Chapter 3

Deep-Learning and Machine-Learning-Based Techniques for Malware Detection and Data-Driven Network Security

Praneeth Gunti
National Institute of Technology, Kurukshetra, India

Brij B. Gupta
National Institute of Technology, Kurukshetra, India

Francisco José García Peñalvo
https://orcid.org/0000-0001-9987-5584
University of Salamanca, Spain

ABSTRACT

A never-ending fight is taking place among malware creators and security experts as the advances in malware are daunting. The machine learning strategies are indeed the new mode of researching malware. The purpose of this chapter is to explore machine learning methods for malware recognition and in general deep learning methods. The chapter gives complete explanations of the techniques and resources used in a standard machine learning process for detecting malware. It examines the study issues that are posed by existing study methods and introduces the potential avenues of study in future. By administering a study to the participants, scholars have a better knowledge of the malware detections. The authors start by discussing simple dynamic modelling methods, their importance to the data analytics of malware, and their implementations. They use open access resources such as virustotal.com that review sample of dynamic analysis in reality.

DOI: 10.4018/978-1-7998-7789-9.ch003

INTRODUCTION

Throughout this section, you will understand the fundamentals of complex malware detection. Unlike static research, which relies on what kind of malware seems like in document format, the dynamic analysis comprises executing malware in a secure, confined area and then determine how it acts. It would be like putting a harmful bacterial strain over to an enclosed area to see the impact against many cells. Utilising dynamic analysis, we may switch through typical static analysis obstacles, like packaging and deception, and also obtain more precise visibility into the intent of a specific malware test. Security and Privacy is needed everywhere like in web applications (Gupta 2016), IoT devices (Tewari, 2017)(Ab Malek, 2016), cloud computing (Al-Qerem, 2020), routing protocols (Jerbi, 2020) in WSN etc., so that our data and the transactions are protected.

Does Dynamic Analysis Needed?

To grasp why dynamic analysis need, remember the topic of packaged malware. Recognise that packaging malware relates to compacting or misrepresenting the x86 base source code of the software to conceal the deceptive existence of the application. The bundled malicious code unwraps its own as it attacks the target computer of that kind by which the program could be executed. We can attempt to dismantle a sealed or abstracted malicious payload utilising static detection methods; however, it is a tedious method. For instance, through static testing, we would first have to identify the position of the ambiguous script within the malicious document. After that, we will need to locate the deception functions that will successfully de-obfuscate this script to be executed. After finding the macros, we would discover why this de-obfuscation technique operates to execute that on the program. And afterwards, will we start the actual method of reversing the malicious script. An easy but intelligent solution to this method is running the malicious program in a secure, enclosed atmosphere termed a sandbox. Testing malicious code in a sandbox helps this one disassembles its own as it does before harming a primary target. By merely executing malware, we will figure out how many servers a certain binary malware binds to, which machine configuration settings it adjusts, as well as which Input/Output interface it is attempting to do.

Resources for Dynamic Malware Identification

Dynamic modelling is valuable for both malware processing and malware reverse engineering. Since dynamic modelling shows how a malicious test performs, we may equate its behaviour with many certain malware mixtures. For instance, since dynamic analysis reveals which documents the malware suspects send to the filesystem, we could use this information to link the malicious documents that write related config files to the disc. These kinds of hints allow one to categorise malicious files focused on general characteristics. It will also support us in recognising malicious files created by similar organisations or are members of common groups. Most notably, dynamic processing is valuable for creating machine-based malware indicators. We may prepare a detector to differentiate between harmful and benevolent clones by analysing their actions through dynamic study. For instance, by analysing millions of complex review reports from either malware or innocuous archives, a machine learning algorithm will understand that if msword.exe executes a program called powershell.exe, this behaviour is harmful; however, while msword.exe runs Browser, it is pretty safe.