Pruthvi Punwar
Midterm 2 Report
CS 185C

-- Pre-processing

I decided to do the pre processing in python, as there were some inbuilt libraries that would have made the coding easy. The malware dataset that we were given contained a lot of different families but I focused on three malware families named Winwebsec, Zbot, and ZeroAccess. So, malware families are basically a collection of malware that is produced from the same code base. In the families, each file was a collection of already extracted, already cleaned opcodes. I used an OS module and numpy library in the code. These modules provided us a portable way of using operating system dependent functionality. Mobile itertools helped to implement a number of iterator building blocks. First off all, after reading the files in the malware families, we counted the occurrence of all the different opcodes and sorted them from most occurrence to least occurrence upto 29 different opcodes and the 30th was named discarded which contained all the other opcodes.PI was also during the iterations and assigned to initial_values variable. Then, I used the json functionality in python to clean this data and print them in two different files observation_matrix and transistion_matrix. I also used an array of symbols to assign the opcodes to different symbols. The next step was to use these data to train our Hidden Markov Model.

-- Training

We used the alpha_pass and beta_pass function of the HMM model to train the data. After printing the formatted data to the files, we used the data for transistion_matrix, observation_matrix and PI and assigned those values in the main function. Observation matrix was also declared in the main function. After getting all the data, I used it to train our HMM Model.

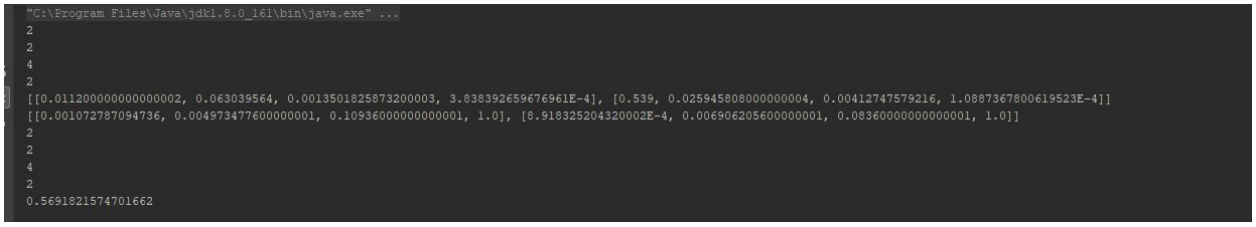
This is the result for zbot, while having a AUC score was 0.089.

```
2
2
4
2
[[0.1184, 0.00490086, 0.0017517141930799998, 5.260146587366327E-4], [0.0825, 0.0073520569999999995, 7.094305484999999E-5, 7.40084783971382E-5]]
[[0.004188770963439999, 0.08952669599999998, 0.33879999999999993, 1.0], [0.0012614867280299996, 0.021934722999999993, 0.09221999999999998, 1.0]]
2
2
4
2
0.08999700853324714
```

For zeroaccess, AUC score was 0.0071

```
2
2
4
2
[[3.2E-4, 2.1845399999999996E-5, 2.1272437599999994E-8, 1.3883903853119994E-10], [0.00715, 3.37552E-5, 2.7751526399999994E-8, 5.6275883833599987E-11]]
[[8.122657423999999E-8, 6.105175999999999E-6, 0.006579999999999999, 1.0], [2.3653485119999994E-8, 1.8291969999999995E-6, 0.001987, 1.0]]
2
2
4
2
0.007183783007802284
```

For winwebsec, AUC score was 0.569. This result leads me to believe that winwebsec is very different from the other two families. I can also think that Winwebsec may be a little more similar to the other two than they are to each other



-- Conclusions

As malware has become really complex to detect in recent times with old techniques, new techniques are needed, and machine learning based tools seem a promising avenue for it. Hidden Markov models are one such technique that has been used effectively for malware detection as seen in the project. For future work, we can use the Hidden Markov model to detect encrypted malware. We can do that while working on top of the current project.