A FRAMEWORK FOR M[illegible]SED ON BEHAVIOR

By

MOHAMAD FADLI BIN ZOLKIPLI

Thesis submitted in fulfillment of the requirements
for the degree of
Doctor of Philosophy

September 2012

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
CARO	Computer Anti-virus Researchers Organization
DLL	Dynamic Linked Library
DNS	Domain Name System
FTP	File Transfer Protocol
GDT	Global Descriptor Table
GUI	Graphical User Interface
IDT	Interrupt Descriptor Table
LDT	Local Descriptor Table
LSH	Locality Sensitive Hashing
MSW	Machine Specific Word
OOA	Objective Oriented Association
PE	Portable Execution
ROM	Read Only Memory
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus
VMM	Virtual Machine Monitor

SUATU KERANGKA BAGI PENGENALPASTIAN MALWER BERDASARKAN KELAKUAN

ABSTRAK

Malwer merupakan salah satu ancaman keselamatan utama dalam komputer dan persekitaran rangkaian. Malwer moden mengaplikasi beberapa teknik bagi merumitkan prosedur membendung malwer. Isu-isu malwer semasa seperti serangan hari pertama, teknik elakkan malwer dan malwer dengan ciri-ciri yang dikisar diketengahkan. Tambahan pula, pendekatan yang biasanya digunakan dalam mengawal malwer tidak memberikan penyelesaian yang lengkap bagi mengekang serangan malwer moden. Berdasarkan isu-isu tersebut, satu kerangka baru bagi pengenalpastian malwer berdasarkan kelakuan adalah dicadangkan. Kerangka ini terdiri daripada tiga komponen utama; i) analisis tingkah laku, ii) ramalan malwer dan iii) pengkelasan malwer berasaskan sasaran. Pengenalan tingkah laku mengaplikasikan pendekatan dinamik dengan gabungan *Analisis Masa Larian* dan *Pemantauan Sumber*. Bagi ramalan malwer, empat bidang dari ciri-ciri malwer seperti i) proses, ii) fail, iii) pendaftar dan iv) aktiviti rangkaian digunakan. *Peraturan ramalan IF-THEN* yang dihasil berdasarkan teknik perlombongan data iaitu *Algoritma ID3* adalah digunakan. Dalam perlaksanaan pengkelasan malwer mengikut sasaran, *Peraturan Berasaskan Struktur Berlapis* digunakan untuk mengkelaskan malwer ke dalam kelas yang sepatutnya. Ketiga-tiga komponen utama tersebut bersepadu bersama sebagai satu unit melalui *Pengkalan Data.* Ujikaji terhadap kerangka tersebut menunjukkan bahawa kerangka yang dicadangkan menyediakan penyelesaian yang lebih baik bagi pengenalpastian kelakuan

malwer, ramalan dan pengkelasan sasaran berbanding dengan beberapa teknik yang berkaitan. Daripada hasil ujikaji, ia membuktikan bahawa rangka kerja ini boleh digunakan sebagai salah satu daripada amalan keselamatan bagi menghadapi serangan malwer moden terhadap komputer.

A FRAMEWORK FOR MALWARE IDENTIFICATION BASED ON BEHAVIOR

ABSTRACT

Malware is one of the major security threats in a computer and network environment. Modern malware embeds several techniques in order to complicate malware defence. The current malware issues such as zero-day attacks, malware avoidance techniques and hybrid malware are highlighted. Furthermore, a common approach in malware defence does not provide enough solution to prevent modern malware attacks. Considering the above issues, a new framework for malware identification based on behavior is proposed. This framework consists of three major components; i) behavior analysis, ii) malware prediction and iii) malware target classification. The behavior analysis applies a dynamic approach with a combination of *Run Time Analysis* and *Resource Monitoring*. For malware prediction, there are four areas of malware features which are i) process, ii) file, iii) registry, and iv) network activities. The *IF-THEN Prediction Rules* which is generated using the data mining technique, *ID3 Algorithm* is used. In the implementation of malware target classification, *Structure Level Rules* are utilized to classify malware into possible target class. These three major components are integrated together as a cohesive unit for malware identification through *knowledge storage*. The experiment on the framework shows that as compared to several other related works, this framework provides better solutions on malware behavior definition, prediction and target classification. From the results, it is proven that the framework can be implemented as one of the security practices to counter the modern malware attacks in a computer environment.

CHAPTER 1

INTRODUCTION

1.1 Overview

Malicious Software or *Malware* is one of the major threats in a computer and network system today. In fact, the root cause of most Internet security problems originated from malware threats. Malware comes in a wide range of forms and variations, such as viruses, worms, Trojan horses, backdoors and spyware (Kramer and Bradfield, 2010). Each malware type has specific characteristics and can be classified based on its behavior and spreading mechanisms. Although all types of malware have specific objectives, its main purpose is to break into computers or network operations. Malware will exploit vulnerabilities of software which is running on operating systems. In some cases, the attacker may use social engineering techniques to trap system users into running the malware.

Malware causes billions of losses to computer operations worldwide. Nowadays, malware attacks have become worldwide issues due to the dramatic growth of new unique malware. In year 2010, McAfee (2010) reported that averages of 60,000 of new malware threats were identified daily. The numbers are nearly four times more than 16,000 of new malware detected per day in year 2007. G Data Security Labs (2010) also reported the higher growth of the malware attacks in recent years as shown in Figure 1.1 (Ralf and Sabrina, 2010). Based on the figure, the highest growth of new malware was very serious

since 2008. The situation has occurred due to the technological advancement in malware creation.

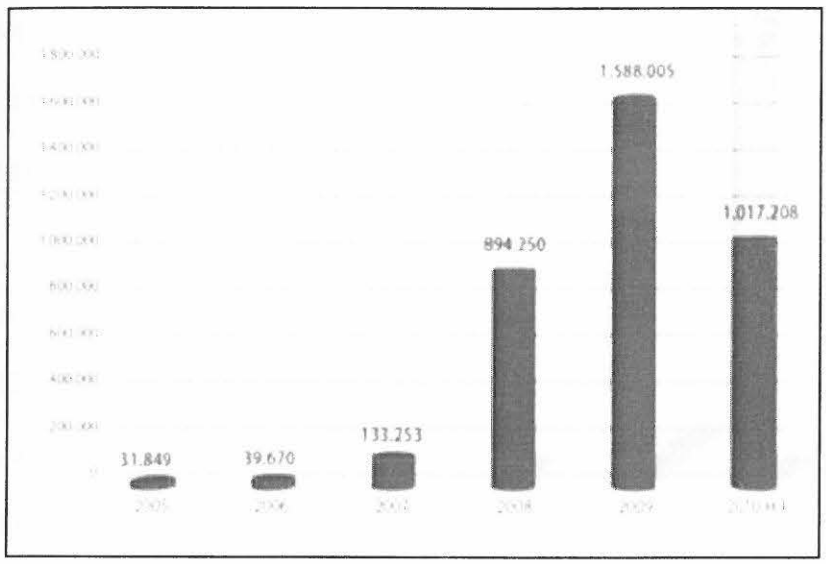

Figure 1.1: The number of Malware Growth since year 2005 until year 2010 (Ralf and Sabrina, 2010)

Malware attacks which are known as zero-day attacks are common these days. Zero-day attacks are previously unseen malware, which are launched by attackers as a first time assault of an unknown malware or a new existing malware (Ye *et al.*, 2009). Basically, these zero-day attacks propagate fast and cause catastrophic damage to the computers or network systems before the new identifying fingerprints are forwarded to end-users. Therefore, the signature-based malware detection using unique byte strings is unable to detect zero-day attacks.

As malware detection and prevention technology improves, malware writers have become more creative in creating hiding techniques to avoid analysis and detection (Christodorescu *et al.*, 2005). Furthermore, malware avoidance techniques that are applied by malware writers are difficult to be detected by using commercial anti-

malware systems that use signature-based techniques. One example of the hiding techniques is by combining the previous behavior to attack and at the same time evading signature-based detection. Other techniques assimilate the existing malware with an avoidance technique. Some commonly used avoidance techniques used by malware writers are file packing, polymorphism, metamorphism, anti-debugging and anti-virtualization (Moser *et al.*, 2007; Igor *et al.*, 2007; Royal *et al.*, 2006).

In general, malware classification is used to cluster malware types into possible groups. However, other common techniques that are used by malware writers to make the malware analysis process become more complicated by creating modern malware with hybrid characteristics. Modern malware created by a hybrid technique has characteristics that belong to several malware families. The modern malware with hybrid charecteristics complicates the malware classification tasks because the malware belongs to multiple malware families (Xiao-Yong and Wei-Wei, 2009). Thus, new possible solutions should be identified in order to solve the issue of malware classification.

Based on the modern malware issues, new innovative solutions in malware identification need to be implemented to minimize the modern malware attacks. The next section will address the motivation of the thesis by covering the limitations on current solutions.

1.2 Motivation

Malware was created for the purpose to damage the host oriented systems. Most of the malware were created to attack host systems because most of the user operations and

information that are located on the host oriented systems (Heiser, 2004). Moreover, the technology advancement causes most of the daily operations are conducted with the use of the computer system. Based on the importance of a computer system in daily life, the host oriented system has become the potential target by malware writers. Malware can infect the host system by targeting various file formats in the system environments such as *boot sector*, *program executable file* and *data file.* In order to prevent the host system from malware infection, sophisticated malware defences should be implemented in the host oriented system environment to minimize the potential damages which may cause by malware.

The most common approach for malware detection in a host oriented system is the signature-based technique. Although this technique is very popular and reliable for the host oriented security tool, limitations of this technique need to be addressed. The main limitation of the signature-based technique is detecting unknown malware which are zero-day attacks. This technique uses unique byte strings but it fails to detect previously unseen malware (Ruili *et al.*, 2008). Nevertheless, as agreed by Hsu *et al.* (2006) and Ye *et al.* (2009) this technique always fails to detect variants of known malware which is defined as polymorphic malware. The signature-based technique also relies on human expertise in creating a signature that represents each of the malware programs. In general, the signature-based technology suffers from two main shortcomings: i) inability to detect zero-day attacks and ii) inability to cope with an exponential growth of zero-day attacks (Mehdi *et al.*, 2009). Therefore, an alternative solution is needed in order to prevent zero-day attacks.

The main requirement in malware identification is malware analysis. Malware analysis is a process to learn and understand malware characteristics by using specific approaches. Two common approaches which can be used to analyse malware are static analysis and dynamic analysis. Static analysis examines the program files without running the source code (Tzu-Yen *et al.*, 2008). Static analysis provides a more comprehensive analysis compared to dynamic analysis because it is not bounded to any specific execution. It analyses the malware program by going through every single line of a binary program. However, static analysis which relies on code tracing is still hard to be implemented due to malware avoidance technique.

Dynamic analysis monitors malware activities through the memory, file, registry and network activities during program execution. When compared to static analysis, dynamic analysis also has a quick approach to analyse malware behavior. In general, recent works on dynamic analysis rely on virtualization architecture in order to create a controlled environment. The controlled environment is an isolated environment that prevents the analysis host from possible damage (Bryan *et al.*, 2008). However, the virtualization based controlled environment is ineffective against malware with an anti-virtualization feature.

Currently, many researchers work on predicting unknown malware by using different strategies such as machine learning and artificial intelligence (AI) techniques. Prediction is a process to classify an unknown program by using generic strategies (Feng and Han, 2010). The motivation on using those strategies is mainly to overcome the limitations of the signature-based technique in detecting unknown malware. Machine learning techniques mimic the functions of a human brain for deriving solutions based on

specific rules. Common machine learning techniques are *Naïve Bayes*, *Support Vector Machine*, *Decision Tree* and *K-Nearest Neighbor*. These techniques contain specific algorithms that are capable to address malware prediction. Nevertheless, the most common issue in malware prediction systems is the amount of *false alarms* that reduces the accuracy of the systems. In order to minimize the false alarm issue, a new solution which based on *IF-THEN* rules will be implemented in malware prediction.

Considering the above issues, this study will focus on several research questions. The problems are formulated from the current issues in malware and the limitations of malware defence as shown in Figure 1.2. The specific research problems are as follows:

i. How to learn and understand an unknown malware binary?

ii. What are the possible features in malware binary that can be used in predicting unknown malware?

iii. What is the possible technique to predict malware based on a binary file?

iv. How to classify the identified malware binary into a possible group?

v. How to evaluate the proposed solutions in the malware behavior-based identification?

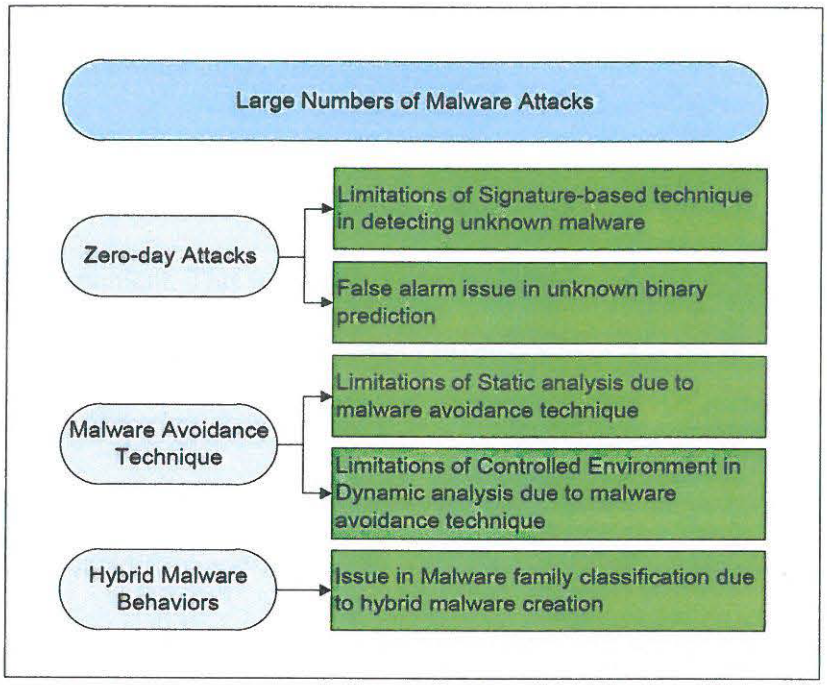

Figure 1.2: The Current Issues in Malware and the Limitations of Malware Detection and Prevention

1.3 Goal, Objectives and Scope

The main goal of this study is to improve the malware identification process in host oriented system. To achieve the goal of this research, the works are divided into three main following objectives:

i. To investigate different characteristics of unknown malware by using a combination of *Run Time Analysis* and *Resource Monitoring* in order to define malware behavior and select useful malware features for prediction and classification processes.

ii. To predict unknown malware using *IF-THEN Generated Rules* based on data mining techniques for predicting future malware.

iii. To classify malware into specific classes based on *Structure Level Rules* in malware target classification.

The scope of this study is on the host oriented malware behavior identification in a Windows environment. This is because most of malware attacks target Windows-based operating system. In fact, Ralf and Sabrina (2010) reported that 99.4% of new malware inflict Windows operating system. Thus, this study will only focus on host oriented malware in Windows-based operating systems. The process of analysing the malware program is conducted in the Windows operating system with a controlled environment setup. The controlled environment also includes client and server host that is directly connected in an isolated environment.

Another part of this research involves the use of a database for malware prediction and target classification processes. The database is known as *knowledge storage* which is used to store malware profiles that are created during the behavior analysis process. The datasets in forms of Windows portable execution (PE) files of the malware program will be used in malware behavior-based identification.

1.4 Methodology

In order to develop a framework for malware identification based on behavior, the research processes are divided into five main stages. The stages are i) data collection and analysis, ii) behavior analysis, iii) malware prediction, iv) malware target classification, and v) evaluation. Figure 1.3 shows the process flow of this research.

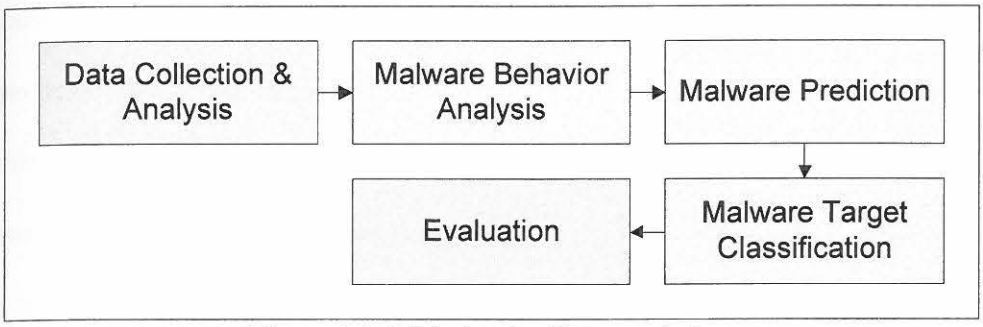

Figure 1.3: Main Task of Research Process

Data collection and analysis are very important tasks at the beginning of this research process especially in collecting malware programs from Windows PE files. The task is to collect unknown malicious binaries from the respected malware database. The dataset which is used in this study is based on the standard data that is commonly used in in malware research by Malheur (Rieck *et al.*, 2011). The binaries corresponding to the malware originated from a variety of sources. The data collection and analysis task has two purposes. The first purpose is to understand the structure and content of the current malware program. In order to fulfill the research needed, several malware from different families are selected. The second purpose is to use these data collections in the implementation and testing phase of the framework. Hence, it is used to evaluate the workability and accuracy of the proposed solutions in the framework.

The first objective can be achieved by investigating different characteristics and operations of malware. The first task is by implanting a controlled environment. In this study, the controlled environment is based on a real operating system which is used as an alternative to the virtual controlled environment. The main purpose of this controlled environment setup is to provide a trusted controlled environment for the host oriented malware behavior analysis especially to prevent anti-virtualization malware. Conducting