WinstonSoft Weekly Report 2.8

This week :

- Implemented the SVM module and trying to integrate it to our code.
- Conducted research on libsvm library to use it in our program.
- Developed the identifier module of YMSG protocol.
- Applied to Yeni Fikirler Yeni İşler with our senior project.

Next week :

- Improvements on SVM module.
- Research on libsvm library.
- Continue implementation of YMSG matcher.
- Improvements on FTP matcher.
- Implementation of the identifier module of NNTP protocol.

Can HOŞGÖR: This week, I made major improvements to the SVM module. The SVM module now calculates the frequencies of all bigrams that occur in payload fields of packets. Each connections is treated separately, so each connection has its own copy of the bigram vector. After a connection is closed, the SVM module takes match results from all other rule based matchers in order to see if any of them could identify the connection. If a reasonable match is found, the SVM matcher appends this to the dataset.txt file in the format that libsvm understands. Currently I'm looking for sample pcap files in order to build a larger dataset. Next week I'm planning to make some further improvements to the SVM module, and if time allows I'm going to fix some of the bugs that are present in the user interface.

Elvan GÜLEN: This week, I analyzed the details of some unofficial documents (about YMSG) that I found in the previous week. The basic structure of the payload is quite fixed. The first 4 byte is YMSG, and the next 4 byte is version and goes on. However, the most important part of the payload is the data which includes the information of the user's acts. In this data there are keys and values separated by a argument separator, there is no significant information online about what are those keys and values. So, I captured some YMSG packets by WireShark and tried to figure out what is going on with this keys and values. By this observation, now I'm nearly able to catch chat messages, the friend list, the status of the friends and the user, the file name and size of the file that is transfered and etc. But I haven't implemented this part yet. On the other hand, I implemented the general structure of the YMSGMatcher class and also I use some specific parts of the payload to say that whether it's YMSG protocol or not. The result is very satisfying; it finds YMSG nearly %100 and for packets that don't include YMSG protocol it gives nearly %0. Besides, I find the keys and values in the payload and show them in the summary pane temporarily. The next week, I'll continue to work on the YMSGMatcher class. I'll analyze some other YMSG pcaps for further knowledge about the protocol and make the program to give a more systematic summary. Also if time allows, I'll deal with the SVM part of the project.

Nazif Ilker ERCIN: This week, I have worked on the libsvm library for the SVM module of the project. I have conducted some research about the library and tested the executables downloaded with the libsvm package from the website of libsvm with the data that our bigram generator supplied. Additionally, I helped Elvan a little about the YMSG matcher module. As the deadline for Yeni Fikirler Yeni İşler was this Friday, we have tried to fill the application form in detail, with Cagla. Next week, I am planning to do some further research about libsvm and try to integrate it to the project. I'm planning to work on a generic n-gram generator. If the time allows, I will be dealing with the "uploaded file capture" of FTP Matcher.

Çağla Çığ: Since the deadline for the applications of Yeni Fikirler Yeni İşler was 18th of April, İlker and I spent the first days of the week preparing the necessary documents and filling in forms related to the contest. Then, we had a brief meeting with Onur Tolga Şehitoğlu for assistance about the preparations for the contest. After we completed the application, I started working on YMSG documents which is the binary protocol we chose to implement. Then, we chose to input n-grams of pcap files to svm using libsvm however the parameters were hard to decide and input wasn't large enough. So, using wiki.wireshark.com I started collecting appropriate input to feed svm. The implementation of 3 text-based and 1 binary protocol is finished, so the following week I am going to read the RFC document of NNTP protocol and implement it. Since the class hierarchy is formed and I have already implemented a text-based protocol, I am planning to finish it in a week.