Ain Shams University
Faculty of Engineering
Computer and Systems Engineering Department

Infection Immune System

by

Ahmed Osama Hasan Abo El-Mal
B.Sc., Electrical Engineering
(Computer and Systems Engineering Department)
Ain Shams University, 2008

A THESIS
SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE IN ELECTRICAL
ENGINEERING

DEPARTMENT OF COMPUTER AND SYSTEMS
Supervised By

Dr. Ayman Mohammad Bahaa Eldeen Sadeq
Dr. Mohamed Ali Sobh

Computer and Systems Engineering Department
Cairo, 2014
Cairo, Egypt

Ain Shams University

Faculty of Engineering

Computer and Systems Engineering Department

Examiners Committee

Name : Ahmed Osama Hasan Abo El-Mal

Thesis : Infection Immune System

Degree : Master of Science in Electrical Engineering

Name, Title, and Affiliate	Signature
Prof. Dr. Nawal Ahmed El-Fishawy Electronics and Electrical Communications Department Faculty of Electroni Engineering, Menoufia University, Menouf, Egypt	.....
Prof. Dr. Hani M. Kamal Mahdi Computer and Systems Engineering Department Faculty of Engineering, Ain Shams University, Cairo, Egypt	.....
Dr. Ayman M. Bahaa-Eldin Computer and Systems Engineering Department Faculty of Engineering, Ain Shams University, Cairo, Egypt (Supervisor)	.....

Date: / /

Abstract

Ahmed Osama Hasan Abo El-Mal

Infection Immune System

Masters of Science dissertation

Ain Shams University, 2014

Internet provides a comfortable environment for malwares to spread faster. This poses a great threat on individuals and companies. Becoming not bound to only professionals, malware writing imposes a huge burden on anti-virus labs in malware analysis for signature extraction. Also, the introduction of obfuscation techniques makes the malware signature extraction even harder by using static analysis only. Dynamic analysis on the other hand proves more robust in defeating the different obfuscation techniques where the malware is being analyzed at its runtime. Many researches are also addressing the automation of the analysis for better response to malware introduction in the wild and less error prune. Different researches address the different challenges that are possessed while trying to automate the analysis process.

In this thesis, we present a novice automated malware analysis system. The devised system introduces solutions for different challenges in the whole analysis process. We present an advanced interception technique specifically designed for malware monitoring. Added to that a stealth controlled environment for better malware behavior monitoring. And an enhancement to a machine learning engine that automatically detects unknown malwares based on previous knowledge.

Keywords:

Malware Analysis, Behavioral Analysis, Dynamic Analysis, Malware Detection, Malware Types, Malware, Virus Detection, Malware Monitoring, Code Interception.

List of publications

Abo El-Mal, A. Osama, M. Ali Sobh, and Ayman M. Bahaa Eldin. "Hard-Detours: A new technique for dynamic code analysis." EUROCON, 2013 IEEE. IEEE, 2013.

Acknowledgements

First, I would like to thank ALLAH for his great support to me in accomplishing this work.

I would like to express my gratitude to Dr. Ayman Mohamed Bahaa El-din for his encouragement to me on continuing this work.

I would like to express my gratitude to Dr. Mohamed Ali Sobh for his leading effort in the development of different parts of this work from the technical development to the documentation work.

I would like to express my deepest gratitude for my dear mother for here great support and encouragement. Without here this work would have never developed or come to real. I also want to thank her for saving my time even if this was at the expense of here comfort.

I would like to thank my father for everything. Without his efforts I would have never became what I am today.

I would like to thank my wife for her encouragement and for providing me with the suitable environment to complete my work.

I would like to thank my sister. I depended on her a lot to complete many of my homework towards the family to be able to develop my Masters.

At last but not least, I would like to thank my friend Ahmed Alaa El-Din for our technical discussions that helped me a lot in developing this work.

Statement

This dissertation is submitted to Ain Shams University for the degree of Masters of Science in Electrical Engineering (Computer and Systems Engineering)

The work included in this thesis was out by the author at Computer and Systems Department, Ain Shams University.

No part of this thesis has been submitted for a degree or qualification at other university or institution.

Date : 18 / 10 / 2014

Signature :

Name : Ahmed Osama Hasan Abo El-Mal

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Abbreviations and Nomenclature

PUA:	Program under Analysis
PE:	Portable Executable Format
MA:	Monitor Application
AUT:	Application under Test

Chapter 1 Introduction

Malware is used to refer to all the types of harmful programs attacking computers [1]. The term is a general term including trojans, adware, spyware, worms, viruses, rootkits, ransomware, malicious Browser Helper Objects and botnets. As can be seen malware is a broad band of malicious program types.

Malware programming starts from a proof of programming skills to be a job and a way of gaining money and even more as weapons to attack enemies. Some viruses and like are specially targeted to certain companies and users to leak important data. Victim computers in Botnets on the other hand are rented to maybe launch a wide range attack on a site. Many malicious actions can be accomplished by different types of viruses.

From boot-sector early experimental viruses in the 1979 to the modern worms like Code Red, Slammer and Nimda [2], malwares have grown to be fast and aggressive in actions. Also, now not only powerful assembly computer programmers can write sophisticated viruses, but also novice programmers can write ones with the aid of the virus generation kits.

In 1990, members of the "Verband Deutscher Viren liebhaber" introduced the first virus generation kit VCS (Virus Construction Set) [3]. After that, other generation kits were introduced to the market like GenVir, PCL, PS-MPC and others. The introduction of those kits adds more burdens on antivirus researchers.

The currently common, with minimum false positives, way to detect virus is by signature comparison. Human researchers try to reverse engineer the virus code and behavior to capture a unique sequence of instructions which can be published as a signature. If a known signature is found in any application on the machine, this application is considered infected.

The process of reverse engineering the virus and the extraction of a suitable signature is called malware analysis. In accomplishment for this process many tools are developed to aid the researchers in their job, increasing the throughput to catch the virus generation train. The developed tools are based on two major analysis techniques: Static code

analysis [4] and Dynamic code analysis [5] (sometimes called behavioral code analysis). Not only tools, but also researches are addressing some proposals to fully automate the whole analysis and detection process to either old known family malwares or new malwares.

Static code analysis is based on offline exploration of the virus code without running it, predicting the behavior of the virus at runtime. This technique is safe and enables the researcher from close looking at all paths of the code, but cannot escape the traps of obfuscation [6] and packing of virus code.

Dynamic code analysis on the other hand is based on runtime exploration of the virus code. The virus is left to run and its behavior is recorded for further provision of the researcher. Although this technique is a little bit risky from the point that the virus is left to run on the analysis system, it is very promising and is not affected by obfuscation and code packing because at some point of time, the virus code should be clear in memory for execution, and this point is where its signature is really available. Also the behavior of the malware itself can be considered as a signature instead of just the sequence of bytes.

Different research paths were taken to make use of the strength of dynamic analysis. Some researchers make use of virtual machines to analyze the virus host application [7]. Although this type of tools are promising in protecting the detection system from being evaded itself by the virus, it is un-shielded against counter techniques developed by malware writers which detect the virtual machines and hide their actual malicious behavior [8] [9]. Also, those tools miss an important part in the analysis process which is the interaction of the virus with the surrounding applications running on actual systems while being in the wild on a real user machine. In addition to the introduction of the semantic gap resulting from the difference in the monitored information level (only memory and hardware objects can be seen from outside of the guest system on virtual machines).

Other researchers try to Sandbox the application using API interception techniques [10]. Which itself split into different techniques including inline API interception, SSDT (System Service Description Table) interception and IAT (Import Address Table) interception. This provides the analysis process with a more realistic taste of being on a real machine

that has other programs running. And to protect the host system and the analysis system from being attacked, tools using this technique usually block the PUA from writing to the system files [10] and try to decide if the PUA is infected or not before it crashes or fails to complete execution due to the unsatisfied write requests issued. As can be depicted, malware writer can embed a code to check if its actions are fulfilled or it is being faked.

Solving the running of malware code part of the problem, many researchers try to automate the whole process. This includes other areas of investigation. The usage of machine learning algorithms and neural networks are good targets for research. This includes problems like finding a good representation of the data and the best technique that has high rate of detection, low false positives and negatives, all this with reasonable performance.

In this thesis, solutions for problems in different areas in the automated malware analysis flow are presented as a complete system for automated malware analysis. The system addresses different areas from a robust stealth sandbox to a machine learning part that all interact together to detect previously unknown malwares. The sandbox presented is designed to be strictly bound to the program under analysis and defeat any attempt from the malware to bypass the detection mechanism. In addition, the sandbox implements an advanced algorithm that simulates the actions being requested by the program under analysis without actually applying those actions on the host. After collecting the required analysis data from the sandbox, the system uses this data to automatically analyse the behaviour of the program under analysis and classify unknown malware samples to clusters previously build for malwares having similar behaviours.

The rest of the thesis is organized as follows. Chapter 2 is a background on the malicious code problem as a whole. After that chapter 3 presents a novice technique that is specifically designed for malware analysis to provide a robust interception system that is not easily defeated by malwares. Then chapter 4 presents a controlled environment (usually declared as Sandbox) that utilizes the technique proposed in chapter 3 to stealthily monitor malwares during execution.. Chapter 5 describes a proposed change to an automated malware analysis system. The change is