Malware Behaviour Analysis and Classification via Windows DLL and System Call

Mohammad Afiq Amirul Abdul Rauf, Syed Muhammad Hazry Asraf and Syed Zulkarnain Syed Idrus
1Universiti Teknikal Melaka Malaysia
2School of Human Development and Techno-communication, Universiti Malaysia Perlis

a hazrtsyed@gmail.com; b syzul@unimap.edu.my; c*

Abstract. Malware is refers to programs that purposely exploit computer systems' vulnerabilities for harmful purposes. It may be categorized by identifying whether it needs a host program to function and whether it makes copies of itself. Malware is an instance of malicious code with the purpose to disrupt the function of system and has potential to destruct a computer or network [1]. Nowadays, computer malware has become more sophisticated, using advanced code obfuscation technique to resist antivirus detection. Classification of malware samples plays an important part in building and maintaining security. The style of a malware classification system capable of supporting an oversized set of samples and adaptable to model changes at runtime is needed to spot the high range of malware variants. It is a supervised learning technique that created a proper model that may justify class and sort out information into correspondent options based. The advantage of classification is it acceptable of the many individuals since heap of researcher using classification. Next, classification method is quicker and it more accurate.

1. Introduction
Viruses, worm, Trojan and bots are all part of a class of software called malware [2]. The overwhelming majority of malware is created to make money illegally often by stealing sensitive and confidential information from victims computer and mobile devices. Malware also known as "malicious software" that have been widely spread in the network and internet nowadays have several of type that can be classified base on what are their type and behavior. In order to differentiate and distinguish the true unique of the malware type, it is the best way to classified it so that differences can be easily seen. Classifying and distinguishing the malware each of it is the most important thing to do as it can make a better understanding for user to know how the malware can affect computers or devices, how to protect their devices from malware and what action that they will take after being infected with the malware. To differentiate the malware, the suitable experiment will be based on how the intrusion detection system works. Intrusion detection system (IDS) is a software application or devices that used for monitoring packet that occur in their network traffic and detecting the malicious software or malware to send alert for user to make action to that particular malware [3]. As there are various types of IDS, the most common type of it was network intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS). The difference of both type of IDS are the NIDS were placed at the most strategic point or points within the network to monitor traffic to and
from all the devices that are connected to the network. While HIDS run individual hosts or devices on
the network such as monitors only the inbound and outbound packets from the device only and will
alert the user or administrator if suspicious activity were detected. The IDS have two different
detection methods that are signature-based and anomaly-based.

Signature-based IDS refers to the detection of attacks by looking the specific pattern, such as byte
sequences in network traffic, or known malicious instruction sequences used by malware [4]. While
anomaly-based were primarily used to detect unknown attacks, in part due to the rapid development
of malware. The basic approach of this research is by using the anomaly-based detection method is to use
the machine learning to create a model of trustworthy activity, and then compare the new behavior
against this model. Machine learning is a data analytics technique that teaches computers to do what
comes naturally to humans and even animals: learn from the experience [5]. It has algorithms that can
be used for computational methods to "learn" information directly from data without relying on a
predetermined equation as a model. Thus, the rules of the IDS for detecting malware can be regularly
update because of malware is constantly changing day by day.

Machine learning uses two types of techniques in order to get the result that is supervised learning
and unsupervised learning [6]. Supervised learning is when the trains involved the model on known
input and output data so that it can predict future outputs such as clustering, and unsupervised
learning, was finds the hidden patterns or intrinsic structures in input data such as classification and
regression. This research will be focusing on the classification for the technique of the supervised
learning that will predict the discrete responses such as predict an email whether it is a genuine or a
spam. The models classify input data into categories. Classification was use based on the result that
can be tagged, categorized or separated into specific groups or classes. The algorithms of this research
that will be used to perform the classification part will be Naïve Bayes, Decision Tree, Random
Forest, support vector machine and JRip using the Weka application that was installed. The data from
the Cyber Security Malaysia shows a significant growth and the total of cyber incident in past ten
years [7].

However in machine learning, classification itself is the problem of identifying which of a set of
categories a new observation belongs [8]. As the number of malware increases uncontrollably
nowadays, the signature-based detection has difficulty in detecting the malware type effectively as the
malware itself has various types of it and signature that malware might not be created yet when it
starts to attack. Due to this problem, the malware can evade and the IDS use cannot be detect by using
the current detection technique. After doing some research, the finding was approving to have a lack
of further analysis on malware behavior. Thus, a framework to analyze, detect and categorize all the
behavior of the malware in real-time are needed.

2. Objectives
The objective of this research is to analyze the behavior of several type of the malware that occur so it
can be known. Besides, after knowing the behavior, the aim of this research is to propose a framework
for malware classification using a hybrid detection to detect the malware behavior [9] more accurately
and propose an analysis on Dynamic Link Libraries (DLL) and System Call (SYSCALL) in early
stage for high detection.

3. Methods and Analysis
There are several steps taken to conduct the research, among which are data collection, analysis design
and testing. The data collection activity includes collecting the malware base on their preference. The
researcher managed to collect all of the malware in the web page, https://www.malwares.com/. All of
the malware must be analyze base on the “#conficker” preference. The malware will then be analyze
first in the https://virusshare.com/ To be precise and more accurate, the malware will be search only
for three type that is A, B and C as they are common and keep on growing in current day. The
malware that related with the type of A, B and C will be download and store base on their type in
folder to make an easier way to analyze after this phase. The malware is examined to determine
whether it is an abnormal or a benign malware that cannot affect and attack the file when run. The objective of data analysis is to obtain the malware status, label, type and its behavior. The experiment environment is then designed in order to collect the data accurately. Four main tasks will be performed. First, physical network design of the malware analysis [1] in the experiment are created. Second, the logical network design is created to explain on how it connected. Third, the process of analysis is designed. Lastly, the flowchart of the experiment will be produced to represent the algorithm design. The experiment, analysis and algorithm will be produced to represent the algorithm design and it was designed to ensure and confirm that the experiment can be done smoothly and normally. The analysis of this research about classification result based on the performance metrics. The result of performance metrics produce based on the dataset that already collected. Datasets are an integral part of the field in the machine learning. Based on the testing, the result will be analyzed and discussed.

3.1. Analysis and style square measure the very important stage inside the framework advancement for the research.

It makes a distinction to decide on the clear conception of the task which will be created. System analysis is the prime to find out the objectives of a task or its elements. It acts as a retardant finding approach that is significant in upgrading the system and making certain that every single viewpoints of the machine is functioning proficiently. Meanwhile, style framework might be a way enclosed so as to style a novel business framework or to change the trendy framework that's able to meet specific needs appreciate the parts or modules enclosed. The analysis and format for this research is performed which has the gathering of stipulations and closing the planning stage once all the necessity is outlined. In growth, this chapter furthermore center on a way to turn out logs by implies of the usage of Drakvuf and grep the call from the log. Each task requires an activities need to help the change of the occupations and guarantee that the assignments will run effortlessly through its development. The required software and hardware of the research will be discussed in this section. The software required includes the server hardware, Debian, Drakvuf, XenServer, WEKA and Windows 7 (64 bit).

Figure 1. Physical Design

Figure 1 shows the physical style for the research. Xen guest name that is DomU will have up three virtual interfaces selected thereto. The physical interface on the physical host is renamed to “peth0” that is physical eth0. This finds yourself being the transmission out of this Xen physical host, to the physical computer network switch. truly an virtual network cable association is coupled to out of this peth0 to the virtual bridge created by Xen. Virtual Network interfaces with naming program of vif1.0 is established in Dom-0, as network slots for the bridges, and is additionally coupled to the virtual network interfaces (eth0). If Dom-0 wish to speak, it needs a network interface. an virtual network
interface “eth0” is allotted thereto. This eth0 of Dom-0 is then selected applicable information
processing. This eth0 if Dom-0 is coupled to the virtual bridge already created by Xen. The connected
virtual interface on Dom-0 attributable to this an example could also be vif1.0.

Figure 2. Logical Design

Figure 2 shows the logical style concerning this research. It explained however it connected to
every alternative. Xen guest domain is assigned one virtual interface that is Dom0. The physical
interface is assigned one virtual interface that is Dom0. The physical interface is assigned by IP
address 192.168.2.117 meantime virtual interface in Domain-0 is use IP address 192.168.1.130. If
virtual interface wish to speak with the physical interface, it wants the bridge IP that assigned
1962.168.1.90.

4. Conclusions
The experiment started by configuring the hosts and networking. This research requires a computer
that was installed with Debian 8.9 Jessie as an operating system and worm Conficker must be
downloaded and get the sample from the internet. While installing the operating system (Debian), the
storage must be set up as it needs to create a logical volume for the VMs in Xen Server. Logical
volume is needed in virtual machine Windows 7 for storage volume. After done with the volume, then
we need to install Drakvuf for this research to successfully capture the system call log. The result of
the performance metrics produce based on the dataset that already collected. Datasets are an integral
part of the field of machine learning. Major advance in this field can result from advances in learning
algorithm such as deep learning, computer hardware, and less-intuitively, the availability of high-
quality training datasets [10]. This research uses 200 malware samples from ‘contagiodump.blogspot.com’, https://virusshare.com, and https://www.virustotal.com. After collected
200 samples malware, this research needs another sample for normal application and it collect 135 for
normal application. After already got sample for malware and normal application, then capture the log
and produce the dataset because this dataset need to be used to check the classification result based on
the performance metrics.

DRAKVUF is a virtualization based agentless black-box binary analysis system. DRAKVUF
allows for in-depth execution tracing of arbitrary binaries (including operating systems), all without
having to install any special software within the virtual machine used for analysis. In this research,
Drakvuf is used to trace or capture the log of malware worm conficker to identify parameter for
classifying malwares. Identify parameter for classifying malwares means, captures the entire system
call log for worm conficker and generates the dataset. While got the dataset, use it to identify the
malware using decision tree that used in this research which are random forest and J48 algorithm. This
research used to identify and classify what the worm conficker behaviour using Drakvuf. In this
observation, worm is more to signature-based detection because worm scatter or spread itself. The
research is run by develop a script that will run in the Drakvuf to capture the log. This scripting includes source that run Drakvuf, the domain ID for Windows 7 that create in Xen Server, the time that was taken is 660 seconds while Drakvuf is running and the injection of PID which is ‘explorer.exe’ so that it can capture the log based on the scripting.

5. Perspectives
Every project has its own weaknesses and strength, because of it the weaknesses and strengths the study of certain field can continue to develop more and more. Thus, future study for this research is important. There are several things that need to be explored more in the future, especially in the approach itself and experiment. Possible solutions that can be integrated into Drakvuf in the future are it can capture more log than poolmon, syscall and filetracer. Make Drakvuf can automated analysis the malware behavior without using the manual analysis so that it can generated automatic the log.

References
[1] J. Bergeron, M. D. (2001). Static detection of malicious code in executable programs. Int. J. of Req. Eng.
[2] A. Moser, C. K. (2007). Exploring multiple execution paths for malware analysis. In IEEE Symposium on Security and Privacy, 231–245.
[3] Anderson, B. Q. (2011). Graph Based Malware Detection Using Dynamic Analysis. Journal in Computer Virology, 7, 247-258.
[4] Crowcroft, C. K. (2003). Honeycomb – creating intrusion detection signatures using honeypots. In 2nd Workshop on Hot Topics in Network.
[5] Brownlee, J. (2016, April 11). Naive Bayes for Machine Learning. Retrieved from https://machinelearningmastery.com/naive-bayes-for-machine-learning/
[6] Mohamed, A. E. (2017) ‘Comparative Study of Supervised Machine Learning Techniques for Intrusion Detection’, 14(3), pp. 5–10. doi: 10.15546/aeci2014-0021.
[7] DuPaul, N. (2012, October 12). Common Malware Types: Cybersecurity 101. Retrieved from VeraCode: https://www.veracode.com/blog/2012/10/common-malware-typescybersecurity-101.
[8] Kosmidis, K. and Kalloniatis, C. (2017) ‘Machine Learning and Images for Malware Detection and Classification’, Proceedings of the 21st Pan-Hellenic Conference on Informatics - PCI 2017, (December), pp. 1–6. doi: 10.1145/3139367.3139400.
[9] Pirscoveanu, R. S. et al. (no date) ‘Analysis of Malware Behavior : Type Classification using Machine Learning’.
[10] Dinaburg, A. R. (2008). Ether: Malware Analysis via Hardware Virtualization Extensions. Proceedings of the 15th ACM Conference on Computer and Communications Security, CCS’08, Alexandria, 51-62.