Supervised Graphical Binary Threat Detection (WIP)

Nathan Waltz, Andrew Fritz, Cai Haught, Jose Sainz, Assefaw Gebremedhin
School of Electrical Engineering and Computer Science, WSU

INTRODUCTION

- The distribution of malware is prevalent everywhere, and threat actors are only getting better with time and experience.
- Examples of malware variants include Ransomware, Spyware, Adware.

  To mitigate risks and bring more security amidst chaos – we propose a supervised learning approach to detect malware by statically analyzing and classifying binary executables as threats.

STATIC ANALYSIS

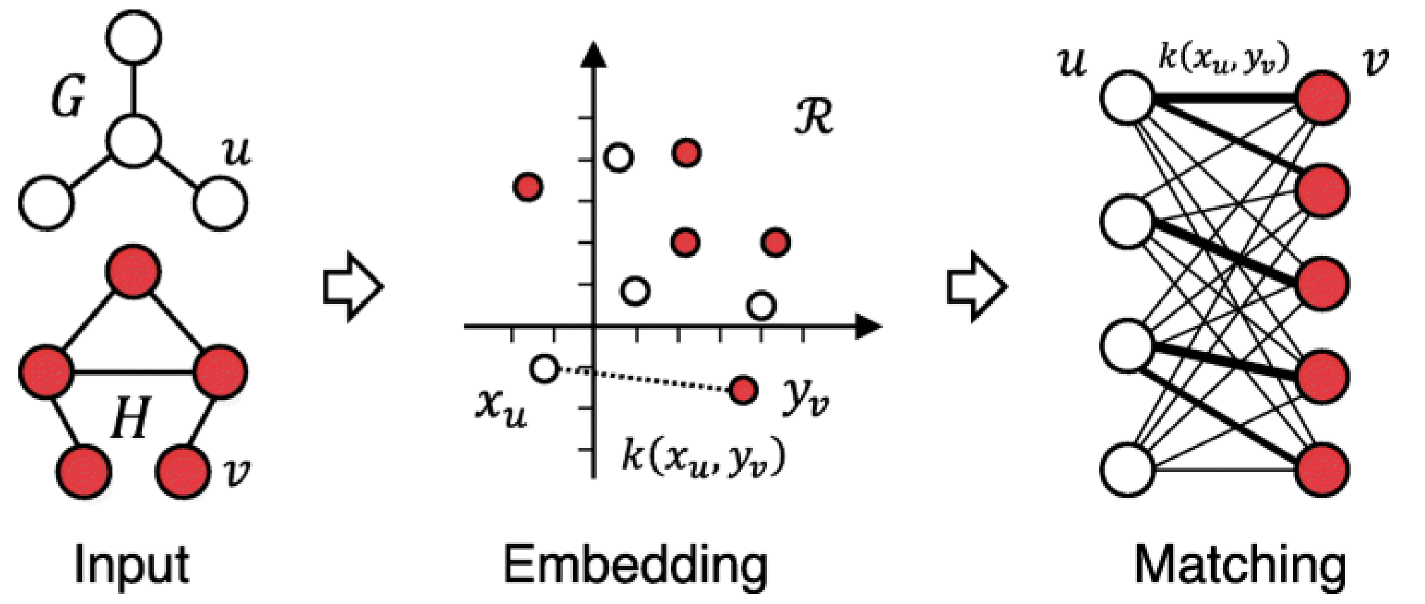
- Static Analysis (SA) essentially describes the process of extracting information from binary executables to recreate the original thing **without running the program**.
- Reverse engineering can be thought of as subsets of SA, and in this project, we will employ RE tools to extract control flow graphs and data flow graphs among other automatically extractable features into a feature vector to train a machine learning model.
- In the interest of pragmatism, focusing on scripting a tool like Radare2 to automatically extract relevant features opens the doors to a tool that is useful in a variety of use cases.
- Our dataset will be like the work done by BODMAS but will also focus on preserving structural data.

SUPERVISED LEARNING

- The machine learning process that most people are familiar with is supervised learning – whereby one labels data (assumed to be ground truth) and feeds a learner this information during the training process.
- This technique is useful to approximate an underlying concept morphism $f: x \rightarrow y$ for regression and classification tasks – where x is the training data and y is the desired output.
- To this end – we found samples of both binary and malicious Linux executables – and will label them accordingly and feed them to a machine learning model.

GRAPH KERNELS

- Graphical data is typically represented in adjacency lists, matrices (sparse or dense) – among other formats – it is important to remember that graphical data can transformed into low-dimensional space by learning an embedding.
- Graph kernels take graphical data as input, embed graphs into a lower-dimensional space, and compute the similarity between graphs.
- From this embedding, we can use kernelized learning algorithms such as support vector machine to work on graphical data – thus opening the door to using SVMs for structural data classification.

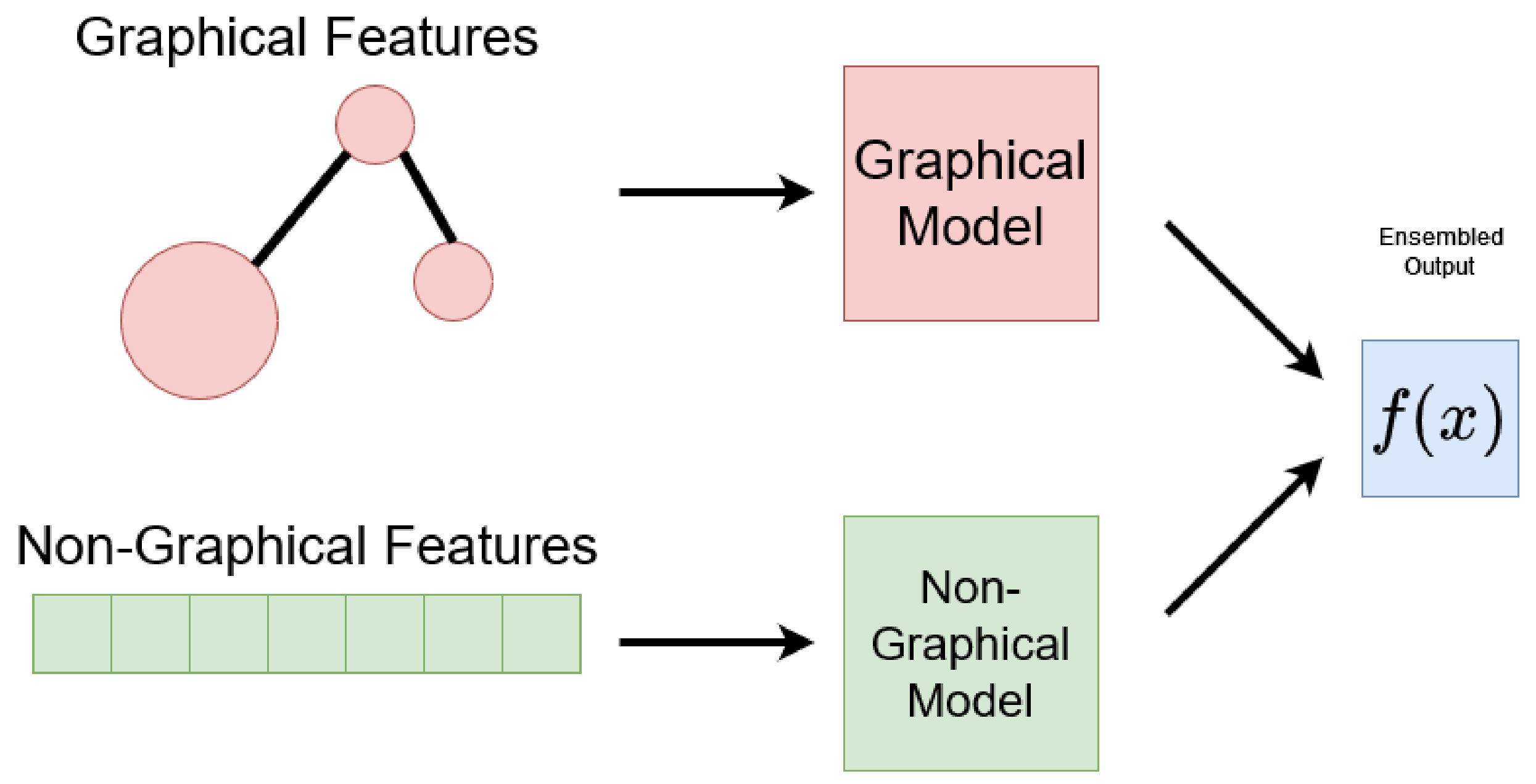


Graph Kernels Summary

Source: A survey on Graph Kernels (https://appliednetsci.springeropen.com/articles/10.1007/s41109-019-0195-3)

GRAPH NEURAL NETWORKS

- Another popular technique these days is to use something called Graph Neural Networks (GNNs).
- Neural networks are great because they can approximate nonlinear functions and typically require minimal data preprocessing.
- Comparing the performance of this method with kernelized methods will be integral in finding a good model architecture to classify threats.
- We combine one of these graphical architectures into an ensemble model – and with application logic in place to extract graphical and non-graphical features and embed those features in a feature vector – we will have a full-featured binary threat detector.



CONCLUSION

We have presented a plan to perform threat detection on executable files – although quite a bit of work that needs to be done. If you are interested in assisting in the development effort on this project, please contact Nathan Waltz @ nathan.waltz@selinc.com.

ACKNOWLEDGEMENTS

This work is supported by the Giffiss Institute under contract No. SA10012021MM0336