Exploring the API Calls for Malware Behavior Detection using Concordance and Document Frequency

G.S.N.Murthy, M.V.V.Chowdary, M.V.Sangameswar, T.P.R.Vital

Abstract: *In the era of ubiquitous sensors and smart devices, detecting malware is becoming an endless battle between ever-evolving malware and antivirus programs that need to process ever-increasing security related data. Malwares are becoming persistent by creating full-fledged variants of the same or different family. Malwares belonging to same family share same characteristics in their functionality of spreading infections into the victim computer. We find that certain malicious functions are commonly included in malware even in different categories. From checking the existence of certain functions or API call sequence patterns matched, we can even detect new unknown malware. For malware detection, various approaches have been proposed. An Application Programming Interface (API) is widely is used for the software to interact with an operating system to do certain task such as opening file, deleting file etc., Users of the computers use this API to make it comfortable for their program to communicate with the operating system without having the prior knowledge of the hardware of the object system. The attacker also use the same type of APIs to create malware, hence it is very much difficult to know about these APIs. There are many researches done in this field, however, most researchers used n-gram to detect the sequence of API calls. Even though, it gave good results, it is time consuming to process through all the output. Hence, we proposed to use Concordance to search for the API call sequence of a malware because it use KWIC (Key Word in Context), thus only displayed the output based on the queried keyword. After that, Document Frequency (DF) is used to search for the most commonly used APIs in the dataset. The result of our experiment gave high accuracy than other methods and also found more categories than other methods. API call sequence can be extracted from most of the modern devices. Hence we supposed that our method can detect the malware for all types of the ubiquitous devices. The results of the experiment show that Concordance can be used to search for API call sequence as we manage to identify Eight malicious Activities (Screen Capture, Hooking, Downloader, Enumerate all process, Anti debugging, Synchronization, Key Logger and Dropper) using this method.*

Keywords: API Call Sequence, DF, Dynamic Analysis, ICF, KWIC, Malware Behavior.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author

Dr.G.S.N.Murth*, Dept of CSE, Aditya college of Engineering, Surampalem, India.
M.V.V.Chowdary, Dept of Computer science, V.S.M.College, Ramachandrapuram, India.
Dr.M.V.Sangameswar, Dept of CSE, GIET, Rajahmundry, India.
Dr.T.P.R.Vital, Dept of CSE, AITAM, Tekkali, India.

I. INTRODUCTION

Malware is becoming more sophisticated in design. Now-a-days, with a new variant of malware being discovered, malware is also becoming more notorious over time, with the rising amount of security breaches around the world as well as the severity of said breaches. According to [1], security breaches can cause significant economic damages to an organization as it takes considerable time to fix the damages made. A survey of Security professional of the IT Industry on the attacks of the malware on different organizations since the last one year reveals that more than half of the breaches damaged the worth more than $600,000 in financial organizations, which shows its severity on an organization. The WannaCry ransom ware in 2017 affected more than 2 lakhs of computers in more than 150 countries through worldwide and causes huge financial damages to its victims [2]. In 2018, the number of users which was attacked with banking Trojans is nearly Ten Lakhs, which is an absolute increase of 16% with the previous year. Most of the users in the developed countries like USA, Russia, Germany, China, Vietnam, Italy and India were attacked by banking malware. With an absolute of 25% users which are affected by banking malware are corporate users. Security breaches that are caused by human mistakes may lead to loss of information or data stored in the system as well as their reputation [3]. Not only lead to the loss of information which may also be used to steal someone's identity by using keystroke logger and some form of spyware [4]. Malware is malicious software that can cause harm to our system or network. The consequence of malware attack is not limited to theft of data, destruction of data, system compromise and denial of service (DoS). Due to DDoS Attack, on February 28, 2018 evening for 4 minutes GitHub.com was intermittently unavailable [5]. There are many types of malware such as Virus, Trojan, Worm and Backdoor. Each of these malware behaves differently from each other. Table 1 shows the different type of malware and their description.

Table 1: Malware and Their Description

Category	Description
Trojan	Trojan is malware that masquerade as a legitimate software. Once runs, the Trojan can spy on the user, delete files, steals sensitive data and many more depending on what that Trojan is for. However, unlike virus and worms, Trojan cannot self-replicate itself.
Adware	Adware is a malware that shows ads on the computer. It is mostly annoying rather than malicious in nature.
Spyware	Spyware can be used for key-logging and similar activities, thus helping the hackers to gain access the personal data. It is used to keep a check on the activities of the computer users.
File-less malware	File-less malware infects by moving from one location to another location without using files and systems. Mostly this malware spread in memory and initiated from the already existing program. It is very difficult to detect and prevent.
Ransomware	Ransom ware, spread enormously swift across organizations, networks, and countries. It encrypts files in a system or network and makes them inaccessible.
Worm	Worm is a malware that can propagate itself within the system. Worm can self-replicate itself using the internet or computer network.
Backdoor	A backdoor is a method, often secret, of bypassing normal authentication or encryption in a computer system, a product, or an embedded device, or its embodiment
Virus	Virus is a malware that attached itself to a piece of software that will infect system that run that infected software. This malware will usually infect other system when user share that infected files.

Malware analysis gives us an impending on the functionality of the malware, based on that we are able to ensure the safety and security of our system. The analysis of the malware is to be done with various goals like the understanding the enormity of the infection of the malware, impact of the malware attack as well as to analyze the behavior of the malware. There are three types of malware analysis known as static, dynamic and hybrid analysis.

Static analysis is done by slicing up and studying the malware's code. It is done without executing the malware. The code is disassembling using a dissembler, hence the analyst of the malware can read the code and find out the code is supposed to do. The static analysis is also known as code analysis. In Dynamic analysis the malware is executed in a safe and sound environment, be virtual or sandbox environment. Dynamic analysis, focus on analyzing the behaviors of malware, hence it monitors the activity of registry, API calls, processes and network when running the analysis. Hybrid analysis uses both static and dynamic analysis to analyze the malware.

Static analysis import table to extract the information of the malware will not be able to extract any resolved APIs of the malware. Hence, in this work, dynamic analysis is used to extract the API calls of the malware in order to extract resolved APIs, we need to run the malware since it can only be extracted during runtime and for static analysis do not execute the malware as they will miss this kind of information. Furthermore, in order to shows that resolved APIs does contain information regarding malware behaviors, this research will only use resolved APIs when creating the dataset that is used in this research.

In this research we use a concordance to map the API call sequence of a malware in the dataset. The n - gram method which is used by most of the researchers displayed a huge number of outputs, as the output is based on the n-value. Hence it takes lot of time to browse through it all. For the moment, concordance use key word in context (KWIC) method to display its output, and the output displayed is only based on the keyed keywords and its close context.

The rest of the paper is organized as follows: Section 2 will discuss about the Motivation of the study. Section 3, will discuss the Materials and Methods of the study. Section 4 will discuss about the experimental design of the research. Section 5 will discuss about the results and discussion. Finally, Section 6 will discuss about the conclusion. The purpose of this research is to identify malware behavior by using API calls.

This research will focus on:

1. Using dynamic analysis to extract API call and map the API call sequence using the concordance tool.
2. To find out the most commonly used APIs in the dataset by using Document Frequency (DF) and Inverse Corpus Frequency (ICF)

II. MOTIVATION

A classification program that uses Voting Expert algorithms to precisely recognize episode boundaries and these episode boundaries are then used to search the API call sequence [6]. Besides, they also track the changes made to the OS by malware. They wrote that the API call combines with OS state changes made by the malware will increase the accuracy of malware classification. Another approach is the extraction of malicious behaviors of malware as a set of k-grams, which is then used to compare the similarity between API calls to the k-grams to identify whether it is the malicious behavior or not [7]. In literature, a novel approach to analyze API call sequence which is to use sequence alignment algorithm and another one use k-gram to analyze API call sequence to determine whether it is malicious or not. The comparative study of the above two to identify API call sequence of malware's malicious behavior is done [8]. Another approach defined for Malware characterization Using Windows API Call Sequences [9]. In this classification only five different classes of malware were analyzed. Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics [10]. Another approach of Malware analysis using multiple API Sequence Mining control Flow Graph [11] is very useful in detecting variants of one malware family and observed that some mathematical research to determine more efficient similarities need to be taken up in future. The above research not addressed the Short comings as there are no new categories of malware to extend the explicable of the framework and also not aimed at the challenges due to the huge number of malware variants, which can require deep learning techniques to improve detection efficiency. Most of the malware detection systems depend on the signature of a malware's static information such as the size of the file, process and its artifacts. Hence, they are unable to detect new unknown malware until the signature has been updated.

All the reviews revealed the importance of API call in the understanding and detection of malware.

The API calls play a major role in understanding the malware itself, thus helps in identifying both existing and new variants of malware. Likewise, by observing the API calls of a malware, one can see how the malware work because API calls represents a specific operation that is used to run specific tasks. Finding this malicious behavior and their APIs is one of the objectives of this research.

III. MATERIALS AND METHODS

Dynamic Analysis

This study uses dynamic analysis to extract the API calls of a malware. Dynamic analysis can be done using two ways, manually or automatically. In Manual dynamic analysis the malware will be executed in a virtual environment and malware analyst will monitor the malware and observe the changes in the processes which caused by the malware. Table 2 shows the list of tools usually used in doing manual dynamic analysis.

Table 2: Manual Dynamic Analysis Tools

Tool	Description
API Monitor	API Monitor is a software which allows us to spy and display Win32 API calls made by applications. It can trace any exported APIs and display wide range of information including function name, call sequence, input and output parameters, function return value and more.
Valgrind	Valgrind is a programming tool for memory debugging, memory leak detection and profiling. Valgrind was designed to be a free memory debugging tool for Linux. But has since changed to become a generic framework for creating dynamic analysis tools such as checkers and profilers.
Process monitor	Process Monitor is a monitoring tool for Windows system that shows file system, Registry and process or thread activity in real time.
Glances	Glances are a CLI curses based monitoring tool for GNU/Linux and BSD OS. Glances use the PsUtil library to get information from your system. It is developed in Python.
WinSpy++	WinSpy++ is a practical programmer's service which can be used to select and view the properties of any window in the system. It is based around the Microsoft Spy++ utility that ships with Microsoft Visual Studio.
SysTracer	SysTracer is a system utility tool which can scan and analyze the computer to trace the changed data into registry and files such as added, modified or deleted. It can scan system and records the information about the opened files, folders and registry information.

On the other hand, automated dynamic analysis is done in a sandbox environment. Sandbox is used mostly to test unverified programs that may contain a virus or other malicious code, without allowing the software to harm the host device. There are many types of sandbox available for free or commercially in use. One of the most used free sandboxes is SHADE sandbox. Using automated sandbox in analyzing malware is simple and easy. We only have to submit the malware sample and wait for its execution. Furthermore, we can download the reports generated by the sandbox and analyze the malware based on the reports. As mentioned before, there are many automated sandbox available which are shown in Table 3.

Table 3: List of Automated Sandbox

Sandbox	Description
SHADE Sandbox	Sandbox application works well for all versions of windows (XP - Win 10) on both 32 & 64 bit.
Anlyz Sandbox	Online free sandbox. User uploads the malware file and can download the reports in HTML or PDF format. However, this sandbox only allows 10 submissions per account per day.
Cuckoo Sandbox	Free automated open-sourced sandbox. Since the sandbox will be set up on user computer, submission is limitless. However, the installation can be an arduous task.
Joe Sandbox	Joe Sandbox is commercial sandbox. However, user can use it for free albeit with limited functionalities and limited submission per account.
Mbox	Mbox is a lightweight sandboxing mechanism that any user can use without special privileges in commodity operating systems.

API

Application programming interface (API) is a collection of rules which enables the developers to develop system or software for a specific operating system with little knowledge about that particular operating system. There are two types of APIs that we will encounter during this experiment which is WinAPI and NTAPI.

WinAPI is a normal APIs that is normally used in developing a new Windows program. The name Windows API refers to numerous platform implementations that are often referred to by their own names (Win32 API) WinAPI uses suffix in their name such as A, W, and Ex. APIs with A and W suffix have the same function, but differ in their accepting parameters as one accept ASCII string while the other accept a wide character string. Meanwhile, APIs with Ex suffix means that it is an API with an updated functionality.

NTAPI is not so commonly used in developing new programs, and most of NTAPI is undocumented. NTAPI used prefix in their name such as Nt, Zw, Rtl and Ldr. NTAPI is commonly used by malware author because it offers powerful and stealthier functionalities than it WinAPI. The Table 4 shows the list of malicious behaviors and their API call sequence.

Table 4: Malicious Behaviors and Their API Call Sequence

Malicious Behavior	API Call Sequence
Modify File Attribute	GetFileAttribute, SetFileAttribute
Modify Time of File	GetFileTime, SetFileTime
Load Register	RegCreateKey, RegSetValue, RegCloseKey
Enumerate all process	CreateToolhelp32Snapshot, Process32First, Process32Next, WTSEnumerateProcesses
Privilege Escalation	OpenProcessToken, LookupPrivilegeValueA, AdjustTokenPrivileges
Terminate Process	TerminateProcess

IV. EXPERIMENTAL DESIGN

Concordance

Concordance is a list of all examples of the search word or phrase found in a corpus. Concordance is a corpus analysis method which fetches all the incidences of a queried search pattern in its instantaneous contexts [12]. It is also possible to sort the concordances. Then it can be identify the patterns that might otherwise go undetected. Concordance is a list of patterns or word in a text that is arranged with surrounding words so that the patterns surrounding the keywords can be visually identified. Concordance usually uses Key term in Context to display their output. It is mostly used in corpus linguistics to analyze the concordance, Document frequency, collocate and cluster of a corpus. Based on the previous researches, we can deduce that concordance KWIC is a pretty powerful tool to analyze patterns. Hence, concordance are used to map malicious API call sequence, which are available for free are shown in Table 5.

Table 5: Concordance Tools.

Tool	Description
AntConc	It is a freeware tool for concordance and text analysis. It is very easy to use and user has selection of option to choose from such as concordance, collocate, word list, keywords, cluster and n gram.
WordSmith	It is a tool that can be used to search concord, keyword and word list. However this tool is only available for Windows user.
TextSTAT	It is a simple program for text analysis. It produces word frequency lists and concordances according to its corpus.

Document Frequency

Document frequency (DF) is used to find the importance of a term within a document. Here the document is considered as the central focus. But, the corpus here is particular sites like yahoo, face book, twitter etc. The count here is not number of occurrences but the number of times the particular document is viewed.

According to [13], DF-ICF is based on two factors Document Frequency and inverse corpus frequency. The document count is the number of views of a particular document in the corpus. The corpus can be the internet as a Whole considering the upper bound or it can be a part of the internet like the data in a particular site or center. The document frequency within a given corpus is as follows:

$$df_{x,y} = \frac{n_{x,y}}{\sum_k n_{k,y}}$$

-

------------- (1)

Where $n_{x,y}$= Number of times the document d_x is viewed from the corpus c_y.

The inverse corpus frequency is a measure of the importance of the document is defined as

$$icf_x = log \frac{|C|}{|\{y: d_x \epsilon c_y\}|}$$

----------------(2)

Where │c│= Total no of corpus

In the DF-ICF, the importance of a document increases with the number of times it is viewed similar to the way the importance of a term increases with its number of occurrences as compared with the other algorithms. When we use DF-ICF, it can filter out the documents which are of less importance and avoid the unnecessary calculations. Only saves our time and also more reliable. It is more suitable for larger data sets.

Finally DF-ICF is calculated as:

$$(df - icf)_{x,y} = df_{x,y} \times icf_x$$

----------------------------- (3)

Malware Samples

Malware samples are used in this research to observe behavior of the malware and its frequency, based on that we may come up with an appropriate solution. This research work is done based on the quantitative methods. The process of the research is shown in Fig 1. Below:

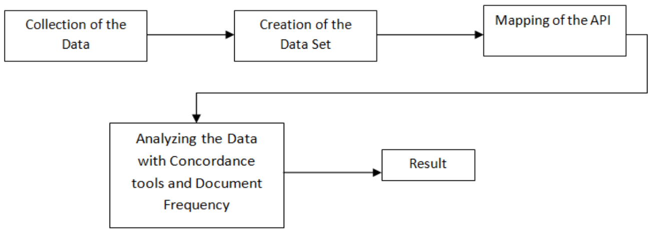

Fig 1: Process of Methodology

Creation of the Dataset: While creating the samples used in the creation of the dataset, the file is taken from the malware file and taken care that it must be detected by Kaspersky which is used for smaller organizations. Along with that it must be detected by quick heal, Norton and Avira also which are used for bigger organizations. After the data set created the selected samples will be submitted to Mbox to do dynamic analysis. Mbox is a lightweight sandboxing mechanism that any user can use without special privileges in commodity operating systems. Then, the generated reports will be downloaded to be used in the next step. The dataset is created using Calligra Suite, WPS Office and Soft Maker Office. The information such as the malware category, RIPEMD160 and the resolved API are extracted from the reports. The population of the sample is 1280; however, only 371 samples match the criteria imposed on the samples. The categories of the malware in the dataset are shown in Table 6.

Table 6: Malware Categories and Their Number of Samples.

Category	Number of Sample
Spyware	25
Adware	20
Backdoor	34
Virus	28
Worm	13
hybrid attack	120

Retrieval Number F9144088619/2019©BEIESP
DOI: 10.35940/ijeat.F9144.088619
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Trojan	109
Others	22
Total	371

Mapping of the API

Mapping of the API is the process of mapping known malicious or suspicious API to the API from the dataset. This step will map the dataset APIs to the most commonly used API calls which is gathered on Section 3. The API mapping is done using Python codes. This process is done to sack any API that is not measured as malicious or suspicious.

Finally the result of this process is that we get known malicious or suspicious APIs from the dataset which is 371 known malicious or suspicious APIs.

Analyzing the Data using Concordance and Document Frequency:

The concordance tool TextSTAT is in this research. The steps of using TextSTAT are as follows:

i) From the dataset Assemble a list of APIs into a plain text file (.txt). Concordance tool's corpus must be in plain text format.
ii) Compile a list of API call sequence gathered into another text file.
iii) After that, give input API from previous step into the keyword box in the TextSTAT and click start.
iv) TextSTAT will display the results.
v) Record the frequencies of the results.

As explained earlier, WF is used to calculate the most commonly used APIs in the dataset. The API calls used in this step are chosen randomly and not based on the categories of the APIs. This is because the WF is used to show which of the malicious or suspicious APIs is favorable by the malware in the dataset. There are a few steps involved when calculating word frequency of the APIs. The steps are as follows:

i) Create a list of APIs based on the malware categories in the dataset and map the API to the result of the API mapping.

ii) Calculate the frequencies of the API.

iii) During the Mapping process of API, we got 371 known malicious or suspicious API in the dataset, which we will use as our N.
iv) Meanwhile, the $n_{x,y}$ is the result and use Eq. 1 to calculate the WF.
v) Calculate ICF. There are 5 malware categories in the data set, which we use as $|C|$.
vi) To calculate $|C|$, calculate how many documents the term appears. Then, use Eq. 2 to calculate ICF.
vii) Finally, calculate WF-ICF, by using Eq. 3, and then Store the results.

V. Results and Discussion

Concordance & Document Frequency

The behaviors that we administered to identify using this method is: Screen Capture, Hooking, Downloader, Enumerate all process, Anti debugging, Synchronization, Key Logger and Dropper. The API call sequences of their behaviors were shown in Table 7.

Table 7: Malicious Behaviors and Their API Call Sequence on the Dataset

Malicious Activity	API Pattern
Screen Capture	(GetDC, GetWindowDC), CreateCompatibleDC, CreateCompatibleBitmap, SelectObject, BitBlt, WriteFile
Hooking	SetWindowsHookA, CallNextHookEx
Downloader	URLDownloadToFile, (WinExec,ShellExecute)
Enumerate all process	CreateToolhelp32Snapshot, Process32First, Process32Next
Anti debugging	(IsDebuggerPresent, CheckRemoteDebuggerPresent, OutputDebugStringA, OutputDebugStringW)
Synchronization	CreateMutexA, CreateSemaphoreW
Key Logger	(FindWindowA, ShowWindow, GetAsyncKeyState) (SetWindowsHookEx, RegisterHotKey, GetMessage, UnhookWindowsHookEx)
Dropper	FindResource ,LoadResource, SizeOfResource

Based on the experiments done, the results for concordance can be seen in Fig. 2. The figure depicts the lists of malware's malicious behaviors and their frequencies observed in the dataset. The highest behavior frequency we observed is Downloader. When downloading free software, user will sometimes be asked to install another component, this technique is called bundled installation, and when the user failed to reject the offer, the malware will be installed on the computer. Likewise, the lowest behavior frequency is Enumerate all process. This may be outstanding to the certain malware which needs to see all processes in the system to find their target. As mentioned before, DF is use in this experiment to show the most commonly used API calls in the dataset, based on the experiments done.

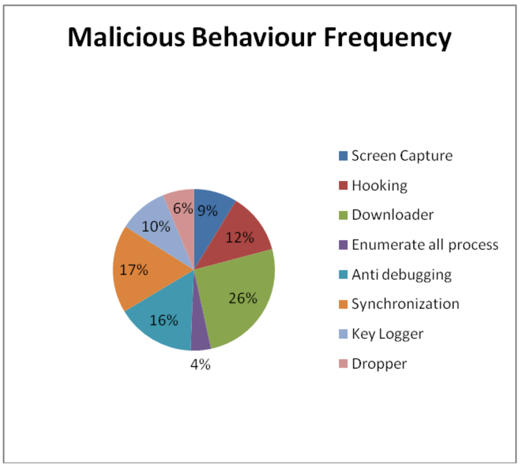

Fig. 2: Concordance Results

DOI: 10.35940/ijeat.F9144.088619

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Discussion

As mentioned earlier, concordance is used to identify the API call sequence from the dataset. The concordance method is easier to use than n-gram because it lists all possible outcomes based on n value, which results a lot of outputs being displayed as compared to this proposed method as this will display results based on the queried keywords. Based on the existing methods if we want to search whether the malware has Anti Debugging, the code the malware allows a malware analyst to run the malware in sequence of steps, initiate changes to memory space, variable values, configurations etc., hence, if debugging is done successfully, it helps the understanding of the malware's behavior, mechanisms and capabilities. For apparent reasons, malware authors would want to prevent. In most cases, the Anti-Debugging process will slow down the process of reverse engineering, but will not prevent it.

If we want to search whether the malware has Key loggers, these will come in two categories—hardware devices and software devices. Hardware devices can be embedded in the internal PC hardware itself, or be an inconspicuous plug in that's secretly inserted into the keyboard port between the CPU box and the keyboard cable so that it intercepts all the signals which we typed. Software key loggers are much easier to introduce to and install on victims' devices. These will inhale out the keystrokes when computer continuously operates. If we want to search whether the malware has Screen Capture behavior, when we enter the keyword as Screen Capture, it will display the circumstance of the Screen Capture in the dataset. Even though by using this method we have to know the API call sequence for the malicious behavior in advance. If we don't know the API call sequence for the malicious behavior, then we can't find for the keyword. These are the disadvantages of the existing ones.

On contrary, based on the results from our proposed experiment, it does provide evidence that this method can be used to search for API call sequence patterns in the dataset. This work can identify eight malicious behaviors from the dataset using this method. Table 8 shows the comparison results for malicious behaviors detected among different methods.

Table 8: Performance Comparison of Malware Behavior Detection Methods

Method	Results (Malicious Behavior Detected)
N-gram Approach	6
Fuzzy Hashing Algorithm Approach	5
correlation and fusion Approach	5
Concordance KWIC With DF-ICF (Proposed approach)	8

Based on the Table 8 above, we can infer that Concordance KWIC can identify much more malicious behaviors Even though N-grams have great capabilities in detecting malware behaviors which gives fewer results as compared to our proposed method. Even though the remaining two approaches Fuzzy hashing algorithm and correlation and fusion are good approaches, but detected fewer malicious behaviors compared to the proposed approach.

We can affirm that this work manages to cover a most important portion of APIs that is considered malicious and we encountered those APIs, we have a mistrust of opinion that the program that uses APIs may be a malware program instead of benevolent. This is because some APIs are dangerous in nature, such as Downloader. If Downloader is used recklessly such as downloading a critical process of a program or information, it may cause loss of data or worse, system failure. This shows how dangerous some APIs as compared to the others.

VI. CONCLUSION

In this experiment, we use concordance tool to identify malicious API call sequence from the dataset. The results are proved that we identified eight behaviors using this method which are Screen Capture, Hooking, Downloader, Enumerate all process, Anti debugging, Synchronization, Key Logger and Dropper. Furthermore, we also use WF to statistically identify which of the malicious or suspicious APIs is favorable by the malware. This study has proven that we can use a concordance to identify malware behavior and based on the results of the experiment, we can conclude that we achieved the purpose of this research. It is proven that this methodology can work for both small and bigger datasets.

REFERENCES

1. Cisco-Annual Cyber security Report 2018, https://www.cisco.com/c/dam/m/digital/elq-cmcglobal/witb/acr2018/acr2018final.pdf?dtid=odicdc000016&ccid=cc000160&oid=anrsc005679&ecid=8196&elqTrackId=686210143d34494fa27ff73da9690a5b&elqaid=9452&elqat=2
2. Business Advantage: The State of Industrial Cyber security 2017.https://go.kaspersky.com/rs/802-IJN-240/images/ICS WHITE PAPER.pdf.
3. Ghazvini A, Shukur Z. "Review of Information Security Guidelines for Awareness Training Program in healthcare Industry", IEEE. 2017;1–6
4. Manap NA et. al., "Cyberspace Identity Theft: The Conceptual Framework", MJSS. vol. 6, no. 4, pp. 595–605, 2015.
5. Dr.G.S.N.Murthy, "An Effective Hybrid approach to Detect DDoS Attacks through Network Anomaly using Enhanced K-Medoids with SVM classification", Handbook on computer and Information Technology, ISBN:978-93-8737425-6.
6. Pektaş A, Acarman T. "Malware classification based on API calls and behavior analysis". IET Inf Security, 2017.
7. Lim H. "Detecting Malicious Behaviors of Software through Analysis of API Sequence k-grams", Comput Sci Inf Technol.Vol. 4, no.3, pp.85–91, 2016.
8. Ki Y, Kim E, Kim HK. "A novel approach to detect malware based on API call sequence analysis", Int J Distrib Sens Networks. 2015.
9. Sanchit Gupta et.al, "Malware Characterization Using Windows API Call Sequences", riverpublishers.com, July,2018
10. HanWeijie et. Al., "Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics", Computer & Security, Volume 83, June, 2019.
11. Anishka Singh et.al, "Malware analysis using multiple API Sequence Mining control Flow Graph", arXiv.org.
12. Bowker L. Corpus "linguistics is not just for linguists: Considering the potential of computer-based corpus methods for library and information science research", Library Hi Tech. 2018.
13. Puneet Goswami, et.al, "The DF-ICF Algorithm- Modified TF-IDF", International journal of Computer Applications., Vol.93, No.13, pp.28-30, May, 2014.

AUTHORS PROFILE

Dr. G S N Murthy is working as a Professor of CSE in Aditya College of Engineering, Surampalem. He completed his Ph.D (CSE) in Rayalaseema University, Kurnool, India. He has 23+ Years of Experience and 5+ years of Research experience. He published various research articles in reputed International Journals and Conferences. He is reviewer for Various Scopus indexed journals and Editorial Board member for Research India group of Journal. He is the member for various professional bodies like IEEE, CSI, and IAENG. His research work focuses on Data Mining, Image Processing, and Cyber Security.

M.V.V.Chowdary is working as a Lecturer in Computer Science, V.S.M.College, R.C.Puram, India. He completed his M.Tech (CSE) in JNTUK, Kakinada. He has more than 20+ Years of Teaching Experience. His research work focuses on Data Mining, Cyber Security.

Dr.M.V.Sangameswar is working as a Professor of CSE in Godavari Institute of Engineering & Technology, Rajahmundry, India. He completed his Ph.D (CSE) in Rayalaseema University, Kurnool, India. He has more than 20 Years of Teaching along with 5+ years of Research Experience. He published various research articles in reputed International Journals and Conferences. His research work focuses on Data Mining, Cyber Security.

Dr.T.P.R.Vital is working as Associate Professor in Department of Computer Science and Engineering, Aditya Institute of Technology and Management (AITAM), India. He completed his Ph.D (CSE) in GITAM University of A.P, India. He has more than 20 years of teaching and 5 years of research experience. He is a member of ACM, ICSES, and ISTE. He has published more than 30 research papers in reputed international journals including SCOPUS indexed and a conference including Springer, Elsevier. He is reviewer of reputed journals like Springer, Elsevier and IEEE. His research work focuses on Machine Learning, Deep Learning, Data Mining, Big Data Analytics, IoT, Computational Intelligence, Voice Analysis, Voice Processing and Bioinformatics.

.