SECURE TELEMETRY: ATTACKS AND COUNTER MEASURES ON INET

Authors: Abiola Odesanmi, Daryl Moten
Advisor: Dr. Richard Dean
Morgan State University Baltimore MD 21239

ABSTRACT

iNet is a project aimed at improving and modernizing telemetry systems by moving from a link to a networking solution. Changes introduce new risks and vulnerabilities. The nature of the security of the telemetry system changes when the elements are in an Ethernet and TCP/IP network configuration. The network will require protection from intrusion and malware that can be initiated internal to, or external of the network boundary. In this paper we will discuss how to detect and counter FTP password attacks using the Hidden Markov Model for intrusion detection. We intend to discover and expose the more subtle iNet network vulnerabilities and make recommendations for a more secure telemetry environment.

KEY WORDS

Hidden Markov Model, iNET, Telemetry, Intrusion Detection System (IDS)

INTRODUCTION

Previously telemetry has been a point to point system. Now the iNet project is attempting to connect the entire telemetry environment, creating a network out of individual pieces that include the test articles (TA) and the ground stations (GS). In this paper we explore the network security risks and security features needed for the proposed environment. Military platforms still have to be protected physically but they are mostly run by software which can be hacked into, compromised or corrupted. It is a necessity to protect the integrity of all iNet telemetry communications with the proper government approved information assurance technology for such sensitive information. Preliminary designs show security implemented in iNet as only data encryption across the radio link. Data within the test article network and on the ground station network are also vulnerable to network attacks. There is then a need to understand how these attacks and losses take place within the iNet network, how to stop them, and how they are used against the network. According to [1], when considering malicious software attacks, the attacker needs only to put in effort in the development of the software, which can then do damage over and over again with no more effort from the attacker. Attackers need only to be good at one or two exploits, but defenders need to continuously identify and address all of the vulnerabilities in the network environment. To secure the data, the current network design places encryption points at the connection with the radio interface. This protects the data from outside adversaries

but does not restrict unauthorized inside users from being able to access information from within the Vehicle Network (vNET) or inside the gateway network (gNET). With the network elements exposed by the iNET design, we need to focus on firewall definitions and network intrusion detection techniques that will minimize and counter attacks against the network.

BACKGROUND

This paper operates on the premise that even networks protected by good security can be compromised by insiders by accident or by design, and by outside adversaries with unlimited resources and a wealth of opportunities. In this case the Intrusion Detection System represents the last line of defense in protecting these sensitive systems. The focus of this work is to demonstrate IDS methods based on the HMM that might be expanded to be a comprehensive solution to these threats. It is useful to describe a few of the attacks that might be accomplished against the iNET system.

Networks are designed to provide many services to multiple users. Network administrators must maintain those services for legitimate users and guard against a variety of network attacks. The three main categories of network attacks are Denial of Service (DoS), passive information gathering, and unauthorized access. A goal of a DoS attack is to deny legitimate users access to a network service(s) by overwhelming system resources. One such DoS attack is a SYN flood, which exploits the TCP-handshake communication. In a normal TCP session, the sender requests a communication session by sending a "synchronize" (SYN) packet to a receiver. The receiver acknowledges the sender's request with a SYN-ACK packet. Finally, the sender confirms and establishes connection with an ACK response. A SYN attack involves the sending of many SYN packets without replying to SYN-ACK packets sent from the receiving network system. The large number of "half-open" sessions would eventually overload the system, rendering the receiving system incapable of replying to legitimate user requests. The SYN exploit is one of many protocol-based vulnerabilities. Information gathering attacks occur after network access has been achieved. This intrusion could be facilitated by undetected software installation from email, flash drives, or network intrusions. Information can be gathered by sniffing network traffic or gathering information on a host, which can be sent to a remote system. Unauthorized access to network systems could be achieved by exploiting vulnerabilities in system software or network protocols. Also, an attacker could use "brute-force" password-guessing software. The focus of this IDS research is password guessing using and FTP client-server setup.

NETWORK DEFENSE

An Intrusion detection system (IDS) is a network defense mechanism that protects networks against inside and outside attacks. These attacks occur with a malicious insider or after the network boundary has been breached by an outsider. Attacks can also occur if a person authorized on a limited part of network attacks network resources to which they have no authorization. Once an intruder is inside a network, individual systems can be attacked and the data in those systems can be viewed or destroyed. Intrusion detection systems collect data using sensors placed at various network locations. The sensors collect data on network and individual system activity and forward this data to IDS servers. The IDS servers analyze this data in search for patterns that are consistent with network intrusion. 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. 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. A crucial aspect of system level IDS is uploading log data to the IDS at regular intervals. Often an attacker is able to break into a network asset and erase log files entries that contain evidence of their presence. In such a case, a network can be attacked multiple times without detection. 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. We propose that the iNet ground station and Test article networks be divided into various subnets. These subnets are physically distinct groups to which network traffic can be regulated. Gateways and routers are used to determine and control the boundaries of subnets. Sub-netting is a recommended feature because it can be used to restrict the visibility of sensitive traffic to limited recipients grouped as a subnet.

Each subnet will need an independent monitor (agent) to observe the subnet's network traffic and watch the unique packets. The network IDS monitor is placed as the first node after the router in the subnet. This way all packet traffic to and from the subnet network will be read effectively. It is possible to suggest that since we have automated security products that identify these kinds if of malicious activities that we no longer have to pay close attention. This is not wise because it is dangerous to just depend upon software that does not have the ability to put the activities in the necessary context and make a commensurate decision [1]. This is why we have proposed a learning model, the Hidden Markov model.

INTRUSION DETECTION SYSTEMS

This section will give a brief introduction to intrusion detection systems and the reason why the Hidden Markov Model has been identified as a viable model for this problem. Intrusion detection is an extra layer of security for computing networks against malicious attacks. An intrusion detection system monitors unusual activities in the system and compares them to previously observed patterns. An intrusion detection system performs misuse detection and anomaly detection to improve the chances of intrusion detection. The observation patterns also called observables [3], that are used in the training data for the model are usually obtained from the network to be protected. They can include logged monitored events like UNIX shell commands, audit events, keystrokes, system calls and network packets. In [4] the author lists five ingredients that IDSs must have to deliver on their objectives. The first is to set intrusion check points to analyze the activity that signify state transitions, from normal to intrusion. The second has to do with the creation of activity profiles; these detect irregular activity by measuring the deviation from regular behavior and serve as signatures of normal activity. The third is a concept drift that measures the variations in user behavior over time. The fourth is the control loop that "adapts the intrusion check point trigger based on the weighted sum of proportional, average and derivate sensor measurements over derivation and integral time window" [4]. Finally the fifth ingredient is the model itself, which predicts the most likely state based on the preceding state and other observed states. Based on the requirements of the fifth ingredient, we identify that the properties of the model match the definition of a Markov process. Which is "A simple example of a stochastic process with dependence is one in which each random variable depends only on the

one preceding it and is conditionally independent of all the other preceding random variables. Such a process is said to be Markov" [5].

MARKOV MODEL

[6] The Markov Model (MM) is a probabilistic, stochastic model in which a system can move from one state to another. Each move is called a transition, and the way the system transitions occur is dictated by the Markovian property. Markov Model is 'stochastic' meaning that the system must be in any of the defined states at any given time, so that the probability that the system will be in any of the defined states is 1. When the states change, transitions have occurred from one of the defined states to another defined state or remained in the same state.

A state is randomly chosen as the starting state using an initial state probability distribution (π) matrix. For any Markov system with N number of states, the π-matrix will be of size [1 x N]. Therefore, for a 3-state MM, π-matrix will be denoted as π = [p_1,p_2,p_3], where, π = probability of the system starting from state i. The probabilities A_{ij}, are called transition probabilities. In the above figure, all the arrows indicate the transition probabilities. These transition probabilities are arranged in a matrix, called the State Transition Matrix (A-matrix). For any Markov system with N number of states, the A-matrix will be of size [N x N]. Therefore, for a 3-state MM, A-matrix will be denoted as

$$A = \begin{matrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & \boldsymbol{A_{13}} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} & \boldsymbol{A_{23}} \\ \boldsymbol{A_{31}} & \boldsymbol{A_{32}} & \boldsymbol{A_{33}} \end{matrix}$$

THE HIDDEN MARKOV MODEL

This section expands on the use of the HMM in IDS modeling, explains the approach and introduces the parameters involved. Hidden Markov models have been found to be useful for ecology, cryptanalysis and speech applications, but the use of HMM for anomaly and intrusion detection is relatively new. Similar to a Markov chain, the HMM has a discrete number of unobservable states. The transitions that take place among the states are controlled by a set of transition probabilities which make up the transition matrix. The difference is that the states can only be inferred from the observations, so the states are hidden. A good question to ask is what properties of the Markov model make it a suitable model for network intrusion detection. [7] explains this stating, many real world processes including networks manifest a rather sequentially changing behavior; the properties of the process are usually held steadily except for minor fluctuations, for a certain period of time, and then at certain instances change to another set of properties. The Hidden Markov Model presents a solution to how these steadily or distinctively behaving periods can be identified, how the "sequentially" evolving nature of these periods can be characterized.

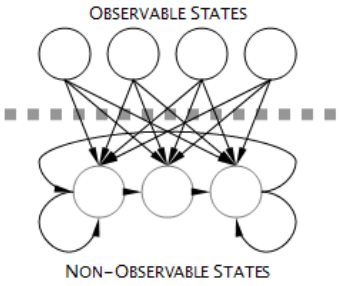


Figure 1: Diagram of hidden Markov model showing non-observable (hidden) states and observable states [3]

The Hidden Markov Model (HMM) will enable us to develop a statistical profile of the traffic network. Using the Viterbi algorithm we will analyze the traffic information obtained from the network packet fields. We selectively look at the packets in traffic, identify the type of traffic, and then transform the data to vectors for use in an HMM as training data. [8] explains that HMM based intrusion detection was previously done using one system call at a time in a trace as an observable and tracking what state transitions and outputs will be required of the HMM to produce that particular system call. The new method described in [8] uses the HMM to model program behaviors and in contrast with first scheme, uses sequences of system calls in a trace as the observables. This is a more attractive way with which to model the IDS because, compared to the aforementioned scheme which tracks individual user behaviors, this traxks program behaviors which are more stable over a time and the range of program behaviors are more limited.

The probability of the HMMs producing a sequence of system calls was computed for anomaly detection. If the probability of a distinct sequence in a trace is below a certain threshold, the sequence is flagged as an anomaly and if the ratio between anomalies and all sequences exceeds another given threshold, the trace is labeled as a possible intrusion. This modeling scheme is preferred as it can be trained and so it identifies intrusion while reducing false positives. "The training problem is a crucial one for most applications of the HMM. It allows us to optimally adapt model parameters to observed training data to create the best models for real phenomenon" [8] Though training an HMM tends to be computationally intensive in the process of modeling program behaviors, testing is more efficient when the model has been built for normal program behaviors [9]. HMM learning can be done utilizing the Baum –Welsh algorithm or forward backward algorithm.

The first challenge in applying the HMM to Intrusion detection is deciding on the number of states the model should have [8]. This must be decided before training and experiments have shown that a reasonable choice of the number of states for an application of the HMM is to choose a number of states that approximately correspond to the number of distinct system calls used by the program. Choosing the number of states then depends on the observation data used in the experiments. The HMM will require training based on observation data and continuous re-assessment, this will create a profile that contains transition probabilities, $A = \{a_{ij}\}$, and observations symbol probabilities represented as $B = \{b_j(k)\}$. The observations probability represents an attribute that is observed with some probability if a particular failure state is anticipated [4] . Finally π , the initial state distribution. A complete HMM sequence model is

represented as $\boldsymbol{\lambda} = \{A, B, \pi\}$. The nature of the hidden process is such that we are not aware of what state transitions are happening at the unobserved states but we can observe the results. For instance a sequence of FTP packets is received and failed login observations are recorded up to T times. Let O represent observations, an observations sequence is generated

$$O = O_1, O_2, \ldots\ldots O_T$$

Given the above experiment, we can develop a number of questions. How do we build an HMM to explain the observed sequence of FTP login codes? Next how do we decide on the number of states needed in this model and how do we choose the state transition probabilities in each state to optimize the model so it rightly describes the observed outcome sequence?

To answer these questions we must solve three problems outlined in [8]. Given the observation sequence O, and the model $\boldsymbol{\lambda}$, how do we compute the probability of the observation sequencePr $(O|\lambda)$. This problem can be solved using the forward backward algorithm which efficiently solves this complex intensive computation. Solving this problem we wish to calculate the probability of the observation sequence O, given the model λ. For every stated sequence $I = i_1, i_2, \ldots i_T$, the probability of the observation sequence O is Pr $(O|I, \lambda)$, where

$$\Pr(O|I, \lambda) = b_{i1}(O_1), b_{i2}(O_2) \ldots b_{iT}(O_T) \ldots (1)$$

The probability of the state sequence I is

$$\Pr(I|\lambda) = \pi_{i1} a_{i1i2} a_{i2i3} \ldots a_{iT-1iT} \ldots (2)$$

The probability that O and I occur together is the product of (1) and (2). The probability of O is next obtained by the summation of the joint probability over all possible state sequences

$$\Pr(O|I) = \sum_{all\ I} \Pr(O|I, \lambda) \Pr(I|\lambda) = \sum_{i_1, i_2, \ldots i_T} \pi_{i1} b_{i1}(O_1) a_{i1i2} b_{i2}(O_2) \ \ldots \ a_{iT-1iT}\, b_{iT}(O_T)$$

Given the observation sequence O, how do we choose a state sequence $I = i_1, i_2, \ldots i_T$ which is optimal is some meaningful way? This problem involves choosing the states which are individually most likely. To implement this we define a new variable $\gamma_t(i) = \Pr(i_t = q_i|O, \lambda)$, which is the probability of being in state q_i at time t given the observation O and the model λ.The individually most likely state is represented as

$$i_t = \underset{1 \le i \le N}{argmax} [\gamma_t(i)] \quad 1 \le t \le T$$

This problem is solved using a formal technique called the Viterbi algorithm which is used for finding the most suitable state sequence. Finally, how do we adjust the model parameters $\boldsymbol{\lambda} = \{A, B, \pi\}$ to maximizePr $(O|\lambda)$. This third problem is to alter the model parameters to maximize the probability of the observation sequence given the model.

$$\xi_t(i, j) = \Pr(i_t = q_i, i_{t+1} = q_j|O, \lambda)$$

This represents the probability of a path being in state q_i at time t and making a transition to state q_j at time t+1. An iterative procedure called the Baum-Welch method is used to re-estimate the values of the HMM parameters.

EXPERIMENT

In this section we will be describing an experiment done in modeling an intrusion detection system for a packet network based on the HMM. The experiment was carried out to detect an FTP password attack on a network. Traffic data was obtained using Wire Shark, a mainstream

network sniffer. For this experiment we have chosen to characterize the data using a two state Markov model indicating a normal state and password attack state.

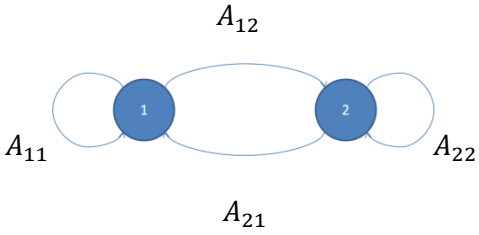


Figure 2: Two State Markov chain representing normal (1) and abnormal (2) states

DATA SET

Two sets of data were collected from the network with the help of the packet sniffer. The first set of data characterized the normal working of the network while for the second set an attack was staged on the network so that the network traffic will indicate the presence of abnormal activity. In creating the testing data, intrusion activities were generated by simulating the intrusion scenario of password guessing. The first set of data was used as the training data set for the HMM while the second set was used as testing data to prove the validity of the model in detecting anomalies. The raw network packet data was then parsed and translated into signatures that can be characterized by our two state Markov model.

IP Sender	Port	IP Destination	Port	Frame number	Time Stamp	Time Delta
10.24.22.150	49160	231.7.134.175	16386	2074	6/21/11 15:53	0.009860039
10.24.22.150	49160	231.7.134.175	16386	2075	6/21/11 15:53	0.020030022
10.24.22.150	49160	231.7.134.175	16386	2076	6/21/11 15:53	4.98295E-05

Figure 3: Training Data Snippet of details of normal network traffic

IP Sender	Port	IP Destination	Port	Frame number	Message	Time Stamp	Time Delta
10.24.22.59	21	10.24.22.198	1805	2090	Response arg: Login or password incorrect!	6/21/11 15:53	0.279700041
10.24.22.198	1805	10.24.22.59	21	2106	Response arg: Login or password incorrect!	6/21/11 15:53	6.98566E-05
10.24.22.150	49160	231.7.134.175	16386	2107	Data	6/21/11 15:53	0.00022006
10.24.22.150	49160	231.7.134.175	16386	2108	Data	6/21/11 15:53	0.000760078
10.24.22.59	21	10.24.22.198	1805	2109	Response arg: Login or password incorrect!	6/21/11 15:53	0.011529922

Figure 4: Testing Data Snippet including details of FTP password attacks

It is observed from the data that usually the future actions of both the attacker and the normal user are related to the last action and over time a pattern can be traced consistently. The nature of network packets make it possible to extract relevant information for this exercise which has been sorted and grouped as displayed in figure 5 and 6. The data then needs to be interpreted by the IDS for use in the detection engine. The algorithm analyses the data in batches and utilizes the frequency of short time intervals between login failures of the attack observation as data. These observations can also be viewed as signatures and serve as data to be used in the training of the

HMM, data processed similarly will also serve as input for the Viterbi algorithm during the testing phase.

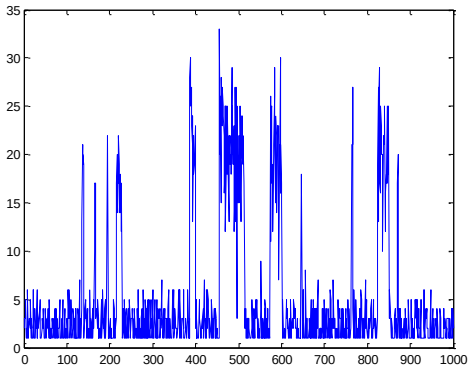
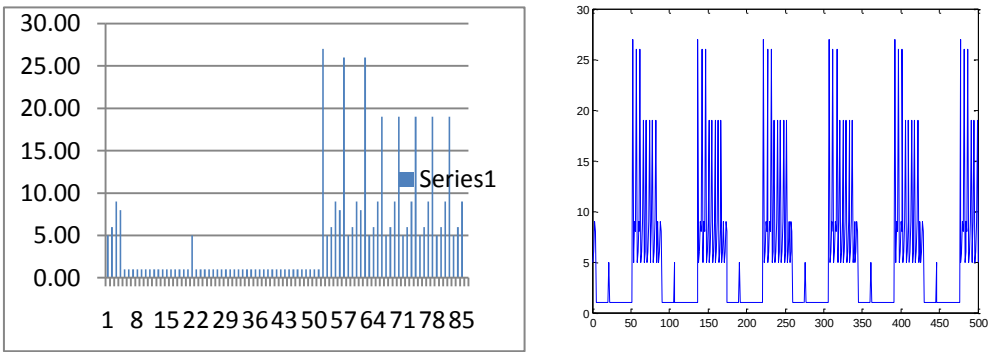


Figure 5: Graph of training data showing the time intervals of similar network activity



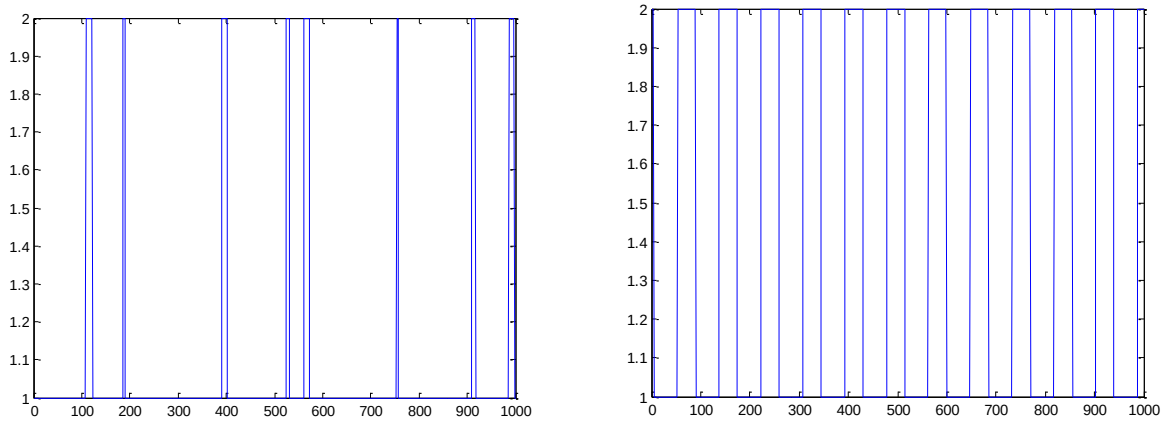
(a) (b)

Figure 6: Graph of data with normal and attack states (a). The data has been repeated several times for use as test data (b)

Elements from the HMM toolbox developed at MIT were used to learn from the training data, these algorithms were implemented using Matlab. We trained the markov model as the normal profile by learning the transition probability matrix and initial probability distribution from the stream of network packet data in figure 7, that were observed during the normal usage of the network.

EXPERIMENTAL RESULTS

A good HMM will assign a greater probability to the normal state, and a significantly lower probability to the attack state. The resultant state transitions by the model are seen in figure 9(a). The testing data was repeated to obtain a lengthier sequence and fed through the viterbi algorithm to determine the most likely states. The result on the right figure 9(b) show the viterbi path estimation of the states



(a) (b)

Figure 7: State Transitions identified by the HMM (a). The Viterbi path estimation of the test data identifying the states (b)

We see for this repeated set of data, the viterbi algorithm is able to distinguish the attack and the normal states. At the end of this experiment we were able to utilize the HMM to clearly distinguish the normal activities from the intrusion activities. This study has demonstrated the feasibility of using the HMM in detecting anomalous behavior for packet network traffic data. These results prove that the HMM model can be used effectively to model the detection engine of an intrusion detection system based on network packet data.

CONCLUSION

In this paper we have shown how the properties of the Markov model can be applied to network intrusion detection systems. Normal program behaviors were modeled using HMM and any anomaly from the model was considered a possible attack. This was a simple two state model but a real network has several possible states. Based on the positive results from this work we plan to increase the number of attacks dimensions discoverable by the HMM. Despite the many positives of this modeling system, we observed some issues. The first is the computational issue in the implementation of the forward backward algorithm in the solving of problem 1. [8] Recommends a scaling technique to mitigate this problem but that is beyond the scope of this paper. The next issue concerns the training data (observables) used for estimating the HMM parameters. It is possible that a probability of a parameter will be set to zero if there is no occurrence of it in the training data used to model the HMM. This is usually because the training set is too small, and does not cover the full range of normal activity on the network. The solution to this problem is to provide a robust data set for training while striking a balance between too little and too much. [10] Proposes a new intrusion detection system based on Fuzzy HMM. It claims that using this method, as compares to classical HMM, will reduce the training times and the speed of detection will effectively be boosted while saving computer resources. This method will be considered for further research.

ACKNOWLEDGEMENTS

Research Advisor: Dr Richard Dean
The author appreciates the support of the DoD/TRMC SRC and CRC for their support of this work

REFERENCES

Bibliography

[1] Shon Harris, Allen Harper, Eagle Chris, and Johnathan Ness, *Gray Hat Hacking The ethical Hackers Handbook*, 2nd ed., Baucom Michael, Ed. New York, United States of America: The McGraw-Hill Companies, 2008.

[2] Wayne Ross, "iNET Architectral Security Design," Morgan State University, Baltimore, 2008.

[3] James Matthews. (2004, June) generation5. [Online]. http://www.generation5.org/content/2004/fyp2004.asp

[4] R. Khanna and Huaping Liu, "Control Theoretic Approach To Intrusion Detection Using a Distributed Hidden Markov Model," *Wireless Communication, IEEE*, vol. 15, no. 4, pp. 24-33, August 2008.

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

[6] Sandarva Khanal, "Aeronautical Channel Modelling for Packet Networks," Morgan State University, Baltimore, Ungerdraguate Project 2011.

[7] Wenke Lee and Stolfo Salvatore J., "A Framework for Constructing Features and Models for Intrusion Detection Systems," *ACM Transaction on Information and System Security*, pp. pp.227-261, November 2000.

[8] L Rabiner and B Juang, "An introduction to Hidden Markov Models," *ASSP Magazine, IEEE*, pp. PP.4-16, January 1986.

[9] Wei Wang, Xiao-Hong Guan, and Xiang-Liang Zhang, "Modelling Program Behaviours by Hidden Markov Models for Intrusion Detection," in *Third Internationsl Conference on Machine Learning and Cybernetics*, Shangai, 2004, pp. 2830-2834.

[10] Yongzhong Li, Yang Ge, Xu Jing, and Zhao Bo, "A New Intrusion Detection Method Based on Fuzzy HMM," in *Industrial Electrinics and Applications ICIEA 2008*, 2008, pp. 36-39.

[11] Wenke Lee and SlavatoreJ. Stolfo, "A framework for Constructing Features and Models for Intrusion Detection Systems," *ACM Transactions on Information and System security*, pp. PP.227-261, November 2000.