Runtime Execution Introspection for Security Protection Using Machine Learning

Researchers

Prof. Shlomi Dolev
shlomidolev@gmail.com

Mohammad Ghanayim
ghanayim@post.bgu.ac.il

Dr. Alexander Binun
binun@cs.bgu.ac.il

Dr. Sergey Frenkel
fsergei51@gmail.com

Prof. Yeali S. Sun
sunny@ntu.edu.tw

Publications

S. Dolev, M. Ghanayim, A. Binun, S. Frenkel and Y. S. Sun, “Relationship of Jaccard and Edit Distance in Malware Clustering and Online Identification,” IEEE International Symposium on Network Computing and Applications, NCA, 2017.

Goals

The goal of this project is to learn and analyze the behavior of labeled traces of API calls and build classifiers based on that analysis to be executed in hypervisor environments in order to detect intrusions and threats.

Description

This research is part of a joint project with Prof. Sun from National Taiwan University, funded by the Israeli Ministry of Science and the Taiwanese National Science Council. We used machine learning to perform behavioral analysis and classification of malwares, based on traces of their calls to OS APIs; we first trained a classifier by clustering data consisting only of malicious API call traces, based on the Jaccard index. Then, during the prediction phase, a query trace is labeled as either benign or malicious by its distance or similarity from the medoids of the clusters, (i.e., a query trace is labeled as benign if and only if it is anomalous to all clusters), which proved to be 90% accurate in our experiments. To perform efficient clustering, we resorted to techniques used in data mining, while keeping our solution platform-independent by disregarding the semantics of API names and arguments. Similar traces were grouped together by a method of Locality Sensitive Hashing, which maps similar items together by hashing them several times, avoiding the costly explicit computation of their pairwise similarity. Math-wise, we relied on the linear time Jaccard index to achieve lower and upper bounds on the more accurate quadratic time Edit Distance; these bounds were then used to approximate the Normalized Edit Distance based on the Jaccard index. The latter approximation is utilized for refinement of the clustering, which was originally done with respect to Jaccard. In particular we were able to use the efficient Jaccard index to gain the accurate, yet time consuming, Edit Distance.