This week, we have done literature survey on three different documents that can be found below. Additionally, we have developed a very simple prototype that recognizes SMTP. We have made a research on “clustering classification” and studied on MSN Messenger Protocol. The website is designed from the beginning. Below you can find the details.

Prototype Development

This week, we began developing our initial prototype. As we have decided to use a pcap-based tool to capture network packets, the first thing we did was to install WinPcap. WinPcap consists of a driver, that extends the operating system to provide low-level network access, and a library that is used to easily access the low-level network layers. This library also contains the Windows version of the well known libpcap Unix API. After installing the WinPcap development package to our development environment, we began studying one of the code samples from the package. This gave us a quick grasp on how to initialize and use the WinPcap library. Next, we started to modify some portions of the code, particularly the callback function that gets called upon sending/receiving a network packet. This callback handler is the core of our project since everything else (for instance, protocol handlers) gets called from it. For the time being, we just added a simple string matcher for protocol-specific strings and some code to dump the packet contents upon a successful match (i.e. when packet payload contains strings particular to SMTP protocol). In the following weeks, we are planning to improve this prototype to handle more than one protocol and implement a more powerful pattern matcher.

The detection fragment of the packet handler:

```
if (size_payload > 0) {
          printf("   Payload (%d bytes):\n", size_payload);
          if    (strnstr(payload,    "MAIL    FROM:",    size_payload)    ||
strnstr(payload, "RCPT TO:", size_payload)) {
                printf("SMTP detected!\n");
                while (size_payload--)
                      printf("%c", *payload++);
          }

}
```

Research on MSN Messenger Protocol

This week, we started to do research on MSN protocol. Since MSN protocol is a proprietary protocol, there is no RFC document for it. So we glanced at various resources about MSN Messenger Protocol. We have a general idea about MSN Messenger Protocol, **M**icro**s**oft

Notification **P**rotocol (MSNP), MSN Client Protocol and in which points we are related to these protocols for recognizing MSN protocol.

The messenger protocol has undergone some several revisions from 1999 until now and some versions 8, 9, 13, etc were released according to these revisions. These protocol versions are written as "MSNP8", "MSNP12". We aren't going to analyze all version protocols seperatly but consider all main commands used in all versions and combine some additional new commands to it. Speaking of commands, we learn some commands sent between the client and the server. For example: When someone signs out, server sends " FLN nilkercin@hotmail.com ". Also there some status commans like NLN for *avaliable*, BSY for *busy*, AWY for *away*, etc. And for changing these status there is CHG command. Changing the display name is done by using the REA command. Also there are some error commands such as "200 Syntax Error", "217 User not online", "Not logged in", etc. After, knowing some of these commands, we inspected a msnms.pcap and observed these commands in real-time. Even if our research about MSN Messenger Protocol haven't finished yet, we gained a general idea about this protocol.

References:

[1] http://msnpiki.msnfanatic.com/index.php/Main_Page
[2] http://www.hypothetic.org/docs/msn/notification/presence.php

Research on Clustering Classification

In the context of our research, classification stands for the derivation of a function that will separate data into categories, or classes, characterized by a distinct set of features. This function is mechanized by a so-called *network classifier*, which is trained using data from the different classes as inputs, and vectors indicating the true class as outputs.

A network classifier typically maps a given input vector to one of a number of classes represented by an equal number of outputs, by producing 1 at the output class and 0 elsewhere. However, the outputs are not always binary (0 or 1); sometimes they may range over {0,1}, indicating the degrees of participation of a given input over the output classes. In our project, we may design the outputs range over {0,100} so that it gives the resemblance percentage.

In a classification method, such as Support Vector Machines, a **training set** (a portion of the data or a different dataset) is used. The algorithm will learn to classify the labelled data into preset categories. Feeding the microarray data to such an algorithm will now classify the data such that the algorithm will decide whether or not each data point belongs to a certain class. Therefore a *classification model* is created, which can be fed any appropriate dataset

and classify data points from there. A good example of this is the classification of cancer-like gene expression patterns and normal-tissue-like gene expression patterns. When applied to our project, each class may be designed to correspond to a different network protocol that is in the scope of our project.

References:

[1] http://documents.wolfram.com/applications/neuralnetworks/NeuralNetworkTheory/2.1.3.html.
[2] http://www.ucl.ac.uk/oncology/MicroCore/HTML_resource/Clus_and_Class_popup.htm.
[3]"Clustering and Classification Methods for Gene Expression Data Analysis", Garrett-Mayer E., Parmigiani G., 2004.

Literature Review
on
***No Port Network Protocols* Detection by Sevgi Yaşar**

In this document, basic steps of the process are mentioned that are filtering, feature analysis and classification. Nearly all these steps are going to be included in our project too. For example; as we designed earlier, our filter will remove unwanted parts of an IP packet like the **filtering** step explained in the document. In addition to this, we will allow the user to choose some filtering ways. Parameter extraction and feature extraction mentioned in **feature analysis** step, seems useful for retrieving probably most useful information from the input data set but just for now we don't know exactly how we are going to choose which elements of data set would be more useful for recognizing network protocols. In the last step, **classification**, four different concepts for assigning the object to a category, generally name it clssification, are mentioned in the document which are:

-Member-Roster Concept
-Common Property Concept
-Clustering Concept
-Neural Aproach

We are going to conduct research about these concepts in the following weeks and find out which concept is better and competible for our project. In fact, in this week we started to search about clustering classification.

Literature Review
on
***Feature Extraction for Integrated Pattern Recognition Systems* by X. Wang and K. K. Paliwal**

Conventional Pattern Recognition Systems:

1. feature analysis (parameter extraction and **feature extraction**) ,
2. **pattern classification**.

Feature extraction and pattern classification can be done independently or jointly. **Support Vector Machine(SVM)** is an integrated pattern classification algorithm.

Aim: to classify input data into given classes

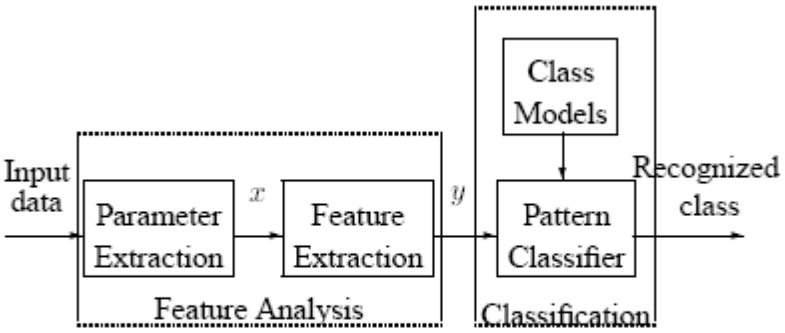

Figure 1: Conventional pattern recognition system.

Drawback of independent feature extraction algorithms: "Their optimization criteria are different from the classifier's minimum classification error criterion which may cause inconsistency between fature extraction and the classification stages of a pattern recognizer and consequently, degrade the performance of classifiers. A direct way to overcome this problem is to conduct feature extraction and classification jointly with a consistent criterion."

SVM: A recently developed kernel-based integrated pattern classification algorithm. Basically, it has the advantage of being able to handle the classes which have complex non-linear decision boundaries. The classification is conducted in parameteric space. "However, the parameteric space normally includes large amount of information irrelevant for classification and has high dimensionality. Thus, SVM classifiers are complex and inefficient." states the paper.

Consequently, although giving us a thorough idea about integrated pattern classification algorithms such as SVN, the paper in general is composed of advanced calculus formulas reflecting the complex character of the algorithm which actually don't help us much in the future development of our project.

Literature Review
on

***Network-Based Application Recognition and Distributed Network-Based Application Recognition* by CISCO**

IP Quality of Service: Provides appropriate network resources (bandwidth, delay, jitter, and packet loss) to applications so that mission critical applications get the required performance and noncritical applications do not obstruct the performance of the former. It can be utilized by defining classes or categories of applications.

NBAR: A classification engine that recognizes a wide variety of applications, including web-based and other difficult-to-classify protocols that utilize dynamic TCP/UDP port assignments. Since this is the feature used for classifying trafic by protocol and the forwarding mechanism of the classified traffic is out of our interest, the rest of the research is limited to the NBAR part of the documentation. In addition to identifying statistically and/or dynamically assigned TCP and UDP port numbers and non-UDP and non-TCP IP protocols, NBAR can also do classification based on deep packet inspection. On the other hand, NBAR is mainly focused on doing network-based application recognition by first inspecting the ports and then if the port information is not enough to recognize the protocol, finally looking deeper into the packets. This is the where our project differentiates from the one of Cisco since the general approach our project is based on is port-independent protocol analysis where we have no information about ports and all we do is to look into the packets to recognize and classify the protocols.

Restrictions: Although able to recognize nearly all common protocols,even including the non-UDP and non-TCP ones, NBAR is not able to recognize fragmented packets which is another differentiation from our project.

Website Design

Using a template, we have designed our website and load it on the server changing the previous one as we do not like it. We added the group information, information about the project and the reports we have prepared up to this point.