CSC3002

Problem Description & Work Plan

Malware Categorisation using machine learning

jon kyle - 40106763

2016

# Problem Description

Malware categorisation techniques play a critical role in safeguarding existing technical infrastructure. Malware comes in many forms and uses many different attack vectors to gain access to computers in order to cause undesired and potentially harmful behaviours to occur. Attacks can often cause significant financial harm to companies and in some cases individuals.

The symptoms of malware differ, depending on the malware type or family; e.g. a root-kit will exhibit different behaviours to ransomware. Some forms of malware are also easier to detect than others; Ransomware being a prime example of a blatant attack which does little to hide itself from the user, and on the opposite end of the spectrum you find Trojans and Rootkits which are extremely hard to detect.

As the symptoms differ, so does the potential risk to systems infected by the malware. Categorisation allows us to make an approximation of the potential risk a file poses. Though there are many existing anti-virus scanners available, the results of these scanners can disagree when it comes to the categorisation of a particular piece of malware, and in some rare cases an anti-virus may classify a file as safe, when in fact it is not.

There is likely no perfect way of categorising malicious files, but the goal of many Security Operation Centres is, to quickly and accurately assess the risk associated with a malicious file so that the necessary mitigations and remediation can be put in place.

For this project we will be focusing on static analysis of the malware file structure in order to determine its category. Our goal is to investigate this new method of malware categorisation, which will data-mine a set of Windows PE files and make use of existing machine learning libraries and clustering algorithms.

# Work Plan

## Overview

Some initial research has been carried out to determine which features will prove useful when extracted from the data-set. Some examples of useful information include imports, exports, date-time stamps, the number of sections in the file and their sizes on disk and in memory, the subsystem of the PE file (GUI or command line) and the resources it contains [1].

As part of my initial research I have making use of tools such as PEView which allow me to explore the PE files in a GUI to familiarise myself with the file structure of the PE format.

The first step in implementing the requirements of the project will involve creating a parser, in C++, Java or Python, which is able to read the PE file headers and extract appropriate features, which will aide in the clustering, and therefore categorisation at a later stage.

Following the feature extraction, the data will need to be marshalled to a suitable format for the, as of yet undecided, machine learning library to process. Some initial online research into what libraries are available has revealed a few candidates for the machine learning library: mlpack(C++), scikit-learn(Python), and H20(Java) as well as others.

The next stage will be representing the data in a meaningful way, possibly using existing graphing libraries. These graphs should clearly show how the data has been clustered and allow an approximate evaluation of the category of a malicious PE file based on where it falls on the graph. E.g. when a file is data-mined and its features added to the data set, if it is clustered with files which are known to be root-kits, then it is likely that this file is also a root-kit and poses a significant risk.

The project will also include the implementation of an algorithm to measure the difference in existing labels given to the data by the algorithm, to those which are determined after the data-set has been expanded and clustered again. I.e. if we doubled the size of our data-set, how different will the category definitions be to those which were determined before the data-set increased in size.

## Milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start Date** | **Finish Date** | **Duration(days)** |
| Background Research + Defining Features | 10/10/2016 | 01/11/2016 | 22 |
| Implementation of Parser | 28/10/2016 | 18/11/2016 | 21 |
| Implementation of data marshalling | 18/11/2016 | 21/11/2016 | 3 |
| Creation of clustered data using machine learning libraries | 21/11/2016 | 01/12/2016 | 10 |
| Graphing of clustered data | 01/12/2016 | 10/12/2016 | 13 |
| Implementation of algorithm for measuring the difference in new and existing labels | 10/12/2016 | 01/02/2017 | 49 |
| Project Refinement and Improvement | 01/02/2017 | 31/03/2017 | 58 |

## Gantt Chart

# References

[1]M. Sikorski and A. Honig, *Practical Malware Analysis*. San Francisco: No Starch Press, 2012.