HIDDEN MARKOV MODELS TO ANALYZE USER BEHAVIOUR IN NETWORK TRAFFIC

Umut Tosun
Bilkent University 06800
Bilkent, Ankara, Turkey
{umutt}@cs.bilkent.edu.tr

Abstract

Understanding the nature of information flowing into a system or network is fundemantal to determining anomalies and characterizing workload.Traditional methods rely heavily on port numbers carried in the packet headers to classify protocols.These methods can fail in the presence of proxy servers,that re-map port numbers, or compromised host services.

Another fact is that in encrypted traffic, we do not have more than packet-level information provided.We present a novel approach to protocol detection with less information than one might consider.We only carry out the detection process with timing and connected server informations.

Keywords: Hidden Markov Models, k-means clustering, two dimensional clustering...

Introduction

The nature of information flowing into a system or network is fundemantal to determining anomalies and characterizing workload.With this information a system administrator understands the basic system facts; like which servers are especially used, what makes the traffic intense in the system, which users are trying to compromise with the servers and which users cause bursty traffic intensionally or not.In this paper, we specifically address the question of whether we can characterize the protocols a user exploits by looking at the sequences of durations that packets took in the transmission environment.

The traditional method of determining the client-server protocol is by inspecting the source and destination port numbers in the TCP header.However, this may fail due to Proxies, Server Backdoors and User-Installed Servers.

There are specific properties of protocols that helps us characterize user behaviour.For example http runs very fast in a short time, WWW is generally remote, SMTP has a tendancy to upload process whereas FTP has tendancy to the download process.Telnet packets are usually small in size such as two or three bytes.Domain is generally used by internal users of the environment to perform DNS queries. An important implementation of "*Behavioral Authentication*" may be found in Early, Brodley and Rosenberg.[2]In this paper, they are using the decision tree approach to detect user behaviour.User behaviour is modeled with many features such as TCP flag, mean interarrival time, mean packet length for window of n packets etc... Inspite of the fact that this approach is strong, in an encrypted environment, we may not be able to desiccate all the features of these packets.Morover, in an environment where there are many users and many user behaviours, a rogue intruder is possible to conceal his attack patterns.

In response to the difficulties outlined above, we have chosen an approach using Hidden Markov Models(in the remainder of the paper, HMM's will be used as the plural of HMM) to classify user behaviour by perusing the sequences of durations that packets took in the transmission environment.

The protocols that we have chosen to analyze are Domain, Telnet, Login, NNTP and SMTP.We have chosen these protocols because they provide complete characteristics of both local and remote users of the system.An important point is that, these are all application layer protocols having TCP in their transmission protocol.Our dataset which we obtained from "*Internet Traffic Archive*"[12] contains only application layer protocols.

The remainder of the paper is organized as follows.We present in detail the salient properties of an HMM that motivated our approach.We present how we classified the user behaviour.We will compare our approach in classification with clustering techniques like k-means clustering.Then we discuss our HMMs and system architecture.We present how the emprical results are.We provide how one dimensional or more than one dimensional data affect the results.Eventually we summarize the results comparing them with our expectations and we discuss our plans for future work, including investigation of memorization, unbalanced training of HMMs, distorted data and more-than-two dimensional analysis of user behaviour patterns.

Hidden Markov Models

The Hidden Markov Model(Figure 1) starts with a finite set of states.Transitions among the states are governed by a set of probabilities(transition probabilities) associated with each state.In a particular state, an outcome or observation can be generated according to a seperate probability distribution associated with the state.It is only the outcome, not the state, that is visible to an external observer.The states are "hidden" to outside; hence the name Hidden Markov Model.The Markov Model used for the hidden layer is a first-order Markov Model, which means that the probability of being in a particular state depends only on the previous state.While in a particular state, the Markov Model is said to "*emit* an observable corresponding to that state.One of the goals of using an HMM is to deduce from the set of emitted observables the most likely path in state space that was followed by the system.

Given the set of observable states contained in an example corresponding to a user behaviour sequence, an HMM can also determine the likelihood of a protocol type of a specific type.In our case observables are the sequences of durations that packets took in the transmission environment.Each example is constructed to contain all of the packet sequences coming from distinct ip's in each hundred packets captured from the network traffic.HMM parameters for classification are fixed by the intervals of the models in one dimension using histograms with respect to the sequences of durations that packets took in the transmission environment.The HMM parameters are the initial probability distribution for the HMM states, the state transition probability matrix, and the observable probability distribution.

The state transition probability matrix is a square matrix with size equal to the number of states.Each of the elements represents the probability of transitioning from a given state to another possible state.For example, the likelihood of transitioning from the state corresponding to a packet size sequence 0,2 ms -0,25 ms -0,25 ms is more probable than 0,2 ms -0,2 ms -5000 ms because the domain packets are mostly accumulated between 0ms and 0,5ms.The observable probability distribution is a non-square matrix, with dimensions number of states by number of observables.The observable probability distribution represents the probability that a given observable will be emitted by a given state.For example, n the late stages of a domain packet sequence duration is more likely to be between 0 ms and 0,5 ms than 175 ms-200ms.

We implemented our system using a seperate HMM for each classification category since the standard HMM is designed to simulate a single category.We are currently working to paralelize these several categories with multiple threads in operating system.

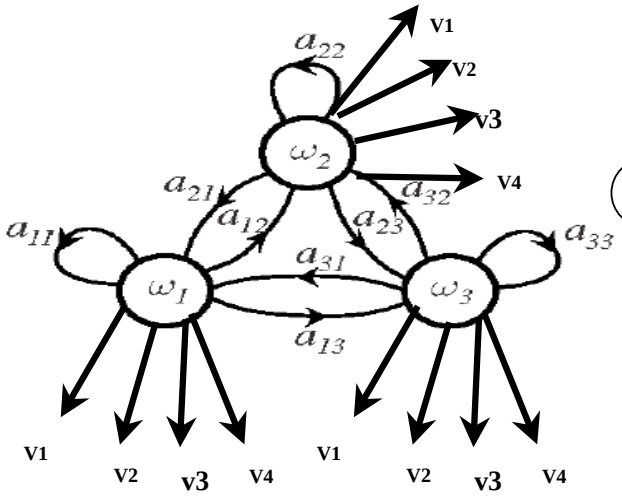
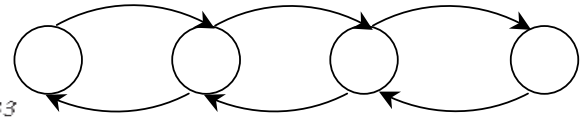


Figure 1



Four sample stages of Domain Protocol HMM.

Stage 4=>0,2ms-0,25ms
Stage 5=>0,25ms-0,3ms
Stage 6=>0,3ms-0,35ms
Stage 7=>0,35ms-0,4ms

Figure-2

Hidden Markov Model Computation

We present some of the computational information about HMM here.One may refer to Rabiner[6] for an extended overview of HMMs.

In general, networks such as Figure 1 are called finite state machines, and when they have associated transition probabilities, they are called Markov Networks.They are strictly casual: The probabilities depend only upon previous states.A Markov Model is called ergodic if every one of the states has a non-zero probability of occuring given some starting state.A final or absorbing state w_0 is one which, if entered, is never left.(i.e. $a_{00} = 1$).In the above figure we denote the transition probabilities a_{ij} among hidden states and b_{ij} for the probability of the emission of a visible state:

$a_{ij} = P(w_j(t+1) \mid w_i(t))$

$b_{jk} = P(v_k(t) \mid w_j(t))$

We demand that some transition occur from step $t \rightarrow t+1$ (even if it is to the same state) and that some visible symbol be emitted after every step.Thus we have the normalization conditions:

$$\sum_j a_{ij} = 1 \text{ for all } i$$

$$\sum_k b_{jk} = 1 \text{ for all } j$$

Where the limits on the summations are over all hidden states and all visible symbols, respectively.With these preliminaries, we can now focus on three central issues in HMMs.

The Evaluation Problem

Suppose we have an HMM, complete with transition probabilities aij and bjk.Determine the probability that a particular sequence of visible states V to the power T was generated by that model.

The Decoding Problem

Suppose we have an HMM, as well as a set of observations V to the power T.Determine the most likely sequence of hidden states w to the power T that led to those observations.

The Learning Problem

Suppose we are given the coarse structure of a model(the number of states and the number of visible states) but not the probabilities $a_{i,j}$ and $b_{j,k}$.Given a set of training observations of visible symbols, determine these parameters.

All of the three problems are solved in detail.[6,10]Our aim is not to tell the theory for the sake of brevity.In our experiment, we used the hidden markov library SparseDiscreteHMM.[11]We discretized the continuous points of sequences of durations that packets took in the transmission environment, to centroids by drawing histograms of the prevailing data for each of the protocols SMTP, NNTP, TELNET, LOGIN, DOMAIN.We trained each model with first training data and then we tested them by test data.We have chosen the number of hidden states to be five after several experiments.

Why Use an HMM?

There are several properties of packet sequences corresponding to protocol usage of a client that match with HMM.A packet sequence consists of meaningful sequence of packets such that it is not so expected that an unexpected size packet arrive after another.For example we do not usually expect a telnet packet as large as FTP to enter the system because usually telnet packets are two or three bytes in size.From an intrusion detection point of view suppose we are trying to learn whether or not a host exist in the given ip address.One method to achieve this goal to attemp to initiate a telnet connection with the hostA second is to consult the DNS server responsible for the network.A third method might involve to establish a login connection.Out of the various possible actions, only one is chosen, resultig in the observable for the step.The likelihood associated with each alert type is calculated according to the protocol usage patterns.The hidden and visible layers of the HMM have semantic value and could support the development of a cognitive model of the way attackers accomplish their goals.For example attackers sometimes attempt to disguise their work as normal network activities.The attacker only reveals his motives when he has gained his objectives-and sometimes no then.The HMM observable layer models the overt portion of the attack, whereas the hidden level represents the attacker's true intensions, a useful property when the attacker attempts to hide his true intensions.In this paper, we are not formerly in the aim of detecting attack patterns.Our main purpose is to model protocols first from as minimum as possible network parameters.

Classification of System Flows

In this section we describe how we classified our data into labeled data and how we obtained the training and test datasets.In our experiments, we observed the Wide Area Network Traffic of the Lawrence Berkeley Laboratory[12] which consists of packets captured from their network traffic.Actually the traffic reveals their server connections throughout the remote and local users.The dataset consists of these features: timestamp, duration that packets expend in the transmission environment, protocol, bytes received, bytes sent, server ip(re-numbered for security purposes) ,client ip(not re-numbered for geographical evaluation purposes) ,SYN/FIN flags to acknowledge whether or not a packet arrived succesfully and another flag to understand whether a client is local or remote.

We implemented our experiments especially in two types of features.We first tried to understand whether we can model all the packets in a network traffic coming from different ip's to various servers and secondly we looked at each, server connection behaviour patterns to detect packet sequences.In our experiments we looked at each hundred packets coming to network from distinct IP's.

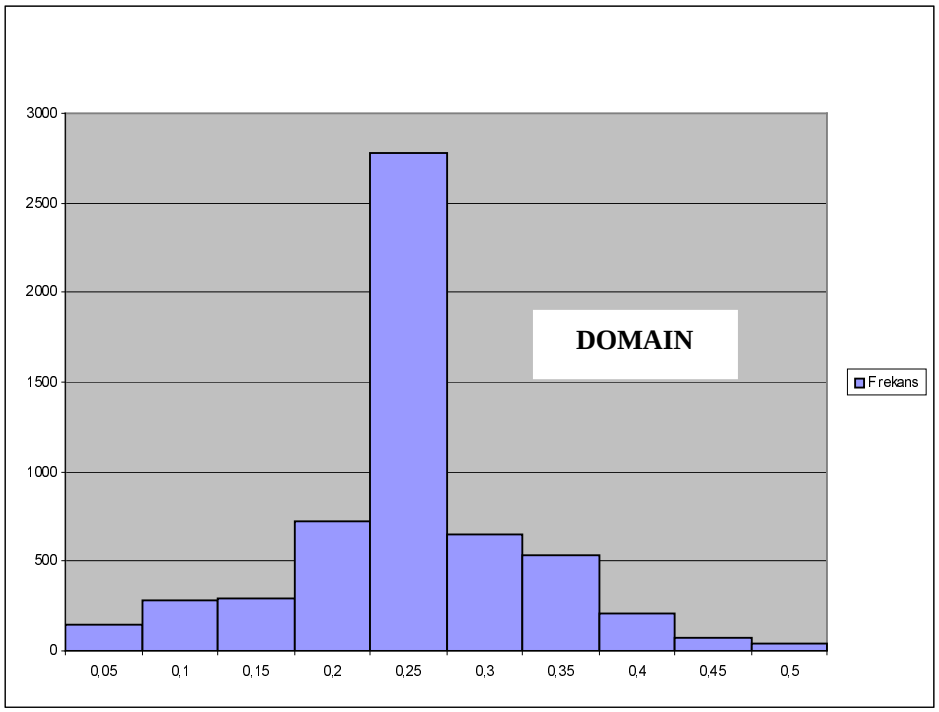


Figure-3

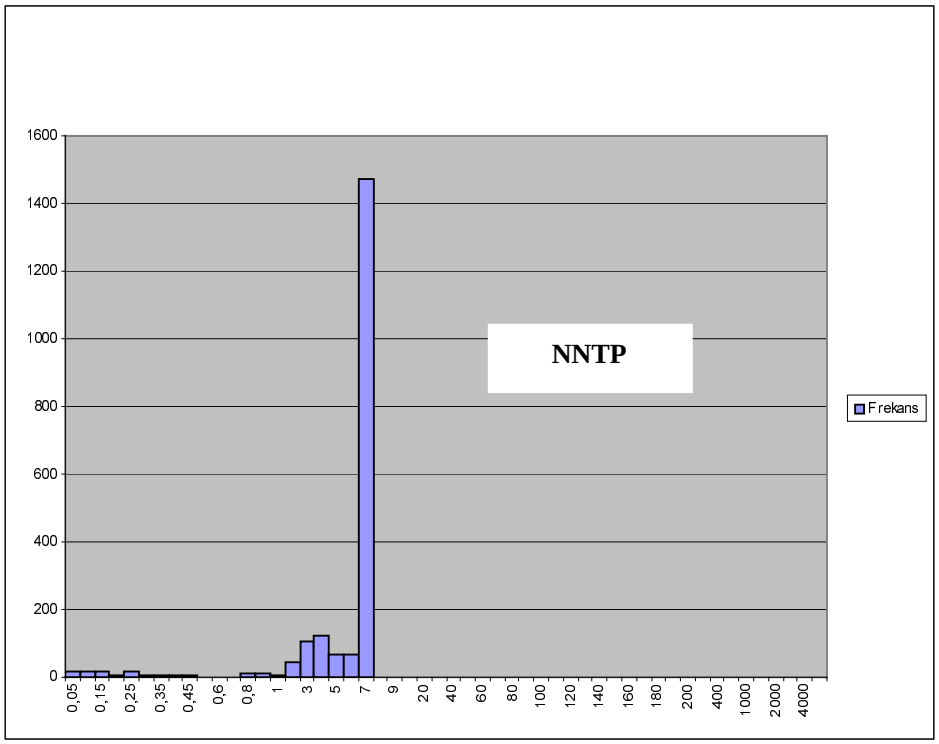


Figure-4

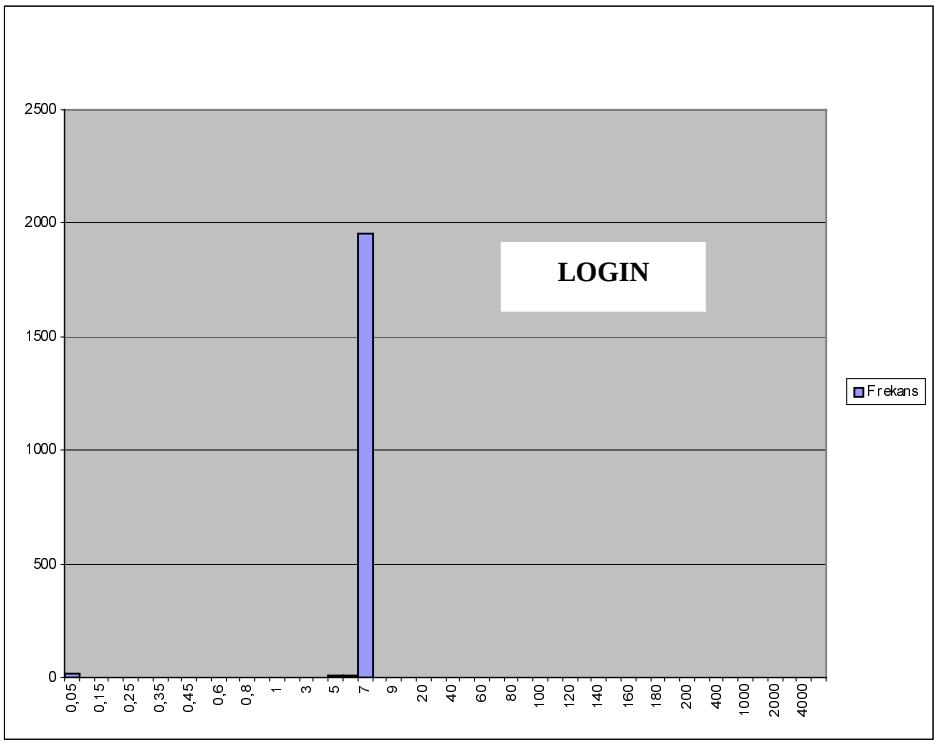


Figure-5

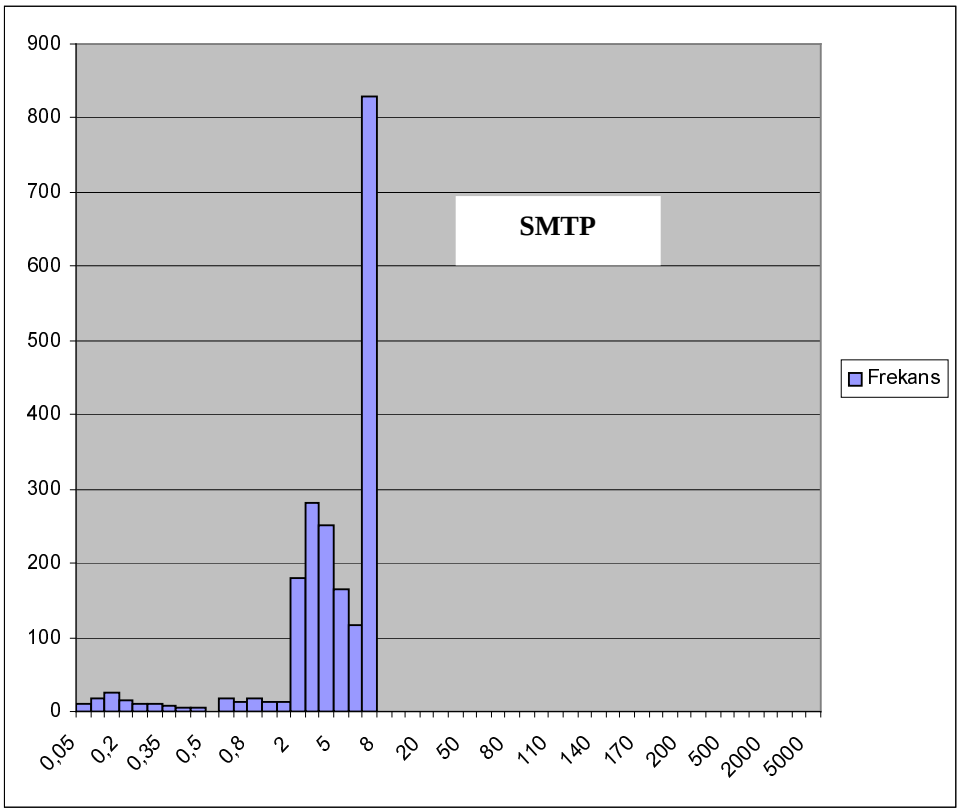


Figure-6

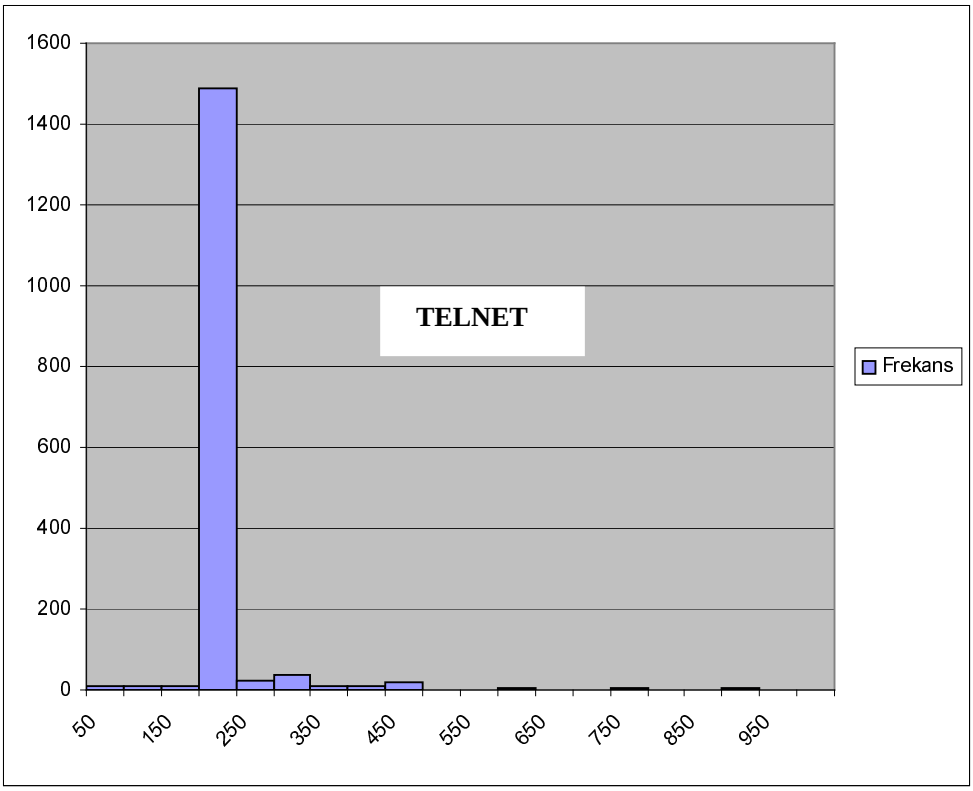


Figure-7

Then we trained our five hidden markov models with these packet sequences for 2000 packets of sequences coming from distinct ip's.(Figure 2) We only used discretized values for duration and we used Discrete HMMs to model the system.There are five HMMs in our system to detect user patterns.These are: SMTP, NNTP, LOGIN, TELNET and DOMAIN.Each state of ten observable sequences is a cluster defined with respect to durations.Normally, in the experiments performed in specific research labs people use huge amount of flows into the system to gain better results.However, we tried to show even in a rare data, HMM could be useful to protocol detection.

Then we trained our five hidden markov models with these packet sequences for 2000 packets of sequences coming from distinct ip's.(Figure 2 gives a snapshot of DOMAIN HMM) We only used discretized values for duration and we used Discrete HMMs to model the system.There are five HMMs in our system to detect user patterns.These are: SMTP, NNTP, LOGIN, TELNET and DOMAIN.Each state of ten observable sequences is a cluster defined with respect to durations.Normally, in the experiments performed in specific research labs people use huge amount of flows into the system to gain better results.However, we tried to show even in a rare data, HMM could be useful to protocol detection.

System Architecture

Our system first consists of five Markov Models and 10 observable symbols for each Markov Model.We investigated the behaviour of each model for observation sequences of hundred packets.We clustered the HMMs into pieces by using histogram intervals.One might think that k-means clustering is a better approach in many cases.However in our case the data is distorted.We observed even so inconsistent packet durations in our clustering trials in Weka[13].For example,

domain packets range between 0ms and 0,5ms in time.If a packet with 5000ms comes as it is the case in our implementation, it is not so effective to use k-means-clustering or its variations.In fact, we need a dense located structure rather than clustering into equal pieces.To observe the packet sequence durations we thought that a dense structure is more meaningful because the observation sequence is quite small in size.

In the second part, we analyzed the traffic per server that is in every 100 packets, which users send what kinds of packet sequences.The HMM models are the same, histograms of the durations are the same.However only difference is the clustering used.Now, we are using two-dimensional data to cluster our training data.

Emprical Results

As stated previously, we first analyzed each hundred SMTP, NNTP, LOGIN, TELNET and DOMAIN packets captured from network traffic.First, we used only one parameter, duration to detect protocols.The results were actually promising.However we realized that only duration is not sufficient to protocol detection.This is because some protocols dominate others in specific regions and some protocols spread over the entire intervals weakly.For example nntp, telnet and SMTP dominated LOGIN in various regions.At our first experiment, we observed that dominant protocols like DOMAIN and NNTP give better results like %90.However other weak protocols give nearly about %0 because they are spread in the region where dominants are effective. Below is the confusion matrix for first part of our analysis.(Left is the real identity of protocols.)

	Login	NNTP	Domain	SMTP	Telnet
Login	0	269	1	157	100
NNTP	0	1843	0	16	204
Domain	0	66	581	0	0
SMTP	0	7705	290	196	17527
Telnet	0	922	0	0	1341

Confusion Matrix

Protocol	**Correctly Detected**
LOGIN	%0
NNTP	%89,34
DOMAIN	%89.8
SMTP	%0,07
TELNET	%59,3

Detection Rates

As it is seen above the results are promising.In fact one might understand from the histograms that some of the protocols would be detected wrong.However, we thought that this will draw a roadmap for us to select more appropriate features.

Performing such traffic analysis requires, at the least, accurate models for common network protocols, using no more information than a packet's size, timing and direction.Here we presented our first attempt at building such a model and demonstrate our early success in applying our models to protocol identification.

After these promising results we decided to cluster the protocols two dimensionally.We first used features duration and bytes sent.However we observed that the histogram just

becomes more complicated in two dimensional analysis.That is not actually surprising because duration is the time packets took in the transmission environment.The same case apply when we look at the bytes received by the client.However features like flags, location information and destination server ip are more realistic features to consider.Therefore we decided to analyze the behaviour using packet destination and duration packet sequences took in the transmission channel.

The results were exactly what we want we got nearly %100 detection of the five protocols.Below are the scatters for the data and confusion matrix.

	Login	NNTP	Domain	SMTP	Telnet
Login	437	0	0	36	54
NNTP	0	2063	0	0	0
Domain	0	0	647	0	0
SMTP	1348	0	0	21462	4256
Telnet	321	0	0	430	1512

Confusion Matrix

Protocol	**Correctly Detected**
LOGIN	%84,5
NNTP	%100
DOMAIN	%100
SMTP	%79,2
TELNET	%66,8

Detection Rates

Future Work

Inspite of the fact that our approach worked completely in this case, generally factors may not be as smooth as we presented here.Most of the time there is not enough information to train HMM's and most of the time protocols are not possesing distinct charactheristics like these five protocols.For example to seperate whois and finger protocols is much more problematic.

HMM's may not be trained like our situation all the time.In case of rare data problem one should allow the HMM's to be trained unbalanced but this will cause specific protocols to dominate others.In our example we trained the models equally.The other point worth to mention about is the memorization problem.When we want to detect user behaviour, two features we presented here may not be sufficient.The easy answer coming to mind is to cluster the data more than two dimensions.However this causes the HMM's to memorize cases and an HMM will not be able to detect as we want.This actually is also the situation in our approach when selecting ip numbers for clustering.If some servers shut down in a period of time and enter with other IP's, we may get incorrect results.But this is in control of network adminstrator and we omitted this in our approach.

Currently, we are working to model all protocols in network with HMM's using only information level data.(Packet size, duration, etc...)Another concern is detecting multistage attack patterns of users to a Local Area Network.Also we are trying to paralelize the multiple training operation with operating system threads to fasten the HMM MultipleTrain reponse time.

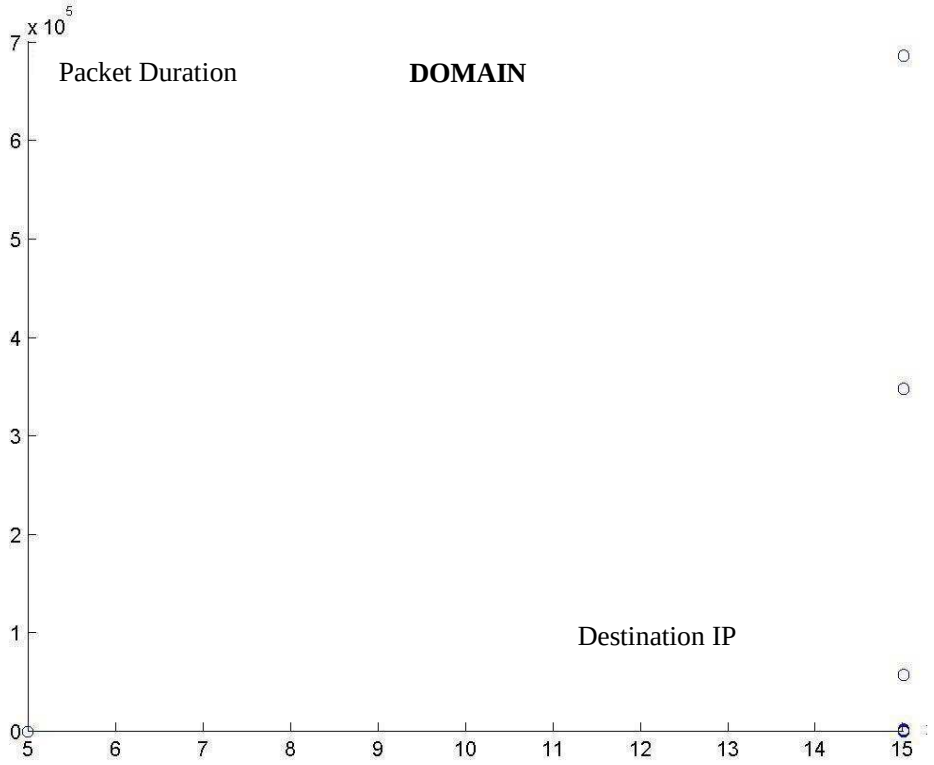


Figure-8

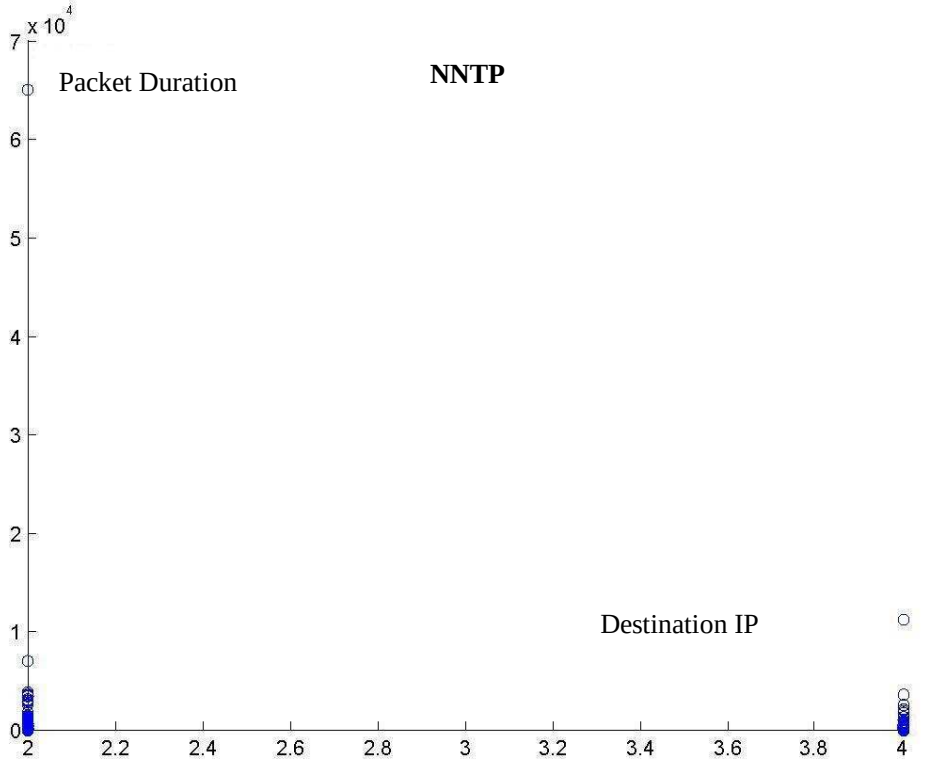


Figure-9

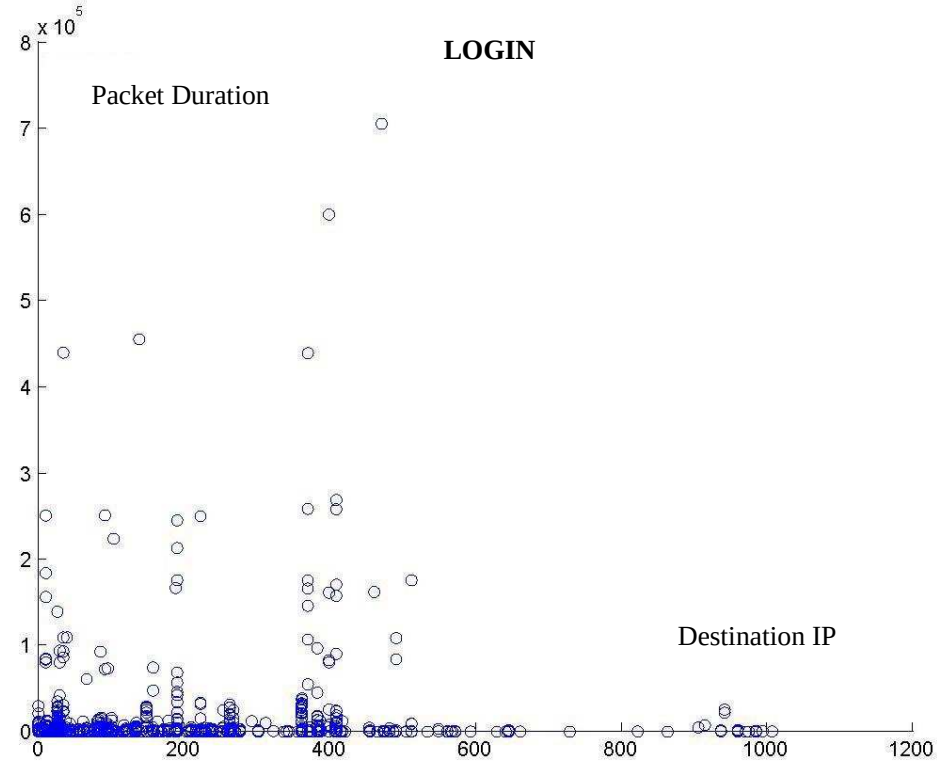


Figure-10

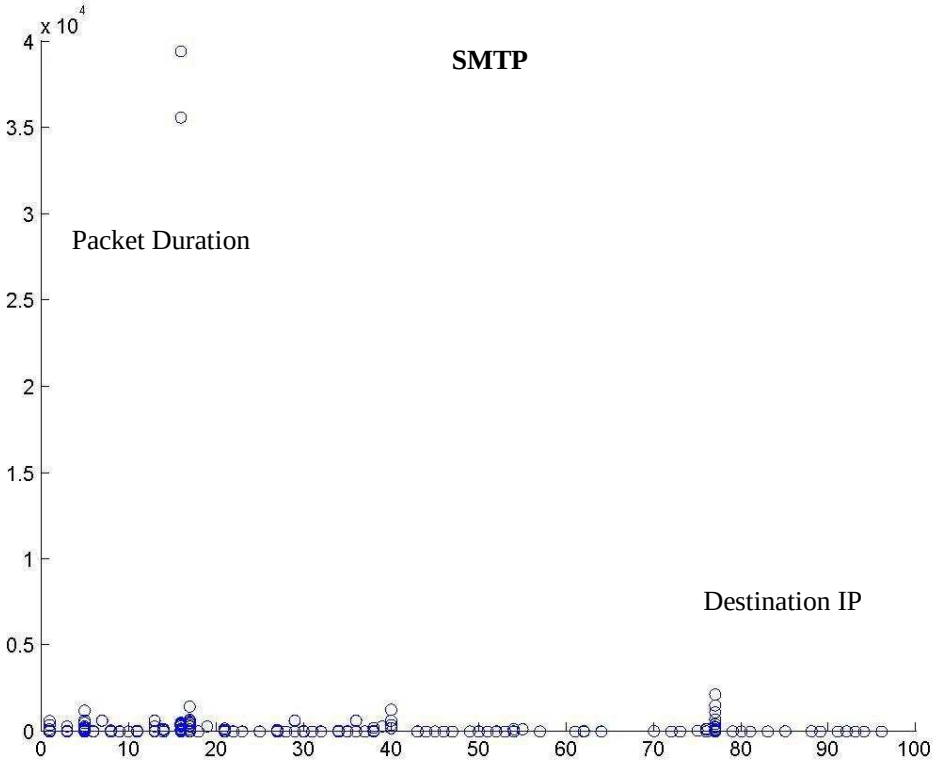


Figure-11

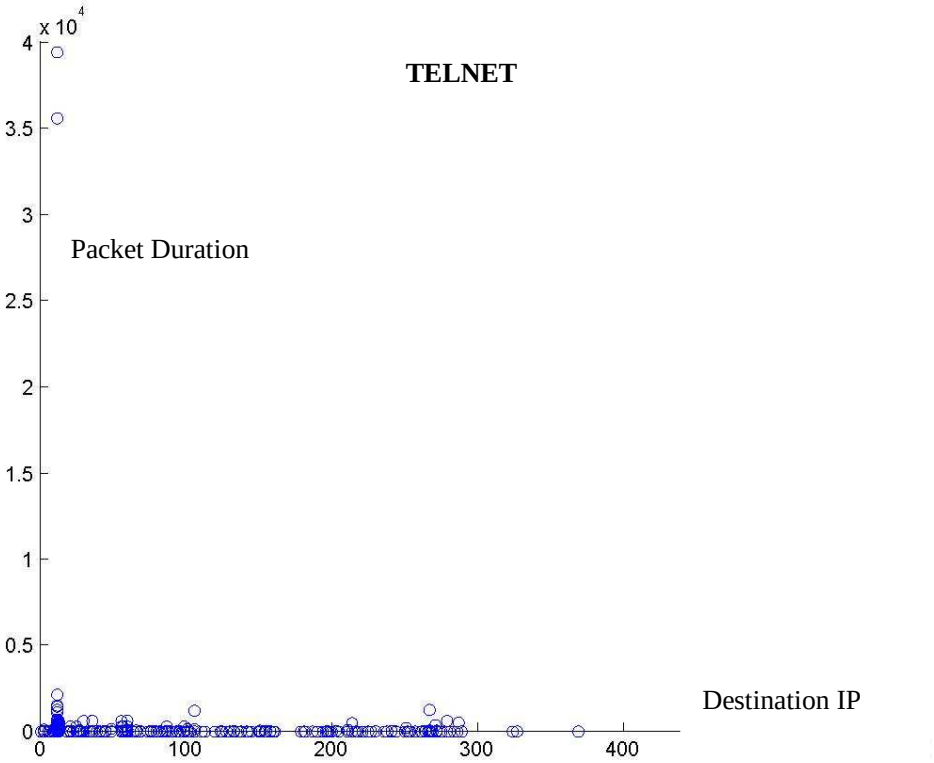


Figure-12

References:

[1]Ourston Dirk, Matzner Sara, Stump William
"Applications of Hidden Markov Models to Detecting Multi-Stage Network Attacks"
Proceedings of the 36th Hawaii International Conference on System Sciences

[2]Early P. James, Brodley E. Carla, Rosenberg Catherine
"Behavioral Authentication of Server Flows"
Proceedings of the 19th International Conference on System Sciences

[3]Zanero Stefano
"Behavioral Intrusion Detection"

[4]Arlitt F. Martin, Williamson L. Carey
"Web Server Workload Characterization"
SIGMETRICS '96 PA, USA

[5]Rabiner L. R., Juang H. B.
"An Introduction to Hidden Markov Models"
IEEE 1986

[6]Rabiner L. R.
"A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition"
IEEE 1989

[7]Wright Charles, Monrose Fabian, Masson M. Gerald
"HMM Profiles for Network Traffic Classification (Extended Abstract)"
VizSEC/DMSEC'04, October 29, 2004, Washington, DC, USA.

[8] Menasce Daniel A., Almeida F. Virgilio
"Capacity Planing for Web Services"
Prentice Hall, ISBN:0-13-065903-7

[9] Tanenbaum A.
"Computer Networks"
Prentice Hall, ISBN:0-13-394248-1

[10] Duda O. Richard, Hart E. Peter, Stork G. David
"Pattern Classification"
Wiley, ISBN:0-471-05669-3

[11]DiscreteHMM library
www.cs.yorku.ca/~cs211169/4080/ summer/4080web/code/SparseDiscreteHMM

[12]Web Trace Archive
http://ita.ee.lbl.gov/html/contrib/LBL-CONN-7.html

[13]Frank E., Witteb I.
"Data Mining"
Morgan Kaufman Publishers, ISBN:1-55860-552-5