Weekly Report 2.13

This week:

- Optimization for making the program to work faster
- SVM training and classifying
- A bug with the YMSG rule based classifier was fixed
- Database module was initiated
- General user interface improvements

Next week:

- Optimization
- Initiating decoder integration
- Working on SVMclassification for finding out the accuracy
- Database implementation and testing

Elvan Gülen: This week, I worked on optimizing the code. There is a fasterMode option which I'll integrate it to the GUI next week. By this option, the connections that haven't got a high match value don't go into the rule based classifiers and SVM. For time being, we look the match value after 10 lines for text based and 256 kb for binary protocols, and the match value limit is 10%. There is a little bit speedup but to have a better opinion about the speedup we should try it with more pcap files. Also, I gave a hand to Can about training the SVM (The training was complete but after talking to Tolga Can we decided to train SVM with -b). Besides, the SVM classification has been initiated. We just try it with POP3 and YSMG. It doesn't give POP3 a good result because of training the POP3.SVM with a little matched POP3 connections. But we have a large data set for YMSG and since there are lots of YMSG connections for training it gives good result for both online and offline capturing. While Can and I were working on the SVM module in an online capturing mode, we noticed that there was a problem with YMSG. For a YMSG connection, the match value came from the YMSG rule based classifier was too low namely %36. To solve this problem, I spent too many hours on debugging. The problem arose from a truncated file transfer. The buffering and the operating on the buffered data had some leaks. However, the problem is solved with a little help of Can. Next week, I'm planning to work on integrating the decoder module. Besides that, I'm going to think and implement further optimizations.

Nazif İlker ERÇİN : This week, I could not work on the project too much. I only read the documentation of SQLite from its website tried to learn how to implement the database. Additionally I tested the current version of the program. Next week I am planning to start implementation of the database structure of ACCIPP using SQLite. Additionally, if needed, I will help optimization of the code in terms of operating speed.

Çağla ÇIĞ : This week I did not work on the project so much. I studied the tutorials for SQLite. Additionally I tested the program. Next week I am planning to implement database module of the project with İlker. If time allows, I will continue with testing the program, including the database module which will be just implemented.

Can Hoşgör: This week, I worked on several parts of the project. First of all, I worked on the SVM module issue mentioned in the previous week. After trying several things, I and Elvan noticed that the issue was caused by the lack of positive examples we placed in the training file. When we tried the SVM module with the YMSG training file, we have seen that svm_predict is functioning without a problem. In addition to that, as suggested by Tolga Can & the author of libsvm, we are now giving an extra parameter to svm-train that allows us to get probability estimates for our predictions. We are using this estimate as the match value. Next, I helped Elvan with the optimizations suggested by Sevgi Yaşar. We have added a "fast mode" switch that stops sending data to classifiers with very low match values, after a certain amount of data is sent or received. Next, I worked on the GUI in order to make it more responsive. Previously, although the user interface and decoder executed in separate threads the GUI would become irresponsive under heavy load. I have implemented a different algorithm (similar to non-persistent CSMA) for updating the user interface which gave more appealing results than the previous one. Next, I initiated the database module. Since the database module is mostly I/O bound, putting it in a separate thread improved a lot. Now we have no performance problem while updating the database. Finally, there was a performance issue we have faced while testing in Windows Vista. The IP Selection dialog which needs to parse the entire pcap file to list the available IP addresses took to much time to pop up. I have examined the code to find out what is causing the performance bottleneck, and tried several approaches to the problem. Despite all my efforts, I have realized that the reason was pcap_next_ex function. Since this function is part of the WinPCap library and we have no control on the WinPCap routines, it is not currently possible for us to solve the problem. Next week, I'm going to continue my work on the SVM module, and train it with more pcap files to improve prediction accuracy. Also I'm going to implement some more final optimizations and do some testing.