graph Algorithms for Graph Analytics,” International Journal of Computer Science and Network Security 2019, 19(1), pp. 85–92.

[13] J. Xu, and H. Chen, “Criminal network analysis and visualization,” Communications of the ACM, 2005, 48(6) pp. 100-107.

[14] R. Mehatari, M.R. Kannan, M.R. and A. Samanta, “On the adjacency matrix of complex unit gain graph”, 2018. arXiv:1812.03747.

[15] O. De la Cruz Cabrera, M. Matar, and L. Reichel, “Edge importance in a network via line graphs and the matrix exponential,” Numerical Algorithms 2019, pp.1-26.

[16] K. Sato, “Optimal graph Laplacian”, Automatica 2019, vol. 103, pp. 374–378.

[17] R. Rossi, and N. Ahmed, “The network data repository with interactive graph analytics and visualization,” 29th AAAI Conference on Artificial Intelligence, pp.4292–4293, March 2015.

[18] G. Drakopoulos, A. Baroutiadi, and V. Megalooikonomou, “Higher order graph centrality measures for Neo4j,” 6th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1-6, July 2015.

[19] S. Noel, E. Harley, K.H Tam, and G. Gyor, “Big-Data Architecture for Cyber Attack Graphs Representing Security Relationships in NoSQL Graph Databases,” 2015.

[20] S. Abraham, and S. Nair, “A predictive framework for cyber security analytics using attack graphs,” 2015. arXiv:1502.01240.

[21] E. Dull, “Cyberthreat analytics using graph analysis,” Cray User Group(CUG ’15), 2015.

[22] R.N Gottumukkala, S.R Venna, and V. Raghavan, “Visual Analytics of Time Evolving Large-scale Graphs,” IEEE Intelligent Informatics Bulletin, 16(1), pp.10–16, 2015.

[23] Y. Ding, X. Xia, S. Chen, and Y. Li, “A malware detection method based on family behavior graph,” Computers and Security, 2018, vol. 73, pp. 73–86.

[24] I. Martin, J.A. Hernandez, and S. de los Santos, “Machine-Learning based analysis and classification of Android malware signatures,” Future Generation Computer Systems 2019, vol. 97, pp. 295–305.

[25] A. Sharma, and B.A Prakash, “Graphs for Malware Detection : The Next Frontier,” Proceedings of the 13th International Workshop on Mining and Learning with Graphs (MLG), pp.8–10, 2017.

[26] F. Xiao, Z. Lin, Y. Sun, and Y. Ma, “Malware detection based on deep learning of behavior graphs,” Mathematical Problems in Engineering, February 2019.

[27] E.B. Karbab, and M. Debbabi, “Togather: Automatic investigation of android malware cyber-infrastructures,” Proceedings of the 13th International Conference on Availability, Reliability and Security, p. 20, August, 2018.

[28] H. Hashemi, A. Azmoodeh, A. Hamzeh, and S. Hashemi, “Graph embedding as a new approach for unknown malware detection,” Journal of Computer Virology and Hacking Techniques 2017, 13 (3), pp. 153–166.

[29] J. Zhang, Z. Qin, K. Zhang, H. Yin, and J. Zou, “Dalvik Opcode Graph Based Android Malware Variants Detection Using Global Topology Features,” IEEE Access 2018, vol. 6, pp. 51964–51974.

[30] M.K. Sahu, M. Ahirwar, and P.K. Shukla, “Improved malware detection technique using ensemble based classifier and graph theory,” 2015 IEEE International Conference on Computational Intelligence & Communication Technology, pp. 150-154, February 2015.

[31] T. Wuchner, A. Cisłak, M. Ochoa, and A. Pretschner, “Leveraging compression-based graph mining for behavior-based malware detection,” IEEE Transactions on Dependable and Secure Computing 2017, 16(1), pp. 99-112.

[32] E. Bou-Harb, M. Debbabi, and C. Assi, “Big data behavioral analytics meet graph theory: on effective botnet takedowns,” IEEE Network 106, 31(1), pp. 18-26.

[33] Y. Chen, F. Liu, Z. Shan, and G. Liang, “MalCommunity: A graph-based evaluation model for malware family clustering,” Communications in Computer and Information Science, 2018, vol. 901, pp. 279–297.

[34] R. Arora, and S. Goel, “JavaRelationshipGraphs (JRG): Transforming Java Projects into Graphs using Neo4j Graph Databases,” Proceedings of the 2nd International Conference on Software Engineering and Information Management, pp. 80-84, January 2019.

[35] Z. Zhu, X. Zhou, and K. Shao, “A novel approach based on Neo4j for multi-constrained flexible job shop scheduling problem,” Computers and Industrial Engineering 2019, vol. 130, pp. 671–686.

[36] D. Allen, A. Hodler, M. Hunger, M. Knobloch, W. Lyon, M. Needham, and H. Voigt, “Understanding Trolls with Efficient Analytics of Large Graphs in Neo4j,” Business, Technologies and Web (BTW) 2019, 2019.