Malware Classification with Machine Learning and Image Clustering

By: Noah Neundorfer, Ryan Morganti
Advisor: Dr. Derek Reimanis

Image-Based Analysis

Because of the various forms of malware, as well as the sophisticated techniques that malware authors use to mask or spoof malware, malware recognition techniques require creative representations of the malware. One such representation is to transpose a byte-by-byte representation of a malicious file into a grayscale image, with the byte value of 0 being black and 255 white. This creates a standard image file with the content of the image being black and white pixels. However, since malicious files may be filled with erroneous data, comparing exact images becomes problematic and distinctive features may be lost. By breaking the binary into its pre-defined segments and generating grayscale images for each section, we can isolate similar sections. Following, we can utilize existing machine learning algorithms to cluster based on the grayscale segments. This process reveals how malware segments have been re-used or spoofed.

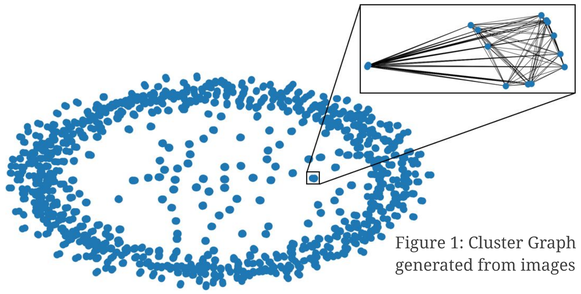

Figure 1: Cluster Graph generated from images

Figure 1 shows the resulting cluster graph when the text sections are compared using a simple image difference algorithm. Each blue dot is a collection of 2 or more connected nodes.

Comparisons & Results

The images shown in figures 2 and 3 show two grayscale image representations generated from two seemingly distinct malware binaries with separate malware classifications from VirusTotal[3] .

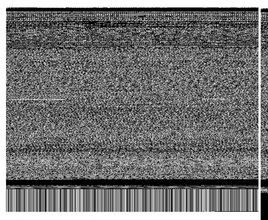

Figure 2: Trojan.murofet/licat

Figure 3 features a longer image because the size of the binary is larger. However, a visual inspection reveals that almost the entirety of figure 2 is included in the beginning of figure 3, with pixels matching almost exactly. This suggests that there is a relationship between these two binaries, perhaps figure 3 uses code from figure 2, or figure 2 is a trimmed down version of figure 3.

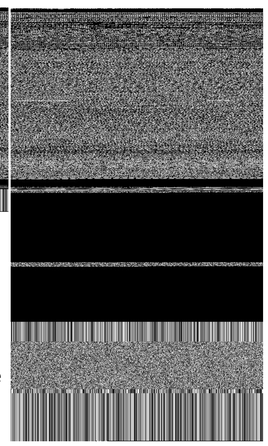

Figure 3: Trojan.swisyn/auzw

This result is interesting as it shows that it is possible to compare malware based on their segments and that potentially different malware may be reusing the same code. This demonstrates that a certain cluster can be represented by a single image, which represents a fingerprint for that class of malware. By using a single image fingerprint per cluster we can speed up malware detection techniques.

```
Cluster 10:
trojan_murofet/licat   -  6
trojan_murofet/zbot    -  1
trojan_swisyn/auzw     - 52
trojan_swisyn/bikxq9ci -  7
trojan_swisyn/buzy     -  1
trojan_swisyn/graftor  -  4
trojan_swisyn/mofksys  - 19
trojan_swisyn/vebby    -  6
virus_murofet/zbot     -  1
```

Figure 4: List of clusters

Further research could focus on creating a more accurate machine learning model, as the one used in this research can be improved upon.

Introduction

Malware, which is a broad term that represents unwanted and potentially dangerous programs, is becoming more prevalent and dangerous, costing companies and governments significant resources and damage. AVTest[6] , a malware tracking company, has found that since 2018, the total number of malware samples collected has increased by over 500 million samples.

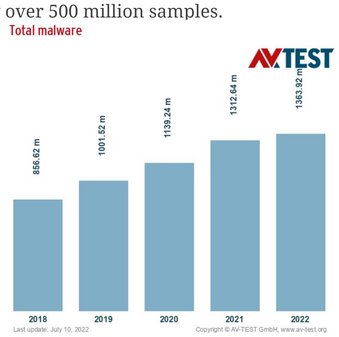

Figure 5: AVTest Graph 2018-2022, image originating from [6]

The goal of this project is to develop more accurate techniques to determine the differences between malware and good programs, or benignware, including new malware recognition techniques. To meet this goal, two approaches were used, image-based classification and meta-data classification.

Current malware differentiation techniques rely on file hashes such as MD5 or SHA256. File hashing techniques are quick and easy, but are defeated with modifications to files, such as new versions of the malware, or even the modification of a single bit within the malware. In this work, we found compelling results that suggest image-based classification and meta-data classification are valuable techniques for malware identification and differentiaion. By exploring new ways to represent and analyze malware, we provide cyberdefenders with advanced techniques to defend against malware.

Background Information

Malware disguises itself in many ways, and malicious code can appear in any website, file, or program. On a Windows computer, malware typically appears as Windows PE Files which are EXE programs. This project exclusively analyzed EXE files.

EXE files contain a defined header that contains information about the program, such as file size, segment offsets, version number, etc. Following the header are segments, each a self-contained chunk of the program with a specific purpose. Commonly, the .text segment will contain the program code, .data data about the program, and .rsrc icons and resources. The meta-data approach uses tools that extract information from the header. The image-based approach compares the generated segment images.

References

1 Bhodia, Niket & Prajapati, Pratikkumar & Di Troia, Fabio & Stamp, Mark. (2019). Transfer Learning for Image-Based Malware Classification.

2 Nataraj, Lakshmanan & Karthikeyan, Shanmugavadivel & Jacob, Grégoire & Manjunath, B.. (2011). Malware Images: Visualization and Automatic Classification.

3 *Virustotal*. VirusTotal. (n.d.). Retrieved July 27, 2022, from https://www.virustotal.com/

4 *Manalyze*. Manalyze. (n.d.). Retrieved July 28, 2022, from https://manalyzer.org/

5 Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

6 *AVTest*. AVTest. (n.d.). Retrieved July 28, 2022, from https://www.av-test.org/en/statistics/malware/

Meta-Data Analysis

Malware authors often mask or spoof their code to confuse malware detection tools. Such techniques, referred to as obfuscating or packing, make it difficult to reverse-engineer the program to detect if the code is malicious in nature. However, begignware seldom features these techniques because compilers do an adequate job at jumbling the original source code. By detecting whether a program has been obfuscated or packed, we gain insight into the author's intentions. Fortunately, these descriptive features and more are saved within the meta-data of the program. We can use existing tools, such as Manalyze[4], to extract the meta-data of both malware and benignware with the ultimate goal to discover the differences between the two. Though, because of the surprising difficulty in collecting a dataset of contextually similar benignware, our analysis focuses on understanding the meta-data differences within malware. Specifically, we sought to classify malware type based on meta-data.

To this end, we extracted notable meta-data, or aspects, of known malware files using the tool Manalyze. In total we extracted 20 different aspects, and examples of these aspects are shown in figure 6.

Aspect:	Data Type:	Aspect Description:
OverlayStatus	Nominal	Whether an executable contains overlaid data
FileSize	Numeric	The size of a file in MB
HidingImports	Nominal	Whether an executable appears to be hiding it's imports

Figure 6: Example Aspects

The full set of data, which included 7000 instances and 20 individual aspects was analyzed using a supervised machine learning method of a decision tree.

Analysis & Results

We used WEKA's[5] implementation of the decision tree algorithm, specifically the ForestPA model. ForestPA constructs a decision forest, which is a collection of individual decision trees based on the significance of the aspects, with a penalty applied to aspects that are used frequently. Our model was trained using 10-fold cross validation, and produced results based on ability to classify the malware type.

```
Correctly Classified Instances         5074               72.7768 %
Incorrectly Classified Instances       1898               27.2232 %
Kappa statistic                          0.6941
Mean absolute error                      0.0014
Root mean squared error                  0.0271
Relative absolute error                 39.4491 %
Root relative squared error             64.2689 %
Total Number of Instances              6972
```

Figure 7: ForestPA Results

As shown in figure 7, our model was able to successfully classify the malware type for 72.7% of the data. That is, when presented with values for 20 aspects, we were able to correctly predict the type of malware program 72.7% of the time. These results are promising, and future efforts will seek to expand the number of malware instances, include more aspects, and generate a collection of benignware. Ultimately we seek to generate a similar model to distinguish malware and benignware.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1947750. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation