Introduction

There exist many tools and methods for analyzing Android malware. This research project takes a look at one of the methods of analysis known as Graph Mining. The goal of this research project was to find a link between static and dynamic malware analysis so that we may combine their advantages when analyzing malware. Android malware is becoming harder to detect and is beginning to take advantage of various evasion techniques. This research project proposes a method of linking together static and dynamic analysis through graph mining.

Method

We begin the process by placing benign samples and malware samples into our static analysis tools in order to generate our Function Call Graphs (see Figure 1). We also analyze some malware samples in various online dynamic analysis tools, such as Andrubis and SanDroid. The obtained API method calls are then fed into the static analysis graphlet generation process which returns frequency-based graphlets for the full Function Call Graph as well as graphlets for only the critical API method calls.

Results

Manually generating the graphlets consists of searching for the critical methods in the Function Call Graph's file (.gexf). Once a critical method is found, the graphs are sketched, and frequency analysis is performed to find the most frequently appearing graphlets. Figure 3 shows a critical FCG snippet with the manual frequency analysis used to obtain the graphlets from the TapSnake malware.

Dataset

The following datasets were used for analysis:
- Android Malware Genome Project
- Contagio Mobile
- Various Google Play Store samples

Manually analyzed malware samples include:
- TapSnake (Genome)
- SnDApps (Genome)

Future Work

Future work consists of analyzing a larger set of malware and benign samples to further understand the differences in benign and malware applications. A larger, more fine-tuned selection of API method calls in the graphlet generation process will give us more a better understanding of malware behaviors. We also need a way to dynamically select critical API calls from many malware samples at a time.

Tools

- DroidBox
  https://code.google.com/p/droidbox/
- Andrubis
  https://anubis.iselab.org/
- AndroGuard
  https://code.google.com/p/androguard/
- SanDroid
  http://sanddroid.xjtu.edu.cn/

References


Acknowledgements

This research was made possible with the support of the Indiana University-Purdue University Indianapolis Department of Computer Information and Information Science, as well as through funding from the National Science Foundation and the United States Department of Defense. The author would like to thank their mentor, Dr. Feng Li, as well as Dr. Eugenia Fernandez, and Sheila Walter for their support during the course of this research.