UNIVERSITY OF BIRMINGHAM

Behaviour-based malware detection using neural networks

Supervisor: Mohan Sridharan

Author: Ángel Alberto Hamilton López

Student ID: 1972652

MSc Cyber Security

School of Computer Science, University of Birmingham

Birmingham, September the 3rd of 2019

Acknowledgements

Parents

Olivia

Friends and family

Mohan Sridharan

Abstract

The objective of this project was to research and evaluate the use of neural networks to detect malware based the behaviour of the software.

Typical anti-malware software relies mostly on signatures and other methods of static analysis, which is only effective against already known malware and is much less useful against polymorphic malware and first-day attacks. The common denominator of all malware is that it behaves maliciously so having a detection system based on behaviour would potentially identify any malware as it executes, regardless of it is known, unknown or polymorphic.

For this project we developed a tracing software for windows using Event Tracing for Windows. This software, given an executable file, executes it and generates a log files with certain system calls done by the executed program. These logs are then processed and fed into neural networks to train them into being able to distinguish logs from a malware program or a normal software.

Even though the results of our tests were not as successful as we expected, there is still a lot to be learnt from this research.

Keywords: Neural Networks, Malware, Machine Learning, Event Tracing for Windows.

Contents

Chapter 1: Introduction

The aim of this project is to research the use neural networks and Event Tracing for Windows [1], also known as ETW, as tools to create a functional behaviour-based malware detection software. The final product was not developed in the end, but the research done in this area has provided interesting insight in this area of research.

Malware, short for malicious software, is defined as “Any software designed to cause damage to a single computer, server, or computer network” [2]. Under this definition we find a lot of different types of software like viruses, trojans or ransomware, each of which has different intentions and different strategies to attack the computer. The common denominator of all malware is that it tries to do something damaging to the machine it is running on. Traditionally, anti-malware software has relied on signature detection and heuristic methods, but with 4.4 new malware programs being created every second [3] it is impossible for anti-virus companies to keep up to date on all the new possible signatures and attack patterns. For this reason, a software able to identify a malicious activity within a system without requiring previous knowledge of the specific malware would be a valuable tool for any machine or network.

Neural networks [4] are a popular type of computing system inspired by biology and try to imitate the behaviour of the cells in the brain to learn without being given specific instructions. Neural networks are designed to receive some input values and then produce an output based on the information received. Neural networks are supervised machine-learning systems and as such are trained with a set of inputs and expected outputs and by trial an error the network slowly learns and adapts. After training, the network can be used to make predictions on new input values. Thanks to their properties, neural networks are used in very different tasks like natural language processing [5], face recognition [6] or self-driving cars [7].

Being a research project, the objective was not to develop a running software, but instead try out different possibilities to train a neural network able to distinguish malware from normal programs. We tried different approaches to the processing of information and the initial parameters of the neural networks in to see if this combination of tools (ETW and neural networks) is viable to make a functional system. Even though the results were negative, there is a lot we learnt from this research which sets up the grounds for future work.

In the second chapter we will talk about background information in more detail, as well as other related works that tackle this problem from different angles. In chapter 3 we detail the methodology followed during the project and explain the tracer program and the neural networks used in detail. Chapter 4 contains our thoughts on the project, the achievements and potential improvements to keep working in this line in the future. Finally, chapter 5 is the conclusion and summary of the project, after which we have the references and appendixes.

Chapter 2: Background

Chapter 3: Methodology

Chapter 4: Discussion

Chapter 5: Conclusion

References

[1] Microsoft, “Event Tracing,” 31 5 2018. [Online]. Available: https://docs.microsoft.com/en-us/windows/win32/etw/event-tracing-portal. [Accessed 3 September 2019].

[2] Microsoft, “Defining Malware: FAQ,” 01 04 2009. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/tn-archive/dd632948(v=technet.10). [Accessed 3 September 2019].

[3] AVTest, “AVTest Security Report 2018/2019,” 2019. [Online]. Available: https://www.av-test.org/fileadmin/pdf/security_report/AV-TEST_Security_Report_2018-2019.pdf. [Accessed 3 September 2019].

[4] S. Haykin, Neural networks: a comprehensive foundation., Prentice Hall, 19994.

[5] J. W. Ronan Collobert, “A unified architecture for natural language processing: deep neural networks with multitask learning,” no. ICML '08 Proceedings of the 25th international conference on Machine learning, pp. 160 - 167, 2008.

[6] S. Lawrence, “Face recognition: A convolutional neural-network approach.,” *IEEE transactions on neural networks,* vol. 8, no. 1, pp. 98 - 113, 1997.

[7] T. Tian, “Deeptest: Automated testing of deep-neural-network-driven autonomous cars.,” *Proceedings of the 40th international conference on software engineering.,* pp. 303 - 314, 2018.