TELEMETRY NETWORK INTRUSION DETECTION SYSTEM

Authors: Nadim Maharjan and Paria Moazzemi
Advisors: Dr. Richard Dean, Dr. Farzad Moazzami and Dr. Yacob Astatke
Department of Electrical and Computer Engineering
Morgan State University
namah1@morgan.edu, paria_moazami@hotmail.com

ABSTRACT

Telemetry systems are migrating from links to networks. Security solutions that simply encrypt radio links no longer protect the network of Test Articles or the networks that support them. The use of network telemetry is dramatically expanding and new risks and vulnerabilities are challenging issues for telemetry networks. Most of these vulnerabilities are silent in nature and cannot be detected with simple tools such as traffic monitoring. The Intrusion Detection System (IDS) is a security mechanism suited to telemetry networks that can help detect abnormal behavior in the network. Our previous research in Network Intrusion Detection Systems focused on "Password" attacks and "Syn" attacks. This paper presents a generalized method that can detect both "Password" attack and "Syn" attack. In this paper, a K-means Clustering algorithm is used for vector quantization of network traffic. This reduces the scope of the problem by reducing the entropy of the network data. In addition, a Hidden-Markov Model (HMM) is then employed to help to further characterize and analyze the behavior of the network into states that can be labeled as normal, attack, or anomaly. Our experiments show that IDS can discover and expose telemetry network vulnerabilities using Vector Quantization and the Hidden Markov Model providing a more secure telemetry environment. Our paper shows how these can be generalized into a Network Intrusion system that can be deployed on telemetry networks.

KEYWORDS

Intrusion Detection System, Vector Quantization, K-means Clustering, Hidden Markov Model, Security, iNET.

1. INTRODUCTION

In response to the rise in network security threats and vulnerabilities, we recommend new strategies for iNET (Integrated Network Enhanced Telemetry) to address network vulnerabilities. A Network is just medium transporting telemetry information from Test Articles (TA) to Ground Stations (GS). Telemetry consists of multiple radio links connecting TA and GS with radio

network (RfNET). This paper presents the network security risks and security features required in the iNET environment [1]. It may be hard to penetrate into the network without modifying a firewall or router with protection such as Zone Alarm, AVG, and MacAfee. Even so there is a possibility of being attacked in a well-planned network by increasingly sophisticated attackers or hackers. Therefore, we propose the use of a novel Hidden Markov Model (HMM) for detecting attacks and anomalies in the iNET network.

Figure 1 represents the pictorial representation of the proposed iNET network. This represents the current iNET network design encrypted at the radio link which protects the data from outsiders but does not restrict the insiders or attackers in the Vehicle Network (vNET) or Ground Network (gNET).

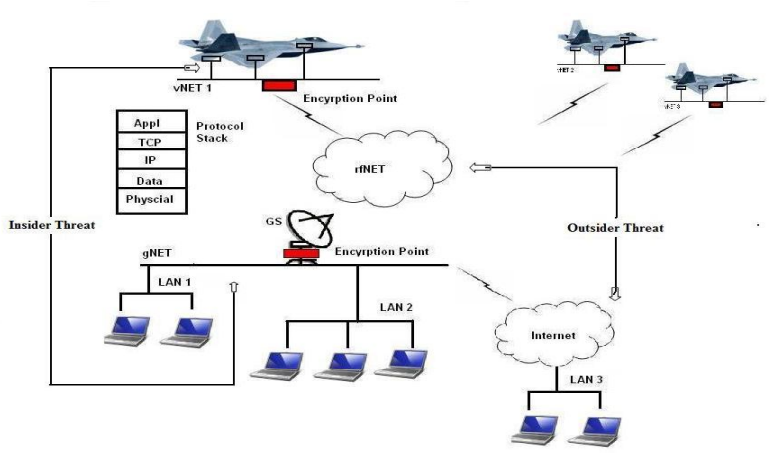


Figure 1: Proposed iNET Security Architecture Design [1]

For that reason, it is necessary to focus on firewalls and network intrusion techniques that will minimize and counter attacks against the network.

2. RELATED WORK

2.1 Network Analyzer

Network monitoring is a system that monitors the network and notifies the network administrator. The Network Wireshark Analyzer was chosen to capture network packets and display packet data details [2]. The Wireshark Sniffer tool has some rich features that help to inspect the packets and to show the layering in the network. In this work Wireshark v1.6.2 was used for the experiments for 'Password' attacks and 'SYN flood' attacks.*Filezilla* software was used to generate "Password' attacks. *Filezilla* is a two way logging communication between server and client which consist of commands send by the client and the server replying those commands.*Engage Packet Builder v2.2.0* was used to simulate the SYN flood attack.

2.2 Network Attacks

Any network can be vulnerable to attacks or unauthorized activities without proper protection. This paper presents two types of attacks:

i) FTP Password Attacks
ii) Syn Flood Attacks

2.2.1 Password Attack

Currently, passwords are used as a means of authenticating users as a convenient way to access systems. Attackers try to guess or crack the password in order to get access into the network. Therefore, attacking passwords is one of the most straight forward attack vectors. The password attack is the most common attack where an attacker or hacker can gain unauthorized access to network and hence confidential information. As shown below the IDS plays an important role in detecting password failure attempts and alerting the administrator or test center.

2.2.2 SYN flood attack

A SYN Flood Attack is one of the Denial-of-Service attacks which exploits the use of the buffer space in the TCP/IP protocol and which sends large amounts of TCP connection requests faster than a computer can handle them.

Normally there are three steps for TCP/IP handshake Protocol.

i. The Client (attacker) initiates the connection by sending the "SYN packet" to the server.
ii. The server (victim) responds back to the client by sending "SYN/ACK packet".
iii. The Client acknowledges the "SYN/ACK packet" by sending the server with an "ACK packet".

2.3 Vector Quantization

Vector Quantization is a simple training algorithm which is used for data compression by reducing the entropy of the data of the network. Our Vector Quantization uses a 2 stage modified K-means clustering, which is a method of cluster analysis. The piece of the K-means clustering algorithm that consumes the most time is the computation of the nearest neighbors where a new centroid is chosen and is used to replace the previous centroid. The majority of the centroid computation time is spent during the last few runs where the algorithm takes a significant amount of time to converge when the centers are very close to their final locations [3].At this point, the data is normalized and clustered with distance measures representing time and attacks.

2.4 Hidden Markov Model

A Hidden Markov Model (HMM) is an extension of the Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states [5]. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the states are not directly visible but outputs, dependent on each state, are visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states [6]. Traditionally, HMMs have been defined by the following quintuple:

$\lambda = (N, M, \pi, A, B)$ (1)

Where,

N = number of states for the model

M = number of distinct observations symbols per state S, i.e. the discrete alphabet size.

A = [NxN] state transition probability distribution given in the form of a matrix Aij = {P(Si/Sj)}

B = [NxM] observation symbol probability distribution given in by Bjk = {P(Bj/Sk)}

π = [1xN] initial state distribution vector $\pi = \{\pi i\}$

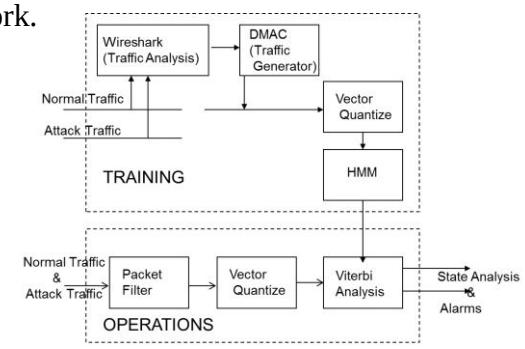
The structure parameters M and N can be taken out to represent the model in the more commonly used compact notation
$\lambda = (\pi, A, B)$.
The HMM includes the Baum-Welch Algorithm and Viterbi Path Algorithm where the Baum-Welch Algorithm determines a likely previous condition of a system by multiplying different combinations of the probabilities of the system being in any of the defined states and Viterbi algorithm works to choose the most appropriate state sequence that maximizes the likelihood of the state sequence for the given observation sequence [6].

2.5 Intrusion Detection System (IDS)

An IDS is considered an effective line of defense that protects the networks from inside and outside attacks or thefts of any valuable data from the network. Since an attacker can be an insider or outsider, we can place the sensors in the vNET and gNET in the iNET environment. The main idea of placing IDS sensor in the network is to establish the perimeter of the network and identify all the possible entry points to the network. The sensors collect protocol based network packets in the network and individual system activity and forward this data to an IDS Server. If a log file pattern is consistent with a possible break, an alarm is triggered and alerts are sent to network administrators. The example of an IDS trigger which we address in our experiment is successive login failures of a network host in a short time frame and continuous flow of SYN flood packets to a particular host. A key feature of IDS is notifying network administrators soon after a compromise occurs. IDS sensors can be placed in workstations, servers, switches, routers, or other network devices.The network level IDS are not designed to track internal attacks, it is used for input and output traffic to trace attacks between nodes, an intelligent system level IDS sniffer is needed for each individual nodes. To ensure log files reach the IDS server, network components with IDS sensors often have additional network cards and/or redundant routes to the IDS server [1]. The IDS must scale well to handle the volume of network traffic or it will drop packets and might miss an attack.This is why we have proposed the Hidden Markov model discussed later in the paper.

3. METHODOLOGY

The proposed strategy applies K-means clustering and the HMM to detect intrusions in the network.



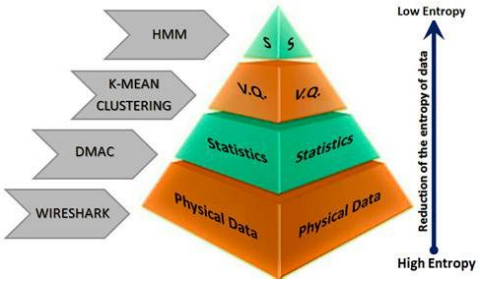


Figure 2 (a): Data Flow Diagram of Design Approach (b) Reduction of the entropy of Data

This paper presents IDS that monitors network traffic, parses the traffic into data streams, and performs the vector quantization and the Viterbi analysis. The figure 2 (a) shows the data flow diagram of IDS whereas the figure 2 (b) shows pyramidal representation of reduction of the entropy of the data. This data flow diagram shows how operational data will be transformed into features that can be used to organize data into Markov States.In the training section, a database is developed that represents a rich set of normal operational network traffic and a subset of attack data where Wireshark Network Analyzer sniffs all live network traffic including normal and attack data. The volume of the network traffic data is so big that it needs to be filtered by reducing the entropy of the data. Entropy is defined as the measure of the uncertainty of a random variable [4].

As a result, it organizes and simplifies the data necessary to answer the question "Is the network under attack" [1]? The exported raw data from Wireshark is processed, evaluated and manipulated in the Data Management and Analysis Center (DMAC) in order to bring high quality data using Microsoft Excel and Microsoft Visual Basic. In other words, the traffic is analyzed so as to enable a synthetic traffic generator to create combinations of normal and attack data that fit the statistics of the data and the requirements for training data for the Hidden Markov Model. The filtered data from DMAC is fed into Vector Quantization (V.Q.) for further reduction of the entropy of the data where the data will be mapped into an N dimensional Vector set which will capture the variations seen in the data and provides a discrete set of candidate packets for analysis. In addition, a two stage modified K-means algorithm is used in V.Q. compressing the data by using iterative refinement techniques which finalizes the final centroids of the clusters. Last in sequence but not least in importance, the HMM plays a vital roleindetecting the states with normal data and attack data.

4. EXPERIMENTS AND RESULTS

4.1 Physical Data

Physical Data Model represents the data design of all the data in the network. Wireshark captures the data and analyzes the physical layer of the network connection. This paper presents two attacks ("Password attack" and "SYN flood attack") detected in Wireshark Network Analyzer. The figure 3 represents the pictorial view of both attacks in Wireshark Network Analyzer.

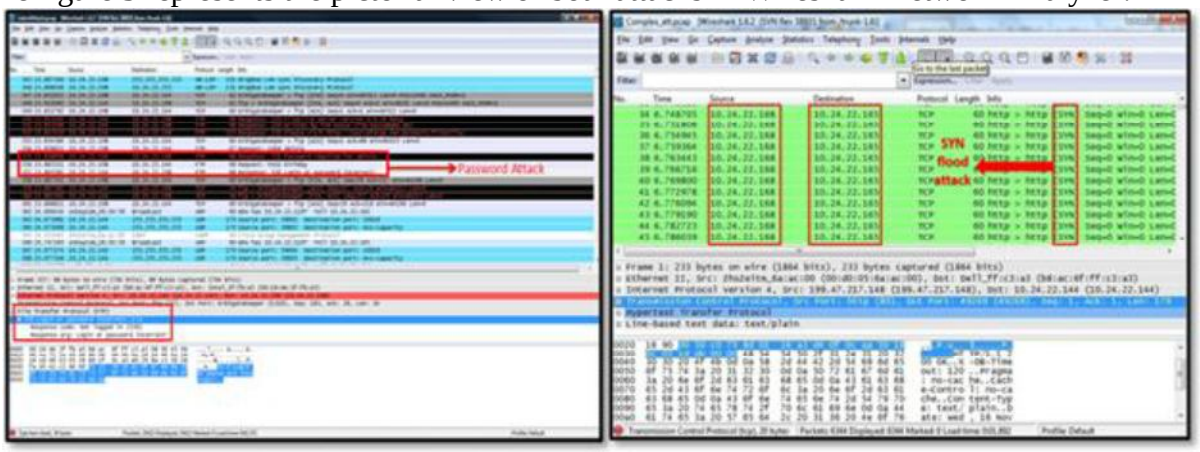


Figure 3: Snapshots of Password attack and SYN flood attack

4.2 Data Management and Analysis Center

The Data Management and Analysis Center is the system where data is collected, processed, evaluated and manipulated. For this experiment, Microsoft excel 2010 was used to manipulate, interpret and analyze the data. The following table 1 represents the data fields of the database that was used in this experiment.

Table 1: Data fields of the database

Column	Data Type
1	Sequence
2	Time
3	Source IP
4	Destination IP
5	Protocol
6	Length
7	Info

Table 2: Flags for Network Traffic Data

Attack/Data	Flags
Normal Traffic Data	0
"SYN flood"	1
"Password"	2

Column 7 has key information about the packets. For example, if there is any attack ('Password' or 'SYN flood' attack) in the network, it can be seen in this column, as seen in Wireshark, in figure 3. It is possible but time-consuming to drag out each line in order to detect the attacks. Therefore, we use Visual Basic (VB) programming which makes much easier and provides the results quickly. To put it simply, using VB in excel can help to automate repetitive tasks. However, we wrote new algorithms in VB code by analyzing statistical characteristics of the data. While analyzing the behavior of the data, flags were set up for each attack to support testing. Table 2 represents the flags for each attack and the normal traffic data.

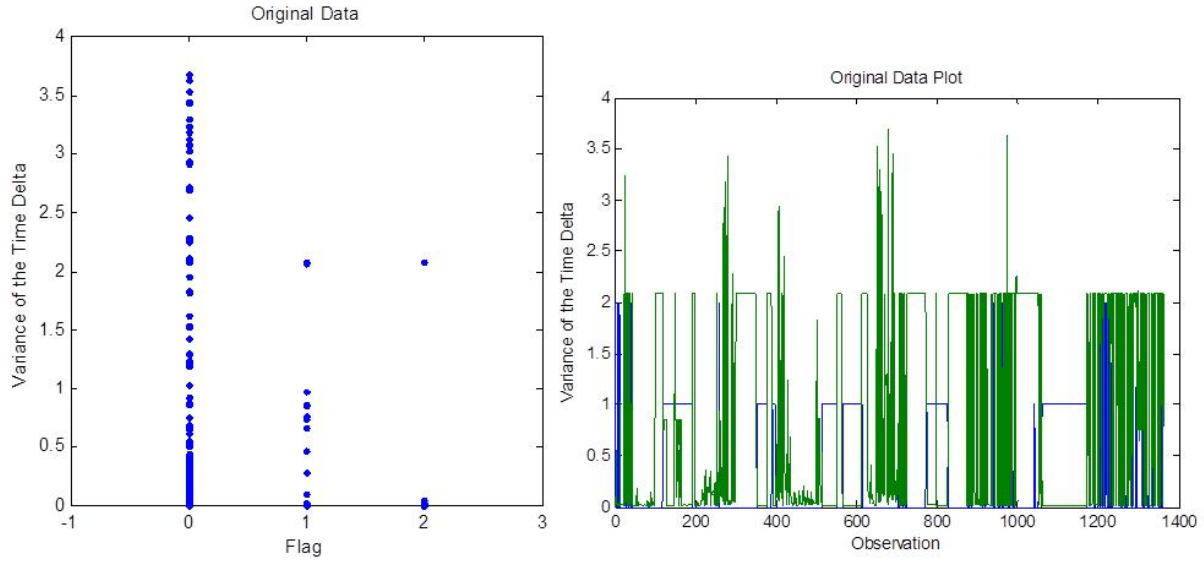


Figure 4: (a) Scatterplot Plot for flags (b) Original Data Plot

Besides generating the flags for the network data, the time delta was calculated as well. Time delta is the time difference between the first packet and the second packet in the network. Wireshark captures the live network traffic data running over the time by measuring numerous

discrete time calculations in order to simulate the smooth behavior of the network. At this time, Data Management and Analysis Center defines the time delta, "SYN flood" attack and "Password" attack which is clearly explained in the above figure 4 (a) and (b).

4.3 Vector Quantization

Vector Quantization is a complex and challenging structure that is key to the design. The Vector Quantization dramatically reduces the entropy of the data by identifying critical fields within the data packets as individual dimension in an N dimensional space. The K-means clustering algorithm is one of the most widely used clustering algorithms that is based on an iterative scheme for minimizing the mean squared distance between each point and its nearest center point in the cluster also known as the cluster centroid. The centroid is obtained by computing the average distance between all points in the cluster. The general K-means clustering algorithm is shown in Table 3[3]:

Table 3: K-means Clustering steps

Step 1: choose the number of clusters k
Step 2: Randomly generate k cluster and assign random centers for each
Step 3: Classify or assign each node to the closest center
Step 4: Re-compute the new centroid for each k cluster
Step 5: Repeat step3 and 4 until convergence criteria is met (centroids do not change anymore)
Step 6: Return the final values of the centroids for each cluster.

The figure 5 represents the scatterplot of randomly generated cluster. The K-means clustering continues until a final convergence criterion is met so that it returns the final value of the centroids for each cluster in the network as shown in figure 6. There are nine centroids to capture the whole data in the network demonstrating one dimensional cluster that represents two-dimensional space.

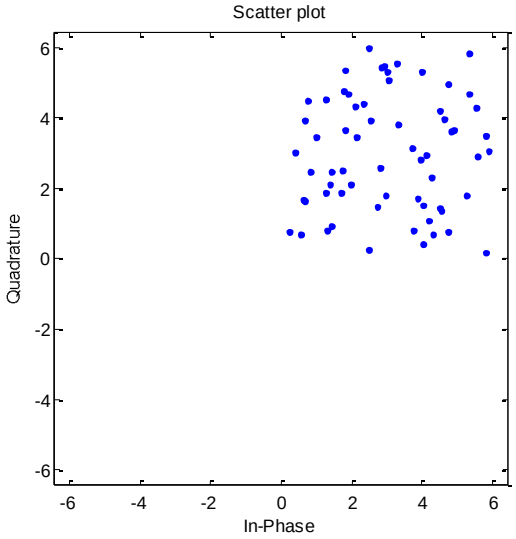


Figure 5: Scatterplot of randomly generated cluster

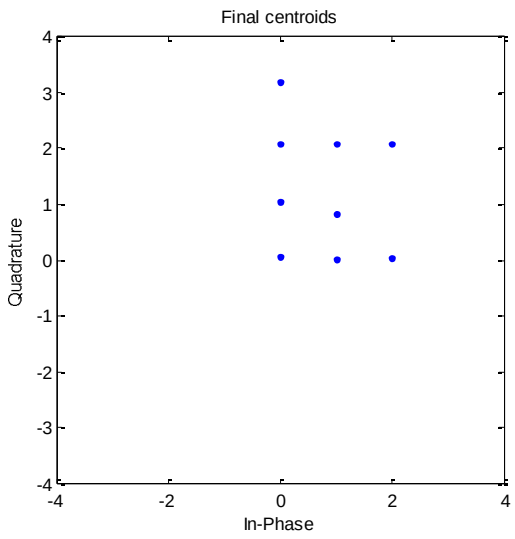


Figure 6: Scatterplot of the Final Centroids

Figures 7 and 8 characterize one dimensional data for each centroid in different illustrations:

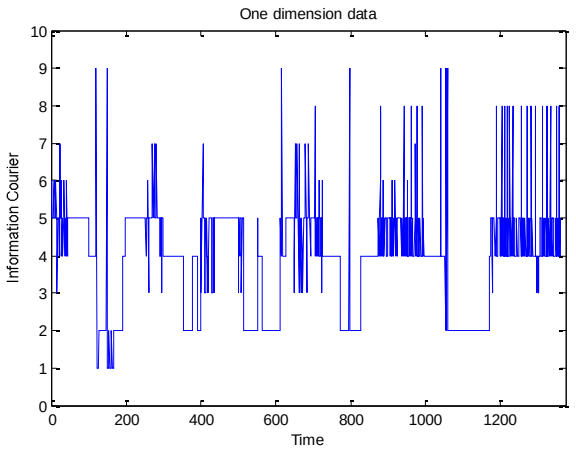


Figure 7: VQ Output: One Dimensional Data

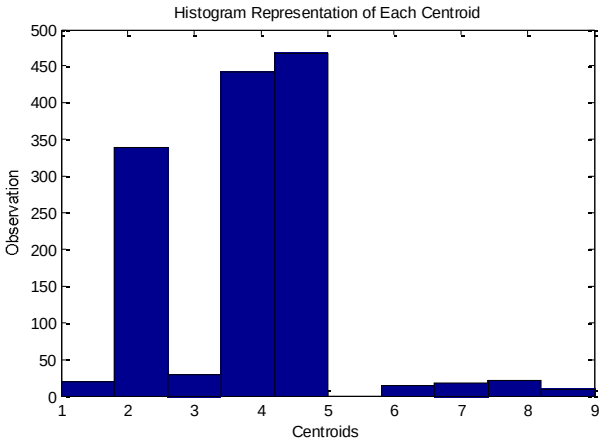


Figure 8: Histogram Representation of Data

4.3 Hidden Markov Model (HMM)

The Hidden Markov Model is "hidden" because the parameters of the model are initially not known. The model has to be "trained". The output data from VQ is the training data for the HMM. The Hidden Markov Model will enable us to develop a statistical profile of the traffic network. Using coded algorithms we will analyze the traffic information obtained from the network packet fields. The HMM toolbox developed at MIT was adapted to this project. This HMM toolbox incorporates both Baum-Welch's Algorithm and Viterbi Algorithm which have all the features needed to accomplish the target of our project. Therefore, given a series of observation data and some random HMM inputs to start from, the HMM toolbox should be able to estimate the most likely state transitions (Viterbi Path) as well as optimized HMM parameters [1]. During Vector Quantization, the original data were quantized into new centroids. This training data is now introduced into the HMM. Although Baum-Welch maximizes the log-likelihood for training sequences, it can over-fit the training sequences and produce inferior results compared with HMMs that have experienced fewer training cycles [7].

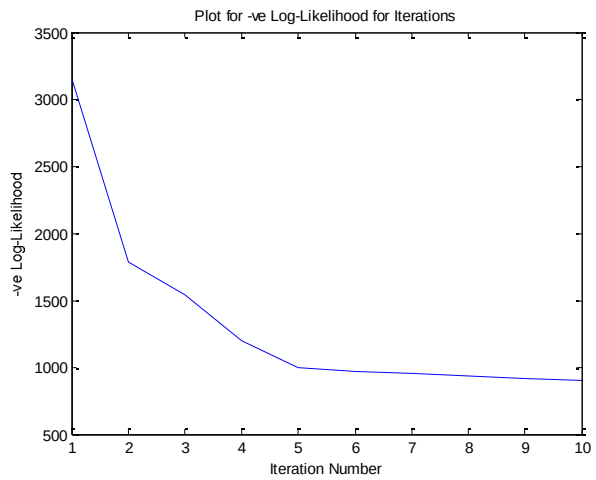


Figure 9: Log-likelihood vs. iteration

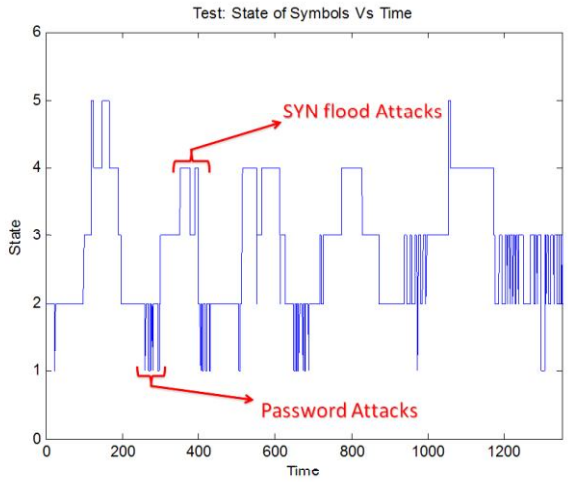


Figure 10: 5-State in HMM identifying attacks

Figure 9 shows the negative log-likelihood versus the number of Baum-Welch iterations in an experiment. In this instance the best generalization performance was found after 5 iterations. The training data obtained from the VQ was fed into HMM. The training data goes through the Baum-Welsh algorithm which optimizes the model parameters. Now,the Viterbi algorithm uses these model parameters to determine the states of the network as seen in Figure 10. Figure 10 highlights both attacks (Password attack and SYN flood attack) with their respective states 1 and 4. Password attacks are more likely to seem like impulse responses because the interval for password login failure attempt was really small based on time delta. The result emits the discrete output of Password attack. Since it is already known that the SYN flood attack normally sends a large amount of TCP connection requests faster than a computer can handle, HMM traces continuous pattern of those SYN packets and pulls out the results with help of Viterbi algorithm.

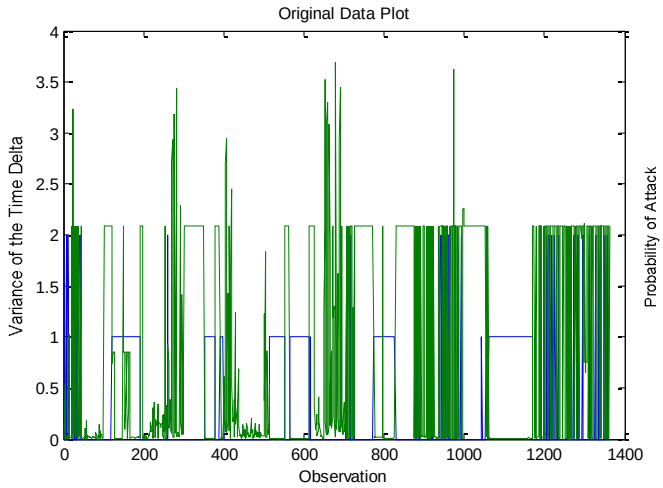


Figure 11: Original Data Plot

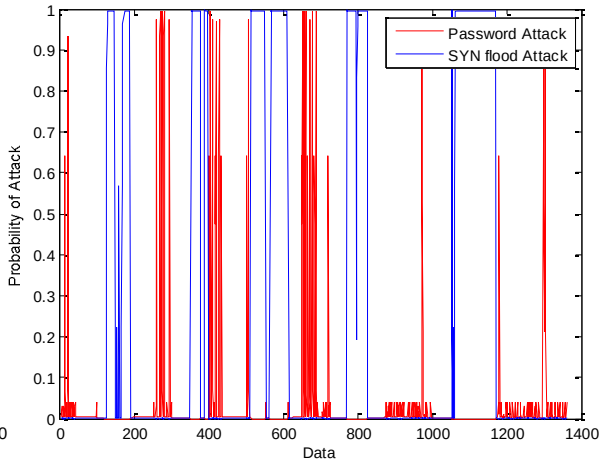


Figure 12: Probability of Attacks

We now verify the probability of attack states with original data plot. Figure 12 exhibits the probability of attacks ("Password" attacks and "SYN flood" attacks) at State 1 and State 4 respectively. Comparing probability of attacks and original data plot, it can be plainly seen that patterns of traces of the attacks by HMM on figure 10 matches with the patterns of section of the probability of attacks and original data plot. While comparing and analyzing with the original data, it is much easier to say that the state is under attack or not. Furthermore, when the data is analyzed by a Hidden Markov Model,it will find Markov behavior in the traffic with A and B matrices. The A matrix will identify the state transition probabilities for the K Markov states. The B matrix will characterize the probabilities of specific vectors given the states.

5. CONCLUSION

To recapitulate, the proposed IDS discovered and exposed the telemetry network vulnerabilities to the "Password" attack and "SYN flood" attack using Vector Quantization and the Hidden Markov Model providing a more secure telemetry environment. This paper showed how these can be generalized into a Network Intrusion System which can be deployed on telemetry networks for many more attacks. To run through the experiment, the live network packets were

captured in Wireshark Network Analyzer. Subsequently, data were collected, processed, evaluated and manipulated in the Data Management and Analysis Center with help of Microsoft Excel and Microsoft Visual Basic. In the Data Management and Analysis Center, time delta, "Password" attack, and "SYN flood" attack were flagged into different categories. The data from Data Management Analysis Center was fed into Vector Quantization so as to compress the data by reducing the entropy of the data. The final centroids were calculated based on old centroids and fed into HMM. The training data from HMM was passed into the Baum-Welsh algorithm which optimized the model parameters. Then, the Viterbi algorithm used these model parameters to determine the states of the network. As a final point, a statistical method for these attacks ("Password" attack and "SYN flood" attack) were proposed, tested, and detected in IDS. When IDS hit an alarm detecting these attacks, the network administrator could be notified. This work successfully extended earlier efforts that identified single attacks. Future work will look towards extending this process to large sets of data, protocols and attacks.

6. ACKNOWLEDGEMENTS

The authors would like to express appreciation to TRMC, SRC and CSC for their support for this effort and heartfelt gratitude to our advisors, Dr. Dean, Dr. Moazzami, Dr. Astatke and my colleagues, Abiola Odesanmi and Sandarva Khanal for providing such fruitful discussions, comments and support.

7. REFERENCES

[1] Abiola Odesanmi, "Secure Telemetry: Intrusion Detection Engine with Hidden Markov Model," Morgan State University, Baltimore, MD, Final Project Report July 2011.

[2] Richard Sharpe & Ed Warnicke, "Wireshark User's Guide," Ulf Lamping, 2011.

[3] YacobAstatke, "Quality of service Management in Mixed Wireless Networks using the Power Performance Measure," Morgan State University, Baltimore, PhD Thesis 2010.

[4] Thomas M. Cover and Joy A. Thomas, Elements of Information Theory, 2nd ed. Hoboken, New Jersey: John Wiley and Sons, 2006.

[5] Lawrence R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, FEB. 1989

[6] Sandarva Khanal, "Aeronautical Channel Modeling for Packet Networks," Morgan State University, Baltimore, MD, Final Project Report May 2011.

[7]Kyoung-Jae Won, Adam Prügel-Bennett & Anders Krogh, "Training HMM Structure with Genetic Algorithm for Biological Sequence Analysis," University of Southampton, Denmark. <http://eprints.pascal-network.org/archive/00000961/01/paper.pdf>