Spring 2009

A Hidden Markov Model Based Approach to Detect Rogue Access Points

Gayathri Shivaraj
Old Dominion University

Recommended Citation
Shivaraj, Gayathri. "A Hidden Markov Model Based Approach to Detect Rogue Access Points" (2009). Master of Science (MS), Thesis, Electrical & Computer Engineering, Old Dominion University, DOI: 10.25777/m6yf-ws29
https://digitalcommons.odu.edu/ece_etds/526

A HIDDEN MARKOV MODEL BASED APPROACH TO DETECT ROGUE ACCESS POINTS

by

Gayathri Shivaraj
B.E. June 2006, MJCET, Osmania University, India

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
MAY 2009

Approved by:

Sachin Shetty (Co-Director)

Min Song (Co-Director)

Dimitrie Popescu(Member)

ABSTRACT

A HIDDEN MARKOV MODEL BASED APPROACH TO DETECT ROGUE ACCESS POINTS

Gayathri Shivaraj
Old Dominion University, May 2009
Co-Directors: Dr. Sachin Shetty and Dr. Min Song

One of the most challenging security concerns for network administrators is the presence of Rogue access points. The challenge is to detect and disable a Rogue access point before it can cause hazardous damage to the network. This thesis proposes a statistically based approach to detect Rogue access points using a Hidden Markov Model, which is applied to passively measure packet-header data collected at a gateway router or any monitoring point. This approach utilizes variations in packet inter-arrival time to differentiate between authorized access points and Rouge access points. This approach used the inter-arrival time of a packet as a distinguishing parameter because it varies drastically for a normal activity and an intrusive activity. The main contribution of this thesis is the design and development of a Hidden Markov Model by analyzing Denial of Service attacks of 802.11 based Wireless Local Area Networks which affect the traffic characteristics like packet size, inter-arrival time, delays etc. Experimental validations demonstrate the effectiveness of the approach. This trained Hidden Markov Model can detect the presence of a Rogue access point promptly within one second with extreme accuracy (very low false positive and false negative ratios are obtained). The success of this approach lies in the fact that it leverages knowledge about the behavior of the traffic characteristics of 802.11 based Wireless Local Area Networks and the properties of Denial of Service attacks. Experiments were also performed to improve the accuracy of

our HMM model. This approach is scalable and non-intrusive, requiring little deployment cost and effort, and is easy to manage and maintain. This research was also accepted and published in MILCOM 2008, a technical Conference held in San Diego.

ACKNOWLEGEMENTS

This thesis could not have been completed without Dr. Sachin Shetty, who not only served as my Co-advisor but also encouraged and challenged me throughout my academic program. I attribute the level of my Masters degree to his encouragement and effort. This work was also supported by Dr. Min Song. I would like to thank Dr. Min Song for being my Co-advisor and guiding me throughout my research. I am most indebted to him for helping me in my research in the absence of my advisor. I would also like to thank Dr. Dimitrie C. Popescu for taking time out of his busy schedule to work with me in this endeavor.

I would like to specially thank my parents for their love, support and for always being there for me and pushing me to strive for my best. Finally, I would like to thank all my friends and well wishers who are responsible for the successful realization of my thesis as well as express my apology that I could not mention each personally.

Dedicated to the God Almighty, Mom, Dad, Sister and Dr. Albin.

For all of your love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

*Deployment of wireless local area networks (WLANs) in commercial and military domains has been growing at a remarkable rate during the past several years. The presence of a wireless infrastructure within an organization's premises, however, raises various network management and security issues. One of the most damaging and severe flaws of WLAN technology is weak security. Intrusion into wireless networks is relatively easier when compared to wired networks. Even though attempts have been made to secure these networks, the technology used is intrinsically insecure and still highly susceptible to active attacks and passive intrusions. Therefore, network security has become one of the essential requirements of any large network. Securing network infrastructure is like securing possible entry points of attacks on a country by deploying appropriate defense.

Threats to WLANs are numerous and potentially devastating. Security issues ranging from misconfigured wireless access points (WAPs) to session hijacking to Denial of Service (DoS) can plague a WLAN very easily. A wireless access point (WAP) is a device that allows wireless communication devices to connect to a wireless network using Wi-Fi, Bluetooth or related standards. The WAP usually connects to a wired network and can relay data between the wireless devices and wired devices on the network. A Denial of Service (DoS) attack is an attempt to make

* Using IEEE Transactions Style is used for a model in this thesis.

a computer system or server unavailable to its intended users. One unique feature of this kind of attack is it saturates the target machine with external communication requests, such that it cannot respond to legitimate traffic or responds so slowly as to be rendered effectively unavailable. DoS attacks are implemented by either forcing the targeted computers to reset or consuming its resources, so it can no longer provide its intended service or obstructing the communication channel between the intended users and the victim, so they can no longer communicate adequately. Wireless networks are not only susceptible to TCP/IP-based attacks native to wired networks, they are also subject to a wide array of 802.11-specific threats. To aid in the defense and detection of these potential threats, WLANs must employ a security solution that includes an intrusion detection system (IDS) to detect several types of malicious behaviors that can compromise the security and trust of a computer system. It is designed to detect unwanted attempts at accessing, manipulating, and disabling of computer systems, mainly through a network, such as the internet. The approach used is a network based IDS. The features associated with the network IDS are low cost, easy deployment, detect network based attacks, real time detection and quick response. This chapter mainly deals with the introduction and background of WLANs and the need for intrusion detection in WLANs.

1.1 Need for Intrusion Detection Systems

WLANs are prone to a variety of threats. The standard 802.11 encryption method, Wired Equivalent Privacy (WEP) is a weak solution for the wireless threats these days [3]. The WEP key of a wireless transmission can be acquired via brute force attack. Therefore, even if WEP encryption is utilized on a WLAN, an attacker can potentially intercept and decrypt sensitive data from wireless communications. An intrusion into wireless networks can be carried in two ways, either by passively sniffing the wireless network or by physically installing access points on the wired networks in unnoticed places. They gather sensitive data by introducing a Rogue Access Point (RAP) into the WLAN coverage area. A RAP is a wireless access point which is setup by an attacker for the purpose of sniffing wireless network traffic. The RAP can be configured to look like a legitimate wireless access point, and since many wireless clients simply connect to the access point with the best signal strength, hosts can be "tricked" into inadvertently associating with the RAP. Once a host is associated, the intruder through the RAP can monitor all communications occurring in the network. In addition to the intruders, hosts can also introduce RAPs. Low cost and easy implementation coupled with the flexibility of wireless network communications makes WLANs highly desirable to hosts. By installing a WAP on an established LAN, a host can create an entry into the network, subverting all the hard-wired security solutions and leaving the network open to intruders. It is for this reason that even organizations without a WLAN implementation must strongly consider deploying a wireless IDS solution. It is very possible hosts can and will install a RAP, exposing even an exclusively hard-wired organization to the risks of WLANs.

WLANs are also subject to a number of DoS attacks that can render a network inoperable. It can slow down the network operations to dragging speeds or actually force it to stop. It has always been a show-stopper in most of the critical applications. Intruders can cause malicious DoS attacks by flooding WAPs with association requests and forcing them to reboot. In addition, they can use the aforementioned RAPs to send repeated disassociate/deauthenticate requests to deny service to a wireless client. Sometimes a DoS occurrence on a wireless network may not be intentional. Wireless communications are inherently vulnerable to signal degradation when encountering physical objects like trees, buildings, rain etc. In addition to physical obstacles, many common devices such as microwave ovens, cordless phones and other wireless devices can interfere with 802.11 networks which cause a significant reduction in WLANs performance. A variety of other WLAN threats exist and additional vulnerabilities are being identified at an ever-increasing pace. The threats are real; they can cause extensive damage, and they are becoming more prevalent as the 802.11 technology grows in popularity. Without any kind of detection mechanism, it can be difficult to identify the threats to a WLAN. A lack of threat awareness can lead to a network not adequately secured against the threats facing it. Only when the threats to the network are realized can the WLANs be properly equipped with the necessary security measures.

1.2 Intrusion Detection in Wireless Local Area Networks

Intrusion detection is the art and science of finding compromises or attempts to compromise a network. The term has been broadened to include the detection of other forms of attacks, such as scanning, enumeration, DoS and so on... In some cases the IDS may also respond to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network. A wireless IDS monitors the traffic on an entire network to determine if an attack or intrusion has occurred. Although intrusion detection technology has improved significantly over the past decade, it's still relatively immature.

Standard tools for monitoring wired networks and ensuring their security examine only network (layer 3) or higher abstraction layers based on the assumption that the lower layers are protected by the physical security of the wires [2]. However, this assumption cannot be extrapolated to wireless networks because of the broadcast nature of such networks and their radio technology that is being used. Ideally, IDS for wireless networks should function at the data link layer (layer 2) or even lower if extremely high security is required. Yet, to go beyond mere detection and to actually provide useful protection for the network, it might be necessary to actively disable unauthorized clients attempting to access the network.

An intrusion into wireless networks can be carried in two ways, either by passively sniffing the wireless network or by physically installing access points on the wired network in unnoticed places. Physically installing an access point is very dangerous compared to the passive sniffing because during the sniffing process the intruder does not have access in to the network whereas by installing an access point in the network, the intruder has got the access into the network, and he is all set to compromise the network. On many networks, intrusions are not limited to unauthorized clients but could include unauthorized access points. Often, these "RAPs" might be installed by valid users attempting to increase the range of the network but doing so without proper authorization [2]. This usually results in a security hole that may be exploited by intruders. One of the most challenging issues in WLANs is RAPs i.e., wireless access points that are installed without explicit authorization from a local network administrator [1]. Although usually installed by employees of the organization for convenience or higher productivity, RAPs pose serious security threats to the local network. Within a wireless network, RAPs are more damaging than rogue users. First, they potentially open up the network to unauthorized parties who may steal confidential information or even launch DoS attacks in the network. Nowadays, we get Wireless network cards (which need a program called "iwconfig") that have a feature of capturing all the 802.11 transmissions and hence people with limited security backgrounds can accomplish the process of driving and looking for vulnerable access points. This thesis attempts to address one of the DoS types of attacks caused due to the RAPs in WLANs.

1.3 Rogue Access Points (RAPs)

Rogue access points (RAPs) are the wireless access points that are installed on a secure company's network without explicit authorization from a local network management or have been created to allow a cracker to conduct a man-in-the-middle attack. Wireless networks are particularly more vulnerable to RAPs because of factors such as open medium, insufficient software implementations, potentials for hardware deficits, and improper configurations. Liran et al. [3] has given a broad classification of RAPs according to their research work. RAPs are classified into the following categories:

- Improperly configured access points: Without any malicious intent, a legitimate access point can suddenly turn into a rogue device because of a minor configuration mistake. There are several scenarios where an access point can be improperly configured. A network administrator with in sufficient security knowledge (e.g., an inability to choose appropriate authentication and encryption settings) could fail to set up the access point properly. It is also possible that an access point's driver is faulty or that device itself is physically defective. A properly specified and implemented security policy can avoid this vulnerability.
- Unauthorized access points: Installing an access point on a secure network without authorization from the network administrator also creates a RAP. Even though the network administrator does not manage the access point, it can still become accepted as part of the official network. The reason for it to be accepted as an authorized access point is the access point transmits and receives network traffic as does a legitimate access point. Driven by the convenience of network access, this class of RAPs routinely exists in large organizations with many employees. Anyone with physical access to the premises can ignorantly or maliciously connect a cheap wireless access point to network.
- Phishing access points: An intruder can set up an access point outside the

wireless network of a facility. It attempts to fraudulently acquire critical credentials, such as usernames and passwords, by masquerading as a trustworthy access point. In addition, it can be configured to replay beacons that it overhears from legitimate access points, thus fooling some clients within the facility to connect to it. This can allow an attacker to conduct a man-in-middle-attack on a Wi-Fi network that does not enforce mutual authentication (client-to-server authentication and server-to-client authentication).

- Compromised access points: Though the access point is properly configured with security features like WEP or WPA-PSK enabled, an attacker can still crack the key being used. Once an adversary discovers the secret key, all of the access points using the same credentials in a Wi-Fi network become RAPs. This reflects a significant payoff for the adversary because a greater number of physical location attacks become vulnerable. Further, a compromised access point allows an attacker to easily masquerade herself as a legitimate user and gain access to potentially sensitive data.

A more relevant scenario would have been an intruder planting an access point with a higher than normal broadcast power to masquerade as a legitimate access point. Unknowing clients would attempt to associate with this access point believing it is valid. The intruder could then use information collected from these association attempts to determine network security settings and other aspects of the network.

In spite of all these, a properly configured access point with security features enforced can still be compromised, thus becoming a RAP [1]. As mentioned above, WEP (Wired Equivalent Protocol) the most common security protocol was also proved to be broken even when correctly configured. Later, WPA (Wi-Fi Protected access) was created in response to the serious weaknesses that researchers found in WEP which was also not very successful. Due to these pitfalls in WEP and WPA, an access point can be easily compromised. Therefore, the traditional way of protecting networks with encryption and firewalls is no longer sufficient.

According to an early study by Gartner [2], RAPs are present on about 20% of all enterprise networks. The main reason is advancements in hardware and software which have made access point installation, access point discovery, and access point compromise an easy task for attackers. It is convenient to obtain an access point and plug into a network without being discovered for some time. Moreover, commodity Wi-Fi network cards have the capability to capture all 802.11 transmissions. This has led to the increase in the process of driving around and looking for vulnerable access points (war-driving activities).

1.4 Contributions

The main contribution of this thesis is a novel approach for RAP detection based on measurements collected at the edge of a network. The method used is a statistical approach to detect RAPs using Hidden Markov Models (HMM). A HMM is a stochastic approach in which the system being modeled is assumed to be a

Markov process with unknown parameters; the challenge is to determine these unknown parameters using the observable data. We considered intrusion as a pattern of observed sequence; the detection is done by identifying and eliminating anomalies, by measuring deviations from normal processes using the HMM. HMM based-approaches correlate the system observations and state transitions to predict the most probable state sequence. The HMM is applied to passively measure packet-header data collected at a gateway router. The approach detects a RAP by observing the traffic characteristics of the associated individual end hosts. It is probabilistic and uses Markov chains to represent the likelihood of transitions between the different security states of an access point. This approach roughly works as follows. First, the HMM model is trained based on a training data set which has information gleaned from packet traces. The packet traces are collected from a test-bed wherein traffic comprises of normal Internet activities and DoS [2] attacks. Once the HMM model is trained, we monitor the packet arrivals of different flows at the edge of the network. By observing the packet inter-arrival time of these flows, the HMM model detects an access point as a RAP or an authorized access point.

The first contribution in the thesis is the estimation of the number of states of the HMM model and what those states refer to. The states of the HMM model play a very important role in the training and the detection process. The second one is the modeling or the training of the HMM model; the model estimation was done using the HMM toolbox implemented by Kevin Murphy in [14]. The third contribution is the implementation of the HMM for the detection of RAP in a WLAN.

The key strength of this approach lies in the fact that it influences the knowledge about the behavior of the traffic characteristics of 802.11 based WLANs and properties of DoS attacks. This approach is scalable and non-intrusive, requiring little deployment cost and effort, and is easy to manage and maintain.

1.5 Outline of the thesis

This thesis is organized into six chapters, including the present chapter. Chapter II gives an elaborate description of background and related work based on the extensive survey done during the course of this thesis. It discusses several successful methods previously implemented by researchers for identifying RAPs in wireless environments and the like. Chapter III discusses the problem statement and the proposed approach to solve the problem. Chapter IV provides a detailed description of the methodology and analytically explained the approach. Chapter V gives a detailed explanation of the experiments performed and the results obtained. This section also talks about the pitfalls of the approach. Chapter VI provides a conclusion to this thesis by summarizing the techniques used in the development of the methodology proposed, some critical results obtained, and suggestions for future research in this area. This work was presented at the MILCOM 2008 technical conference held in San Diego.

CHAPTER 2

BACKGROUND AND RELATED WORK

Not much work has been done in the field of RAPs unlike the research done in Intrusion detection. Most of the current approaches for detecting RAPs are rudimentary and easily evaded by hackers. Some organizations have equipped IT personnel with wireless packet analyzer tools (e.g., sniffers) on laptops and handheld devices, forcing IT personnel to walk the halls of the enterprise or campus searching for RAPs. This method is generally ineffective because manual scans are time-consuming and expensive and therefore are conducted infrequently for detecting RAPs. The frequent attacks on network infrastructure, using various forms of DoS attacks and worms have led to an increased need for developing techniques for analyzing and monitoring network traffic and perform anomaly detection.

2.1 Different Approaches for the Detection of RAPs

A comprehensive taxonomy of RAP detailing different categories of RAPs has been presented by Ma et al. [3] as mentioned in the previous chapter. The authors have categorized access points in the following four classes: improperly configured, unauthorized, phishing, and compromised as mentioned above. The brute-force approach of RAP detection used by most enterprises is to equip IT personnel with wireless packet analyzer tools and scan the network traffic [4-5]. AirDefense [4] is one such product. It uses a combination of radio frequency sensors and an intrusion detection server to capture process, and correlate network events. However, the latest release, Air Defense 7.2, has a starting price of US $7,995. Also, the radio frequency (RF) sensors make it difficult to guarantee a complete coverage of the network to ensure effective RAP detection.

To the best of our knowledge, there are few research efforts for detecting RAP. Fault diagnostics in IEEE 802.11 networks is presented in [6]. Multiple access points and mobile clients perform RF monitoring to help detect the presence of RAPs. Each client is equipped with special diagnostic software, and RAPs are assumed to transmit beacon messages and respond to probe requests. Further, its detection ability is not based on the assumption that RAPs will function properly.

Differences in inter-packet spacing between traffic flows on wired and wireless networks is used in [8-9] for identification of RAPs. However, the scheme does not differentiate between wireless traffic from authorized and unauthorized access points. It also assumes that access points will be connected within one hop to a switch monitoring the traffic, and relies on visual inspection of traffic characteristics. Kim et al. [2] has used multiple network sniffers for detecting RAPs and eavesdroppers. Each sniffer has three network cards, and the intrusion detection capabilities are stymied by MAC address spoofing. Yeo et al. [21] improves the performance of wireless monitoring by merging packet captures from multiple network sniffers and carefully selecting sniffer placement. The techniques are exploited to characterize MAC layer traffic and perform retrospective diagnoses.

Recently, Wei et al. [10] proposed two passive online RAP detection algorithms. The core of these two algorithms is the sequential hypothesis tests applied to packet-header data that are passively collected at a monitoring point. Both algorithms exploit the fundamental properties of the 802.11 CSMA/CA mechanisms and the half duplex nature of wireless channels to differentiate wired and wireless TCP traffic. Once TCP ACK-pairs are observed, prompt decisions are made with little computation and storage overhead. Yin et al. [11] proposes a layer-3 RAP detection approach using the combination of a verifier and wireless sniffers. In this approach, a verifier on the internal wired network is employed to send test traffic towards wireless edge. Once wireless sniffers capture an access point relaying the test packets, the access point is flagged as rogue. In addition, binary hypothesis testing technique is adopted to improve the robustness of detection.

2.2 Different Types of Intrusion Detection Systems

Bahl et al. [7] propose a distributed monitoring infrastructure called DAIR. It attaches USB wireless adapters to desktop computers for more comprehensive traffic capturing ability. The effectiveness of DAIR is dependent on access point functionality that can be easily turned off. Additionally, both of [6] and [7] assume that characteristics of IEEE 802.11 standards cannot be violated by the adversaries.

David et al. [16] introduced a router throttle mechanism which was used for countering Distributed Denial of Service (DDoS) attacks directed at an Internet server. A DDoS attack is one in which a multitude of compromised systems attack a single target, thereby causing DoS for users of the targeted system. This mechanism specifically targeted Neptune type of DDoS attacks. The authors have advanced a control-theoretic, server-centric model useful for understanding system behavior under a variety of parameters and operating conditions. The adaptive throttle algorithm is effectively used to protect a server from resource overload, and increase the ability of normal traffic to arrive at the intended server. The results indicate that server-centric router throttling is a promising approach to prevent DDoS attacks, but several nontrivial challenges like low computation and memory overheads remain that prevent its immediate deployment in the Internet.

Krishna et al. [18] developed a tool for statically validating a TCP server's ability to survive SYN flooding attacks proposed in their paper. The tool automatically transforms a TCP-server implementation into a timed automation, and it transforms an attacker model, given by the output of a packet generator, into another timed automation. Together the two timed automata for a system for which the model checker UPPAAL can decide whether a machine is in a bad state, which is based on the buffer overruns that are reached in the system.

Mohan et al. [25] developed a distributed agent based intrusion detection system for WLANs that can detect unauthorized wireless elements like access points, wireless clients that are in promiscuous mode etc. It is one of the most common approaches used in this field. Due to the property of wireless technology to face attenuation of signals with distance, they consider multiple access points at different locations to increase the area of coverage. This has a central administrator which is the main central server that monitors all the wireless cells in the network. It maintains a record of all the access points and agents present on the network and also a list of all the clients which need to access the wireless network. Any new access point or wireless client which needs to be installed on the network, should be registered with the central administrator, failing to do this they would be considered as an unauthorized element. When a wireless element like an access point or a wireless client card is registered, the element's information is sent to all agents and the agents, upon receiving this information act accordingly.

The methods used for detecting intrusions in WLANs are based on physical layer features extracted from the RF waveforms of individual network packets. The features considered include those intrinsic to the packet source (wireless user node) as well as those related to the propagation path between the source and a network access point. This intrusion detection methodology, which is applicable to any WLAN, can determine whether the source of a wireless packet is a legitimate node or a rogue transmitter. The proposed wireless intrusion detection (WIND) system exploits the unique transmitter and propagation channel characteristics that are inherently encoded in the electromagnetic wave of each packet sent by a wireless user node. WIND measures a set of RF features for each packet transmitted within or into the physical bounds of the network and uses the statistics of the feature set to derive a fingerprint that uniquely identifies the packet's source. The uses of physical-layer features to identify wireless nodes make it much more difficult for an adversary to mimic a legitimate node. The WIND system could be used either independently to produce alerts or block suspicious in-coming traffic or as an additional input to a conventional IDS system.

Schmoyer et al. [19] gave an overview of the wireless IDS and their functionalities. The authors preface with the fact that the most basic level for intrusion detection is to track the Media Access Control (MAC) address of network adapters attempting to associate with the network. If the MAC address does not occur in the white list or is blacklisted, it is flagged as a possible intruder. Such a procedure is commonly known as MAC filtering and might not be practical in a large organization where users may employ their own wireless cards. By checking each MAC address against such patterns, it would be possible to determine forged addresses randomly generated by intruders. It is possible for users or attackers to change MAC addresses reducing the effectiveness of using patterns. To improve detection accuracy, it should be possible to utilize any number of algorithms to profile the attack, including rule-based algorithms, expert systems, or even artificial neural networks that "learn" the normal behavior of the network. To minimize the requirements of each device and to improve detection accuracy by polling more devices, additional intrusion detection logic could take place on the central server where more information and more processing capabilities are available.

One of the current approaches has been to detect RAPs from a central location (a switch that supports a subnet) with the detection independent of the wireless technology. It is a scalable solution, thus not attempting to reassemble data before analysis, and this solution functions independently of the signal range of the RAPs.

Tomko et al. [22] discussed the processing and decision-making performed at the switch with the input as the link layer traffic traversing. The number of hops between the switch and the end point will most likely affect the temporal characteristics of traffic as observed at the switch. Queuing and congestion tend to mask the temporal shaping of traffic through end points. The reliability of wired links makes the temporal characteristics of traffic in a path, to be shaped mostly due to higher layer (e.g., TCP) mechanics, with a relatively simple shaping from the link layer. A wireless link, however, shapes traffic differently. Due to variations in channel conditions, wireless link capacity varies and random delays are introduced. The difference in link speed between wired and wireless links also shapes the characteristics significantly. Accordingly, this detection scheme is based on the premise that if traffic at a switch port is observed in both directions over time, and input-response correlated, different patterns may be observed for segments with and without wireless links. The input-response correlation involves estimating which part of traffic is in response to which impulse (or input). Thus, for every response quanta of traffic from an end point in the segment, temporal characteristics may be analyzed by classifying mean, variance, and other frequency response characteristics of inter-packet spacing. As time progresses, RAPs (or a number of them) are detected, when the difference in state variables between ports crosses a threshold. The main aim was to experiment and derive state representations and its derivation from the observed temporal characteristics of traffic, and they have observed differences in inter-packet spacing in wired and RAP scenarios.

There are several other research works on RAP detection and one of the approaches is to detect RAP detection in a heterogeneous network. These research studies have classified the traffic originating from a wireless LAN and an Ethernet and for packets originating from wireless link; they check whether the host is authorized to use the wireless network.

Shetty et al. [9] showed the classification between the traffic originating from the Ethernet and the WLAN is done by considering the number of hops between the end host and the gateway router assuming that the wired and wireless end hosts are connected to the gateway router. This approach also states that the wireless links use a contention based MAC protocol to access the shared link whereas the Ethernet links use a non-contention based access to a switched wired link. Ethernet links have a greater data rate as compared to wireless links. The difference helps to detect WLAN hosts. The authors have also demonstrated the detection of RAP by distinguishing traffic generated by authorized WLAN hosts from unauthorized WLAN hosts, and considering the unauthorized users are interested in gaining access to any vulnerable host, the request packets are sent to random end host machines, thereby increasing the crossing-access. If the frequency of the crossing-access exceeds a threshold, the NTA (Network Traffic Authority) detects the unauthorized WLAN host as connected to a RAP.

An attack fingerprinting system to identify instances of repeated attack scenarios on the network was proposed in [23]. This tool is a combination of attacking hosts and attack tool. Since packet contents can be easily manipulated, they have based their fingerprints on the spectral characteristics of the attack stream which are hard to forge. The application of pattern matching techniques made use of the maximum-likelihood classifier to identify repeated attack scenarios. Their study indicates the spectral fingerprint is primarily defined by the attacking tool; however, the network influences the fingerprint when it is saturated.

Bahl et al. [24] analyzes two alternative approaches for anomaly detection over system call sequences and arguments. A Deterministic IDS which built a Finite State Automation (FSA) model complemented by a network of dataflow relationships among the system call arguments. The main contribution is the use of Self Organizing maps (SOM) to model path similarity. They adapted the Symbol SOM algorithm to make it suitable for computing numeric distance between two paths. They also proposed a new model for commuting the frequency of traversal of edges on the FSA prototype to make it able to detect DoS attacks.

2.3 Hidden Markov Models used in Intrusion Detection Systems

Hidden Markov Models (HMMs) have also been used to detect and classify Network Intrusions [17]. The HMMs were modeled to detect buffer overflow based attacks. The disadvantage of this method is it cannot be applied for detection of attacks that are performed over a long period.

Wei et al. [20] have discussed the use of continuous-time HMMs for network protocol and application performance evaluation. They developed an algorithm to infer the continuous-time HMM (CT-HMM) from a series of end-to-end delay and loss observations of probe packets. They infer a CT-HMM from delay and loss observations seen by a sequence of probes sent from one end host to another end host. This CT-HMM can then be incorporated into a simulator or an emulator to drive the simulation of a network protocol or application by providing losses and delays to packets in the network at arbitrary points of time. They have demonstrated that the CT-HMM is a good model of the network settings by showing that the behavior of a flow driven by the model is similar to that of a flow in the original network. Here, the flow can be governed by a network protocol or an application. They validate using both TCP and a streaming video application.

The main contribution of the Arnes et al. [27] in their paper is an approach to network risk assessment. They determined the risk level of a network as the composition of the risks of individual hosts, providing a more precise, fine-grained model, and they used HMMs to represent likelihood of transitions between security states. Finally, they integrated their risk assessment tool with an existing framework for distributed, large-scale intrusion detection and applied the results of the risk assessment to prioritize the alerts produced by the intrusion detection sensors. The implementation of this model processes the alerts produced by a set of sensors monitoring a number of hosts. The main idea was to show the level of risk activity; therefore, the HMMs used allow the risk to gradually decrease even if the host in question has been assessed to be in a compromised state. The primary advantage is that HMMs provide an established framework for state estimation, modeling both the probabilities of entering certain states as well as the probabilities of receiving different observations in each state, as well as the probabilities of receiving different observations in each state effectively providing a framework for representing the false-positive and false-negative effects of IDS. In addition, the risk can be only be interpreted by using knowledge of the normal risk level of the system as well as the maximum risk of the system which is obtained from the learning or training procedure of HMM. A limitation of this definition of network risk is that it does not consider dependencies between hosts. Another important limitation of this approach is the need for model parameter estimation.

Salamatian et al. [28] proposed a HMM model for the transmission channel where the different states of the HMM indicates the different states the channel goes through. Each state in this model corresponds to the probability that a packet sent by the transmitter will be lost. A Markov chain governs the transition between the different states of the channel; this Markov chain is not observed directly, but the received packet flow provides some probabilistic information about the current state of the channel as well as some information about the parameters of the model. They have explained some useful algorithms for the estimation of the channel parameters and for making inference about the state of the channel. In this method, they have analyzed the end-to-end loss process and used it to make inferences about the state of the network as seen by the application. They have developed a two-step network state estimation procedure: first, a model calibration step that chooses a number of states and calibrates a HMM for the loss, and a second step will use this HMM to estimate the actual state of the network by observing the sequence of packets. One of the problems omitted by the authors is the tracking of non-stationarity and variation in the subjacent network channel by the HMM. The HMM obtained from this paper can also be used in adaptive applications. The integration of such an estimation mechanism in a video diffusion over the internet is being studied.

Our proposed framework differs from previous work in which it provides an efficient and prompt detection of RAPs by analyzing the traffic characteristics of WLANs. It also defends against a more insidious type of RAPs, i.e., the compromised access points, that have never been addressed in the literature before. According to Queuing theory, "average service time must be less than the inter-arrival rate or the system is unstable". Our model can detect RAPs, which are the source of specific DoS attacks. Moreover, the deployment of this model does not require modifications to the underlying wireless standard. This makes our framework an efficient and cost-effective solution.

CHAPTER 3

HMM BASED RAP DETECTION APPROACH

This chapter describes the RAP detection problem at a high level and the approach towards solving this problem. The approach proposed for the RAP detection in WLAN is a Hidden Markov Model (HMM). This detection process is a typical form of anomaly intrusion detection because the intrusion detection system collects and processes a large volume of the network traffic where the processing techniques include machine learning and statistics. An anomaly based intrusion detection system is a system for detecting computer intrusions and misuse by monitoring system activity and classifying it as either normal or anomalous. This is a kind of system where in order to determine what attack traffic is, the system must be trained to recognize normal system activity and then by observing abnormalities in the data, the intrusions or attacked activities can be detected. It has been recently proved that HMM is a good tool to model normal behaviors of authorized hosts or processes for anomaly intrusion detection. The main idea of using HMM for RAP detection is for reducing demands on training time and memory resources.

A HMM defines two concurrent stochastic processes: the sequence of HMM states and a set of state output processes. The first process specifies the probability of transition between any state and each of the other states in the system. The second stochastic process is observable symbols at each state of the system. In our network model, we have a very similar scenario where the state of the access point is not observable, but the packet traces are the observable parameters which help us determine the state of the access point. The packets coming from the end hosts through the access points provide information about the access point and help us determine the state of the access point. As a machine learning method for constructing a finite state machine, HMMs have been widely used in knowledge discovery, pattern classification, speech recognition, DNA sequence modeling, and so on. This is due to the fact that HMMs efficiently model the sequential characteristics between the events of normal behaviors. A remarkable feature of HMM is that it suggests a paradigm shift in the applied detection techniques from the structural pattern recognition techniques to sequential learning techniques.

3.1 Problem Statement

Consider a wireless local area network, e.g., a university campus or a military network, as illustrated in Figure. 1. End hosts within this network only use 802.11 WLAN to access the network. A monitoring point is located at the gateway router of this wireless local network, capturing traffic flows coming in and going out of the network. The end hosts are connected to three access points (AP1, AP2, and AP3). Each of these access points can be termed as authorized or rogue depending on the traffic generated by them. The end hosts connected to authorized access points generate traffic indicative of normal Internet activities (web browsing, email, ftp transfer, etc.). The end hosts connected to the RAPs are the source of DoS attacks. This intrusion detection method using HMM is an offline detection process. The network data is collected at the monitoring point, i.e. the gateway router of the network, and this model is used to carry out the detection process. Our goal is to determine (1) what fraction of traffic flows are the source of DoS attacks (2) for each traffic flow, what is the probability that this particular traffic flow originated from a RAP and also (3) determine when an access point is in good, probed or compromised state.

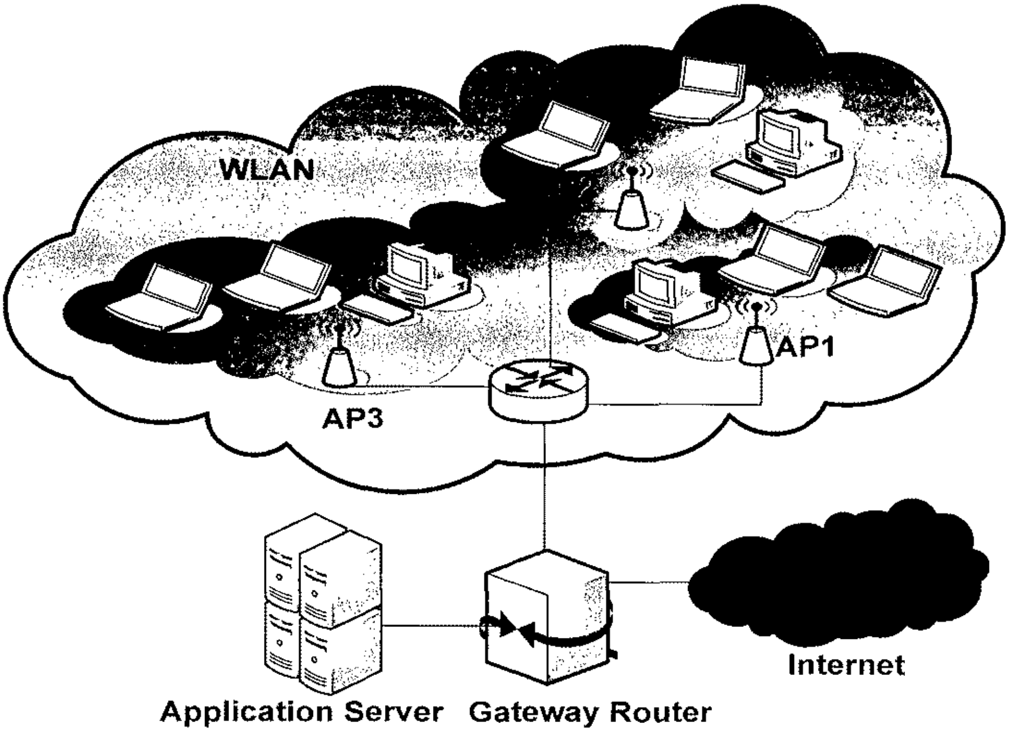

Figure 1. Network Configuration.

Our approach utilizes the intrinsic characteristics of WLAN connections and DoS attacks. The approach operates roughly as follows. For packets belonging to each traffic flow, the inter-arrival times are observed at the monitoring point. As will be shown in the following chapters, the inter-arrival times for packets originating from authorized access points and RAPs differ significantly. Our trained HMM exploits this difference to differentiate traffic originating from authorized access points and RAP.

In following chapter, we present the analytical basis of our scheme, which demonstrates how the inter-arrival times will differ for traffic flows originating from authorized access points or a RAP. We then describe the design of the HMM (the core of our classification scheme).

3.2 Hidden Markov Model Based RAP Detection Approach

The use of Hidden-Markov Models (HMMs) as a method for detecting intrusions in an individual computer system has been proposed in [12-14]. A HMM enables the estimation of a hidden state based on observations that are not necessarily accurate. The reasons for using HMM as our detection approach are as follows:

First, we all know that a WLAN can have a number of access points and a number of hosts connected to these access points, and each of these machines will have a limited number of repetitive requests with unique packet traces. Hence, we can model an access point behaviour with finite number of states which can be described by a HMM. Second, in the WLANs we have a train of packet traces in which a packet from a particular access point depends on the last packet. This dependency of the observation parameters makes it easier for the detection in the HMMs. Third, a transition from one state to another state can be treated roughly as a modified Markov process which is nothing but the HMM. Another feature of the HMM is that it is able to model the probability of false positives and false negatives associated with the observations which is one of the most important parameters in anomaly intrusion detection. The method is based on Rabiner's work on HMMs in [15].

In our problem setting, we use the HMM to describe the current security state of the access points in the network by analyzing the packet traces coming from each of them. The security state of the access point can be identified by observing the inter-packet arrival time in the packet traces. These packet traces help us determine the state of an access point and thereby detect the RAPs to be more precise compromised access point.

The complete parameter set of our HMM model is given as $\lambda = (P, Q, \pi)$. The same notations in [15] are used to describe our HMM model.

- $N = 3$: number of states in the model.
- $M = 3$: number of distinct observation symbols per state.
- $T = 10$ length of the observation sequence, i.e. the number of symbols observed.
- $V = \{v_1, v_2, v_3\}$: The discrete set of possible observation symbols.
- $\pi = \{\pi_i\}, \pi_i = P(i_t = i)$: The probability of being in state i at $t = 1$.
- $P = \{p_{ij}\}, p_{ij} = P(i_{t+1} = j, i_t = i)$: The probability of being in state j at time $t+1$ given that the current state is i at $t = 1$.
- $Q = \{q_j(k)\}, q_j(k) = P(v_k$ at $t / i_t = j)$:The probability of observing symbol v_k given that the current state is j.
- $O = \{O_1, O_2, ..., O_{10}\}$: Observation sequence; denotes observation symbol observed at time t.
- $X = \{X_1, X_2, ..., X_{10}\}$: State sequence; X_t denotes the sequence of states visited by the access point.

Assume that each access point can be modelled by N different security states, i.e. $S = \{s_1, ..., s_N\}$. The security state of an access point changes over time which is an indication of normal or rogue activities. The sequence of states visited by an access point is denoted by $X = x_1, ..., x_T$, where $x_t \in S$. Traffic flowing through each access point is monitored at the gateway router which is the monitoring point in the network. The monitoring process keeps track of the inter-arrival time of the traffic flow. A range of inter-arrival times is represented as an observation message in our HMM model. The ranges of inter-arrival times are represented as observation messages from the observation symbol set $V = \{v_1, ..., v_M\}$, where M is the total number of messages (or unique ranges). The sequence of observed messages is denoted by $Y = y_1, ..., y_T$, where $y_t \in V$ the observation message is received at time *t*. The HMM for each host consists of a state transition probability matrix ***P***, an observation probability matrix ***Q***, and an initial state distribution π. The complete parameter set of our HMM model is denoted by $\lambda = (P, Q, \pi)$. The access points modelled in this thesis are assumed to have three possible security states *S= {G,* ***P****, C}* which are defined as follows:

- **Good (*G*):** The access point is not subject to any attacks. This state represents that the access point is not probed or attacked, and it behaves normally in the network without any intrusive activity.
- **Probed (*P*):** The access point is subject to probing. Port sweeping is a good example of probing. This shows that the access point can be compromised or attacked by unauthorized hosts in order to intrude into the network.
- **Compromised (*C*):** It shows that an unauthorized user which tried to intrude into the network has compromised the access point. This is the state where the access point has been attacked, and the access points begin to malfunction in the network and try to intrude in the network activities.

Figure. 2 shows the HMM model for the security states of the access point. The edge from one node to another represents the fact that when an access point is in the state indicated by the source node, it can transit to the state indicated by the destination node. Note that the graph is fully connected which indicates it is possible to transit from any security state to any other security state.

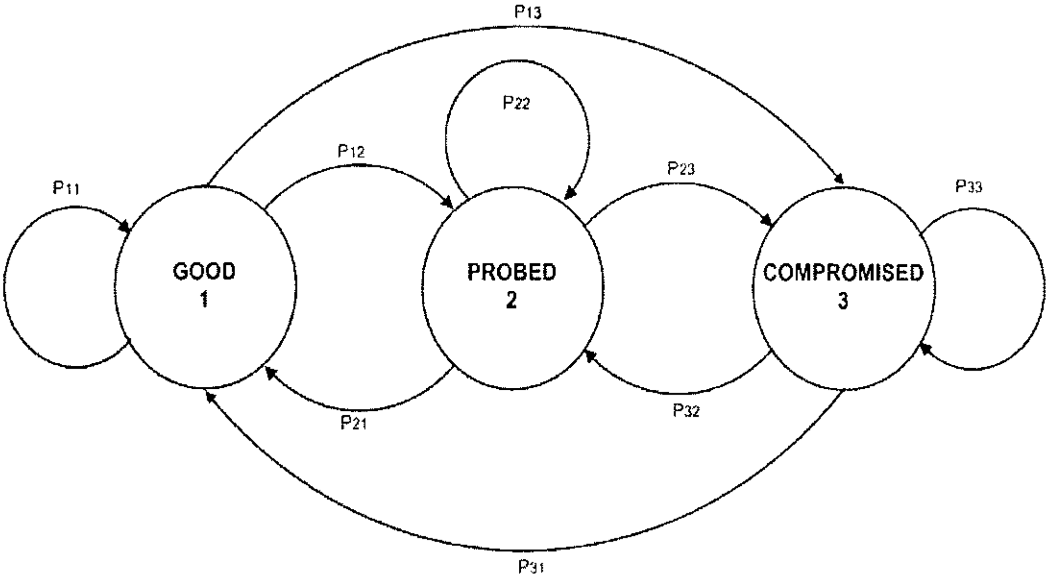

Figure 2. Three state HMM model.

The state transition probability matrix ***P*** describes the probabilities of transitions between the states of the model. The transition probability p_{ij} describes the probability that the model will transfer to state s_j at time $t+1$ given that it is in state s_i at time t, i.e.

$$p_{ij} = P(x_{t+1} = s_j \mid x_t = s_i), 1 \le i, j \le N. \qquad [1]$$

The observation probability matrix $\boldsymbol{Q}$ describes the probabilities of receiving different observations given that the access point is in a certain state. Each observation, $q_n(m)$ represents the probability of receiving the observation symbol v_m at time t, given that the access point is in state s_n at time t, i.e.,

$$q_n(m) = P(y_t = v_m \mid x_t = s_n), 1 \le n \le N, 1 \le m \le M. \qquad [2]$$

Our HMM based intrusion detection consists of two phases namely training and detection phase. In the training phase, the observation sequence obtained from the inter-arrival times of each packet are transformed into HMM observation sequence, i.e. the HMM model is trained with a normal set of network data. Then the HMM is inferred from the observation sequence. In the training phase, the observation sequence is first transformed into HMM short sequences, then the HMM is used to calculate the most probable state sequence in order to determine if it is normal or anomalous. The following chapter explains in detail the HMM training and detection procedure which are the basic and important problems for HMMs.

CHAPTER 4

HMM TRAINING AND DETECTION

This chapter gives the details of the two main stages in our approach. Stage 1 is the training of the HMM based on the packet traces. The most well-known Baum-Welch algorithm is used to train our HMM model, it is considered as batch training because it allows only one observation sequence at a time. Ideally, a well trained HMM can give sufficiently high likelihood only for sequences that correspond to normal behaviours. Sequences corresponding to abnormal behaviours, on the other hand, should give a significantly lower likelihood values. Stage 2 is the detection of RAP by the trained HMM. The main aim of the detection phase is to discover the hidden state sequence that most likely produced a given observation sequence. There are several possible ways to find an optimal state sequence associated with the given observation sequence. One way of doing this is to use the Viterbi algorithm to find the single best state sequence.

There are two crucial aspects of the HMMs which are important in our research. Those aspects are sequence modeling the sequence (Training) and sequence recognition (Detection) using HMMs. The sequence modeling is concerned with the optimization of the various parameters of a HMM to best model a real world phenomena.

4.1 HMM Training – Baum-Welch Algorithm

A very fundamental issue to be solved when using a HMM is the determination of its structure, topology and the number of states, in other words finding a method to determine and adjust the model parameters to maximise the probability of the observation sequence given the model. The goal of HMM training is to estimate appropriate values for the model parameters ***P*** and ***Q***. A uniform initial distribution of the ***P*** and ***Q*** parameters is adequate as a basis for training the parameters, according to [15]. The initial parameters can alternatively be determined by a network administrator based on traffic statistics collected over a period of time. These methodologies provide a framework for identifying threats and vulnerabilities and for determining probabilities and consequences of DoS attacks. Based on a HMM with initial parameters, there are several algorithms available for re-estimating the parameters (i.e., training the models). There is, however, no analytical solution to the re-estimation problem.

A standard approach for learning HMM parameters is the Baum-Welch method or forward-backward method which uses iteration process to select HMM parameters to maximize the probability of an observation sequence. The Baum-Welch method was adapted to estimate the HMM parameters for our model. The first step in the estimation process is to initialize the parameters of the HMM. The second step is to generate the training set. We refer to a set of traffic flows from which the observation distribution is obtained as a training set. For our HMM model, a state sequence of length 10 was considered

$$X = x_0, x_1, \ldots, x_9 \qquad [3]$$

with corresponding observations

$$Y = y_0, y_1, \ldots, y_9 \qquad [4]$$

Then, π_{x_0} is the probability of starting in state x_0. Also, $q_{x_0}(y_0)$ is the probability of initially observing y_0 and p_{x_0, x_1} is the probability of transiting from state x_0 to state x_1. This concept is extended for a state sequence length of 10 in our model; the probability of the state sequence X is given as

$$P(X) = \pi_{x_0} q_{x_0}(y_0) p_{x_0, x_1} q_{x_1}(y_1) p_{x_1, x_2} ... q_{x_9}(y_9) p_{x_8\ x_9} \qquad [5]$$

And by the definition of ***P*** and ***Q*** from [15] it follows that, the probability of a state sequence given the model is given as

$$P(X / \lambda) = \pi_{x_0} p_{x_0, x_1} p_{x_1, x_2}, ... p_{x_8, x_9} \qquad [6]$$

We know that,

$$P(Y, X / \lambda) = \frac{P(Y \cap X \cap \lambda)}{P(\lambda)} \qquad [7]$$

$$P(Y / X, \lambda) P(X, \lambda) = \frac{P(Y \cap X \cap \lambda)}{P(X \cap \lambda)} \frac{P(X \cap \lambda)}{P(\lambda)} \qquad [8]$$

$$P(Y, X / \lambda) = P(Y / X, \lambda) P(X / \lambda) \qquad [9]$$

From the above equations, we obtain the joint probability of the state sequence. By summing over all the possible state sequences, we obtain

$$P(Y / \lambda) = \sum_X P(Y, X / \lambda) \qquad [10]$$

$$= \sum_X P(Y / X, \lambda) P(X, \lambda) \qquad [11]$$

$$= \sum_X \pi_{x_0} q_{x_0} p_{x_0, x_1} q_{x_1}(y_1) ... p_{x_8, x_9} q_{x_9}(y_9) \qquad [12]$$

In this way the probability of an observation sequence for a given initial set of parameters is calculated. This is an iterative procedure; it is repeated until a limiting point is reached in order to acquire an optimal HMM model. This is a very tedious process because the computation is practically unfeasible. In order to solve this computation problem, an efficient procedure called the forward-backward procedure came into existence.

From Rabiner et al. [15], for the Forward-Backward procedure, a forward variable $\alpha_t(i)$ is defined as

$$\alpha_t(i) = P(O_1O_2...O_t, q_t = S_i / \lambda) \qquad [13]$$

which is the probability of the partial observation sequence, $O_1O_2...O_t$ until a particular time instance t and a particular state S_i for a given model λ. The forward variable is calculated inductively by following the three steps, namely Initialization, Induction and Termination. The procedure is as follows:

Step 1: Initializes the forward probabilities as the joint probability of state S_i and initial observation O_1.

$$\alpha_1(i) = \pi_i b(O_1), \qquad 1 \le i \le N \qquad [14]$$

Step 2: In the induction step, the forward calculation is performed which is most vital part of the procedure.

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad \begin{array}{l} 1 \le t \le T-1 \\ 1 \le j \le N. \end{array} \qquad [15]$$

Step 3: This stage gives the sum of the terminal forward variables $\alpha_T(i)$.

$$P(O / \lambda) = \sum_{i=1}^{N} \alpha_T(i). \qquad [16]$$

Therefore, the probability of the partial observation sequence given the model is nothing but the sum of forward variables. In a very similar manner, another variable called the backward variable which is defined as

$$\beta_t(i) = P(O_{t+1}O_{t+2}...O_T / q_t = S_i, \lambda) \qquad [17]$$

which is the probability of the partial observation sequence from $t+1$ to the end, for a particular state S_i for a given model λ. This variable is also calculated like the forward variable which follows the three steps namely initialization, induction and termination. The only difference is the part of the observation sequence which is considered for the probability computation. Together the forward and the backward variable are used to compute the probability of the observation sequence given the model λ. These computations were modelled by Rabiner using MATLAB in [15]. The model needs to be fed with the observation sequence, and the model parameters and the output is the probability of the observation sequence.

For the training procedure, for the given observation sequence $O = \{O_0, O_2, ..., O_9\}$, the algorithm estimates the model parameters $\lambda = (P, Q, \pi)$, to optimize the detection process. The complete procedure for the Baum-welch algorithm can be stated as follows:

Step 1: Let the initial model be λ_0

Step 2: Compute the new model λ based on λ_0 and observation sequence O.

Step 3: If $P(O/\lambda) > P(O/\lambda_0)$, stop the iteration.

Step 4: Else, set $\lambda_0 \rightarrow \lambda$ and repeat step 2.

Step 5: Stop.

This is the most difficult and also tedious process of computing the optimal model parameter λ. The Forward-Backward or Baum-Welch algorithm is an efficient algorithm used to compute the optimal model parameters which minimizes the number of computations to be performed. Having generated the observation distribution from the training set, the final step of the training phase is to estimate the parameters of the model. The parameter estimation was implemented in MATLAB with the help of routines provided by Kevin Murphy in the Hidden Markov Model Toolbox for MATLAB [14].

4.2 HMM Detection – Viterbi Algorithm

After training the HMM model, the next step is to perform detection of RAPs. The detection process was carried out by generating packet traces from the same network setup used for the training purposes. Observation distributions were extracted from the packet traces. For the detection process, we employed the Viterbi algorithm from the HMM toolbox [14]. Viterbi algorithm is a dynamic algorithm for finding the most likely sequence of hidden states called the Viterbi path which results in a sequence of observed events. This algorithm gives the optimal state sequence for a particular HMM model. For the training process, the training data set or the observation parameters is first divided into short length sequences. This is done for efficient and accurate detection process.

The Viterbi algorithm is also an iterative procedure where the optimal state sequence is obtained by recursively finding the most probable sequence of hidden states given an observation and a HMM. From Rabiner's [15], for this algorithm a variable called the best score $\delta(i,t)$ is computed, which is the highest partial probability obtained for a particular state sequence given the observation and HMM.

$$\delta(i,t) = \max_{q_1,q_2,\dots,q_{t-1}} P[q_1, q_2, \dots, q_{t-1} = i, O_1, O_2, \dots, O_t / \lambda] \qquad [18]$$

Thus, $\delta(i,t)$ is the maximum probability of all sequences ending at state i at time$t$. These probabilities are different from those probabilities obtained in the training procedure since the variable represents the probability of the most probable path to a state at a particular time for a given observation sequence.

Given an observation sequence and a trained HMM model, the complete procedure for the Viterbi algorithm can be stated as follows:

Step 1: Define a partial probability variable$\delta(i,t)$.

Step 2: For each intermediate and terminating state, the most probable path to that state is calculated which is associated with a partial probabilityδ.

Step 3: The state with the maximum partial probability and partial best path is chosen as the most probable state sequence.

Step 4: Stop.

In this way the most probable state sequence for the given observation sequence is computed or the detection process is done. The Viterbi algorithm is very similar to that of the training procedure. The major difference is the computation of the most probable path to the state. This algorithm provides a computationally efficient way of analysing observations of HMMs to find the most likely underlying state sequence.

We setup a network as illustrated in Fig. 1. Next, we present the three known DoS attacks detected by our model. There are two types of DoS attacks, logic and flooding attacks. We have mainly focused on the flooding attacks. The three DoS attacks considered in this thesis are presented in Table 1 which is explained in much detail in [16]. The attacks are generated by the end hosts connected to the RAP.

Table 1 - Attack Repertoire.

Attack	**Description**
Pod	DoS using oversized ping packe
Portsweep	Sweep through many ports determine available services o single host.
Neptune	Syn flood DoS

For our detection process, this algorithm will give the state of the access point in the network. The output of the detection procedure is a sequence of security states of the access point corresponding to each packet in the trace file. By knowing the state of the access point, a compromised access point can be discarded from the network. In the next chapter we evaluate the HMM model to analyze the accuracy and promptness of the detection process.

CHAPTER 5

SIMULATION AND RESULTS

This chapter mainly deals with all the experiments conducted for this research work. In our experiments, we use the network setup as shown in Figure. 1 to obtain the performance results in terms of detection accuracy and promptness. Details of all the experiments as well as their corresponding results are presented in this chapter.

5.1 Network Model

A small WLAN was considered in order to perform our experimental analysis which includes only three access points. The laptops are connected via IEEE 802.11b WLAN interface to the access points, and the desktops are connected via Ethernet interfaces to the router. This is a small WLAN setup in the lab using the available resources. For each access point, traffic is generated from the laptop and the desktop respectively. Different types of traffic loads or patterns were experimented in order to verify if the traffic load has an impact on the traffic characteristics which are used to detect the RAP. The different types of traffic load tested were low, medium and heavy. It was observed that the traffic load did not have a noticeable change in the packet characteristics. The traffic load does change the inter-arrival time of the packet train, but this effect does not help in the detection process. For the rogue activity, there is an unauthorized host outside the network trying to perform some intrusive activity in the network and get access into the network by getting connected to one of the access points and compromising the access point. The goal is to detect the compromised RAP attacked by the unauthorized client.

The end hosts which are connected to authorized access points generated traffic corresponding to normal web activities (browsing, email, ftp, etc). Packet arrivals in wireless LAN are modelled as Poisson process with exponential inter-arrival times [13-14]. For example, http traffic was represented by setting the web page inter-arrival time as an exponential distribution with a mean value of 60 seconds, and the number of pages also followed an exponential distribution with a mean value of 10 pages. Experiments concluded at this mean value by conducting multiple runs for different types of traffic load. It was observed that there was a minimum deviation in the mean value for each of the runs conducted.

Packet traces for the three access points in Figure. 1 were collected over a one hour time frame. The key distinguishing characteristic between the traffic generated by the normal end hosts and the rogue end hosts is the packet inter-arrival time. So we used the packet inter-arrival as the observation parameter for our HMM model. Based on the distribution of the inter-arrival times, we have identified three prominent inter-arrival ranges R1, R2 and R3. These ranges address all the traffic in the packet trace. Suppose that a set of n_t packets are identified in the training set. Let x_i denote the inter-arrival times of the i^{th} packet. The value of x_i is discretized as follows: If x_i lies within the range of R1, it takes a value of 1, for R2 the value is 2 and finally for R3 the value is 3. Thus, the observation distribution is obtained from the discretized value of x_i, $i = 1, 2, \ldots, n_t$.

Tables 2, 3 and 4 indicate the corresponding initial values of π and $\boldsymbol{P}$ parameters of the HMM model which correspond to the initial state probability distribution, state transition probability distribution and observational symbol probability for the three HMMs respectively.

Table 2 – Initial State Distribution.

State	**Access Point 1**	**Access Point 2**	**Access Point 3**
Good (G)	0.8106	0.8025	0.7913
Probed (P)	0.1376	0.1392	0.1442
Compromised (C)	0.0517	0.0583	0.0645

Table 3 - State Transition Probability.

Access Point 1			
State	**Good (G)**	**Probed (P)**	**Compromised (C)**
Good (G)	0.92	0.08	0
Probed (P)	0.08	0.754	0.166
Compromised (C)	0.2	0	0.8
Access Point 2			
State	**Good (G)**	**Probed (P)**	**Compromised (C)**
Good (G)	0.93	0.07	0
Probed (P)	0.05	0.84	0.11
Compromised (C)	0.866	0	0.133
Access Point 3			
State	**Good (G)**	**Probed (P)**	**Compromised (C)**
Good (G)	0.892	0.107	0
Probed (P)	0.117	0.764	0.117
Compromised (C)	0.866	0	0.133

Table 4 - Observational Symbol Probability.

Access Point 1			
Observations	**R1**	**R2**	**R3**
R1	0.9914	0.0081	0.0004
R2	0.303	0.472	0.225
R3	0.313	0.05	0.64
Access Point 2			
Observations	**R1**	**R2**	**R3**
R1	0.91	0.0511	0.0149
R2	0.25	0.44	0.296
R3	0.27	0.178	0.539
Access Point 3			
Observations	**R1**	**R2**	**R3**
R1	0.9654	0.0219	0.0127
R2	0.224	0.7313	0.044
R3	0.276	0.09	0.626

5.2 HMM Training and Detection Accuracy

The training procedure was explained in the above sections. The first step in the training procedure is to initialize the parameters and use the Baum-Welch algorithm implemented by Rabiner in [15]. With the initial parameters the HMM is trained to obtain a learned HMM that is to be implemented in the WLAN for the detection process. Tables 2, 3 and 4 gave the initial values of the initial state probability, the state transition probability and the observation symbol probability. Tables 5 and 6 give the trained state transition probability and observation symbol probability.

Table 5 – Trained State Transition Probability.

Access Point 1			
State	**Good (G)**	**Probed (P)**	**Compromised (C)**
Good (G)	0.72	0.28	0
Probed (P)	0.08	0.654	0.266
Compromised (C)	0.2	0	0.8
Access Point 2			
State	**Good (G)**	**Probed (P)**	**Compromised (C)**
Good (G)	0.73	0.27	0
Probed (P)	0.05	0.64	0.31
Compromised (C)	0.666	0.101	0.233
Access Point 3			
State	**Good (G)**	**Probed (P)**	**Compromised (C)**
Good (G)	0.774	0.26	0
Probed (P)	0.117	0.564	0.217
Compromised (C)	0.655	0.112	0.233

Table 6 – Trained Observation Probability.

Access Point 1			
Observations	**R1**	**R2**	**R3**
R1	0.9914	0.0081	0.0004
R2	0.303	0.472	0.225
R3	0.313	0.05	0.64
Access Point 2			
Observations	**R1**	**R2**	**R3**
R1	0.9731	0.012	0.0149
R2	0.25	0.44	0.296
R3	0.539	0.178	0.28
Access Point 3			
Observations	**R1**	**R2**	**R3**
R1	0.9654	0.0219	0.0127
R2	0.224	0.7313	0.044
R3	0.276	0.09	0.626

Detection accuracy is evaluated by computing the successful detection of RAP, false positives, and false negatives. In our experiment setup, the end hosts are laptops which communicate via IEEE 802.11b WLAN interface to the access points. The source of the rogue activity is a "roaming" laptop carrying out attacks depicted in Table 1. An access point which is currently under attack by this "roaming laptop" is a

RAP. The goal of our model is to report all instances of the presence of RAP. Traffic was generated from all laptops for a period of 60 minutes. The traffic was collected at the gateway router, and an offline detection process was carried out using the trained HMM model.

In order to provide a quantitative analysis of the model, we evaluated the detection accuracy of the model. In Figure.3 the detection accuracy of the HMM model is presented for all three access points. A combination of normal and attacked traffic flows from each of these access points at different time instances during the 60 minute time limit. The three access points exhibit all three security states during experiment time limit. The detection accuracy measures the effectiveness of the model to detect the compromised security state. The detection accuracy is computed by analyzing how accurately the HMM model detected a compromised security state from the observation sequence during the 60 minute time limit. A compromised security state indicates that the access point is acting as a RAP. The detection accuracy is consistent for all the three access points. The model exhibits 85 % accuracy with very slight variance. There are several ways of improving the detection accuracy of an access point by optimizing the model parameters, varying the observation sequence length, varying the number of states in the HMM model etc. It is also very true that with a larger training set, the detection accuracy of the model will improve. The reason behind larger training data is the model analyzes more observations or outputs in order to achieve higher likelihood of the security state of the access point. Sometimes the accuracy also depends on the range of these access points because it depends on the information obtained from the packet traces of each host via an access point.

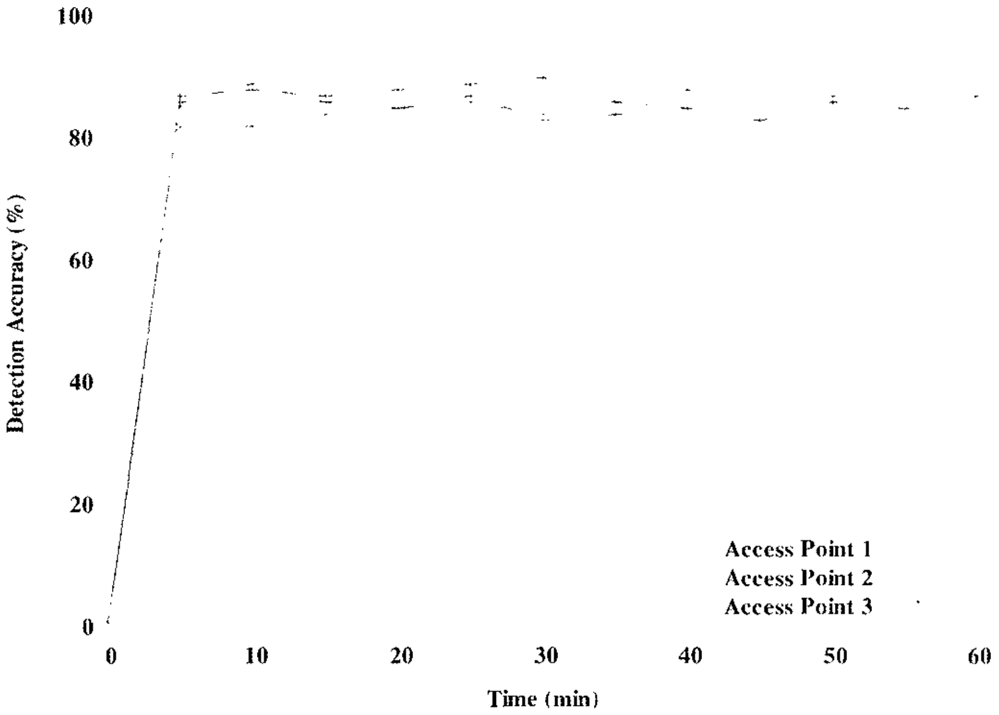

Figure 3. Detection Accuracy.

Figure. 4 illustrates the number of false positives encountered during the detection process. False positives indicate misidentification of an authorized access points as a RAP. Initially, the HMM model first analyzes the packet trace slowly and then starts detecting the most probable state. Figure. 4 show that our HMM model maintained a very low false positive ratio of 8.5 - 10 %. There is definitely room for decreasing the false positive ratios by optimizing the HMM model. The false positive ratio is calculated by measuring the number of times the HMM misidentifies an authorized access point as a RAP during the detection process in the whole 60 minute time period.

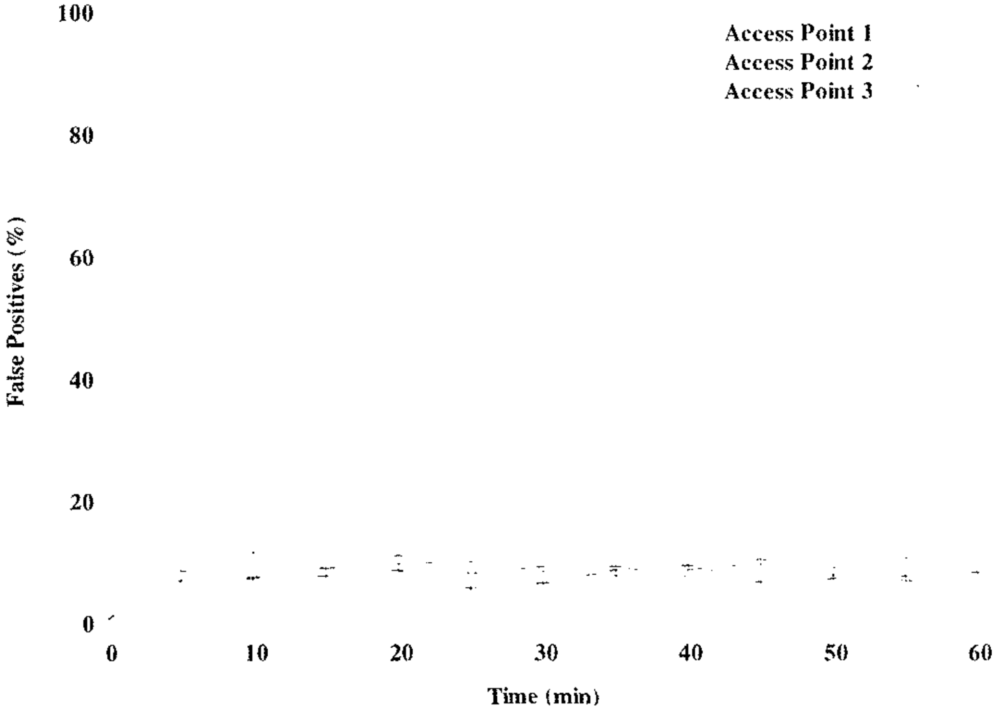

Figure 4. False Positives.

Figure. 5 illustrates the number of false negatives encountered during the detection process. False negatives indicate miss detection of the presence of RAP. As already mentioned the HMM model takes a few moments to stabilize and then starts its detection process. The false negative ratio is calculated by measuring the number of times the HMM misses the detection of a RAP and misidentifies it as an authorized access point during the detection process in the whole 60 minute time period. The average number of false positives and false negatives are close to 8 % with slight variance. The number of false positives and false negatives can be decreased further if the training is performed on a larger training set.

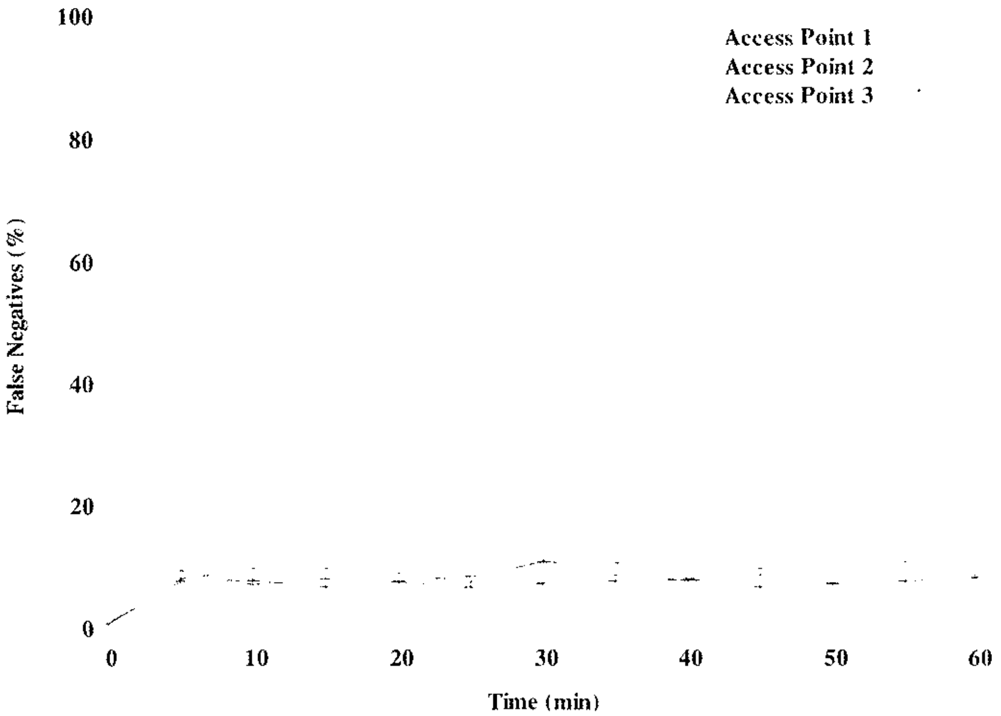

Figure 5. False Negatives.

Finally in Table 7, we demonstrate the promptness of the detection process. Four attack instances are identified in the packet traces. The detection time in milliseconds for two access points are reported. The presence of a RAP is detected within less than a second. The quick detection of a RAP is equally important as increasing the detection accuracy. The detection time is computed by measuring the time taken by an access point to detect a RAP. As already mentioned, the HMM takes a few seconds to actually determine the state of an access point.

Table 7 - Detection Time.

Attack	**Detection Time (milliseconds)**	
Instance	**Access Point 1**	**Access Point 2**
1	345	552
2	797	66
3	797	537
4	803	174

5.3 Detection Accuracy for Varied Sequence Lengths

The sequence length is defined as the length of the observation sequence taken into consideration by the Viterbi algorithm In Figures 3-5, the sequence length was equal to 10 packets. In this section, we present simulation results with sequence lengths varying in the range of 10 to 60.

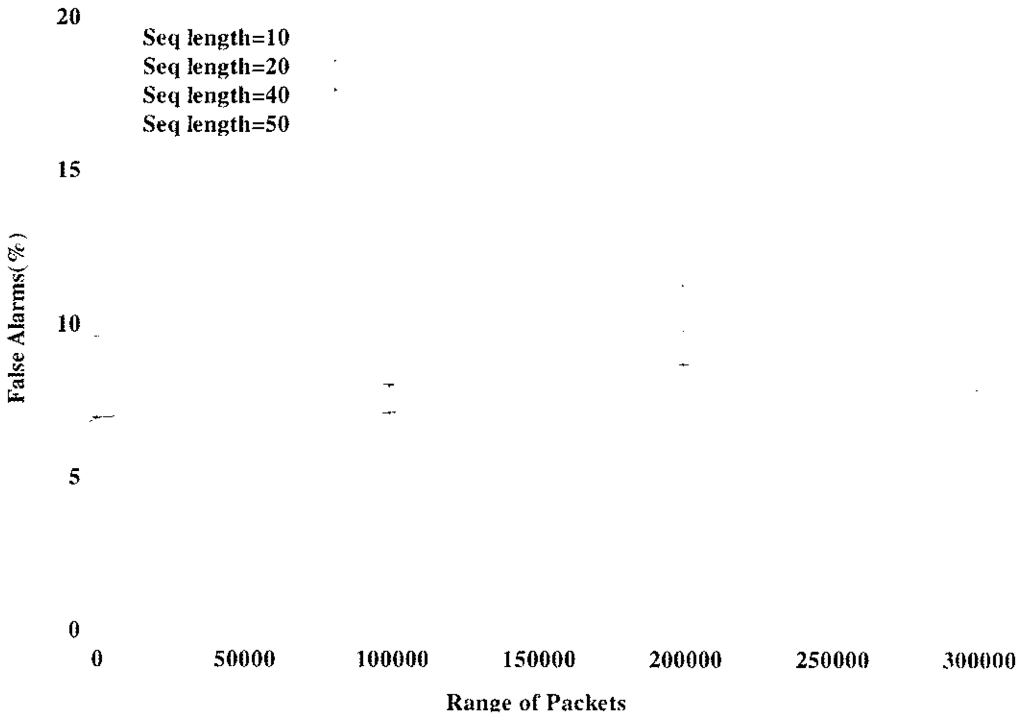

Figure 6. False Alarms.

Figure 6 illustrates the number of false alarms (false positives) encountered during the detection process at AP3. It can be observed that the number of false negatives is not affected by varying the sequence lengths. This property ensures that our HMM is invariant to variation in sequence lengths. With a large training data set, it is possible that larger sequence lengths will be used for the detection process. Thus, the detection accuracy of our HMM will not be affected by larger training datasets.

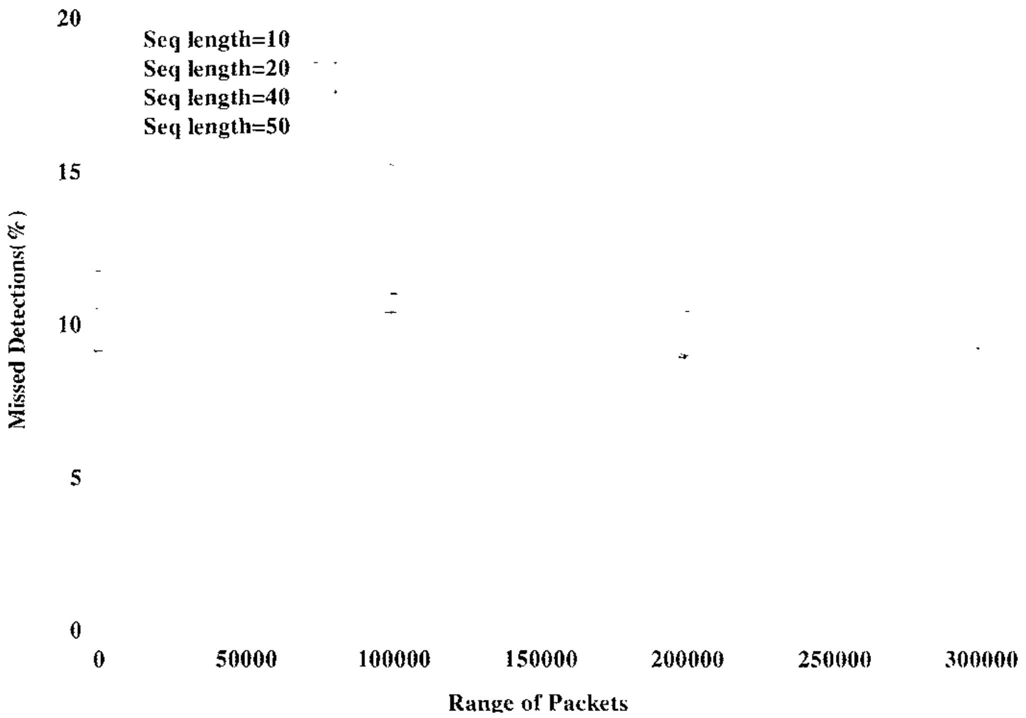

Figure 7. Missed Detections.

Figure 7 illustrates the number of missed detections (false negatives) encountered during the detection process. Similar to Figure 6, the numbers of false negatives are also not affected by varied sequence lengths. This experiment was conducted in order to improve the detection accuracy of the HMM model. Thus, this experiment proved that by varying the sequence length, there is not much improvement in the efficiency of the model.

The complexity of our HMM model depends on the number of states and the number of observation parameters for each of these states and finally, the training procedure which plays a vital role in the accurate modeling of the HMM. For our experiments, we considered a very small WLAN. The scalability issue for our model definitely increases the complexity of the model. Considering a university network or a large organization where there are hundred's of access points, the HMM model cannot be so broad, i.e. the number of states and the observations related to each of these access points would definitely increase. The increase in the number of states is because the HMM model should be able to model all the states of the access point since our HMM is an access point level model. For each access point a HMM has to be implemented in order to perform the detection process.

In this way HMM proved to be an efficient tool in the detection of RAPs in a WLAN. The detection accuracy of the model can be improved by changing the states of the HMM or by modifying the parameters of the model. The most difficult and important problems are the learning or training of the HMM model because the detection process mainly depends on how well the model is trained. Most of the applications using HMM mainly focus on the training procedure. A major limitation of the HMM is the assumption that the successive observations are independent unlike the Markov models. However, in spite of these limitations, HMM has proved to be a successful statistical tool in most of pattern recognition applications.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we designed an efficient and prompt HMM to detect the presence of RAPs in a WLAN. The HMM model is implemented at the gateway router where traffic is captured and analysed. Our approach is comprised of two stages. The first stage is the training of a HMM, and second stage is the detection of RAP based on the trained HMM. The presence of a RAP in our network is due to end hosts performing three specific DoS attacks. Our model is capable of detecting a RAP whenever an end host performs any of the DoS attack mentioned in this research work. The detection accuracy and promptness of the HMM has been evaluated by performing experimental results. The presence of RAP is detected within one second and the average detection accuracy is 85%. Our experimental results also showed that our HMM model performed this detection process with very low missed detections and false alarms. An experiment to improve the efficiency of our HMM model was conducted by varying the sequence length for the detection process. The results showed that the variation in the sequence length did not improve the detection efficiency. In our future work, we plan to improve the detection accuracy of our model by focusing on other parameters like the model parameters, number of states in the model etc. The performance of the model will be evaluated using different network setups and various traffic scenarios. The most difficult and crucial part of HMMs is to train the model.

For our future work in this field, we would like to design another efficient HMM by defining it with different states and parameters. In this model, we would consider the access points in a network as the different states in a HMM which would make the detection process more efficient and straightforward. This model should be able to detect RAP not only the rogue activity. This would be a powerful tool to detect RAPs in large networks because of its model which is a doubly embedded stochastic process with an underlying stochastic process that is not observable(it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations. Such a model would be very efficient in detecting RAPs.

REFERENCES

[1] White Paper, "Rogue Access Point Detection: Automatically Detect and Manage Wireless Threats to your Network". [Online] Available at http://www.proxim.com.

[2] Myung-Sup Kim; Hun-Jeong Kong; Seong-Cheol Hong; Seung-Hwa Chung; Hong, J.W., "A flow-based method for abnormal network traffic detection," *Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP*, vol.1, pp.599-612, April 2004.

[3] Liran Ma, Amin Y. Teymorian, Xiuzhen Cheng, and Min Song, ``RAP: Protecting Commodity Wi-Fi Networks from Rogue Access Points," *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*, Vancouver, British Columbia, August 2007.

[4] "AirDefense enterprise: a wireless intrusion prevention system."[Online]. Available: http://www.airdefense.net/.

[5] "AirMagnet: Enterprise WLAN management" [Online]. Available: http://www.airmagnet.com/.

[6] Adya, A., Bahl, P., Chandra, R., and Qiu, L. 2004. "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks." In *Proceedings of the 10th Annual international Conference on Mobile Computing and Networking* (Philadelphia, PA, USA, 2004, New York, NY, pp.30-44, MobiCom '04 ACM, October 2004.

[7] Bahl, P., Chandra, R., Padhye, J., Ravindranath, L., Singh, M., Wolman, A., and Zill B. "Enhancing the security of corporate Wi-Fi networks using DAIR." In *Proceedings of the 4th international Conference on Mobile Systems, Applications*

and Services (Uppsala, Sweden)., New York, NY, pp.1-14, MobiSys '06. ACM, June 2006.

[8] Beyah, R.; Kangude, S.; Yu, G.; Strickland, B.; Copeland, J., "Rogue access point detection using temporal traffic characteristics," *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, vol.4, pp. 2271-2275, 29 November - 3 December 2004.

[9] Shetty, Sachin; Song, Min; Ma, Liran, "Rogue Access Point Detection by Analyzing Network Traffic Characteristics," *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pp.1-7, October 2007.

[10] Wei, W., Suh, K., Wang, B., Gu, Y., Kurose, J., and Towsley, D. 2007. "Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs." In *Proceedings of the 7th ACM SIGCOMM Conference on internet Measurement* (San Diego, California, USA), New York, NY, pp.365-378, IMC '07. ACM, October 2007.

[11] Hongda Yin, Guanling Chen and Jie Wang, "Detecting protected layer-3 rogue APs," *Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference*, pp.449-458, September 2007.

[12] Nong Ye. "A markov chain model of temporal behavior for anomaly detection," In Proceedings of the 2000 IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, 2000.

[13] Yu Zheng; Kejie Lu; Dapeng Wu; Yuguang Fang, "Performance Analysis of IEEE 802.11 DCF in Imperfect Channels," *Vehicular Technology, IEEE Transactions on*, vol.55, no.5, pp.1648-1656, September 2006.

[14] Kevin Murphy, "Hidden Markov Model (HMM) Toolbox for MATLAB," online at http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html.

[15]Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. In *Readings in Speech Recognition*, A. Waibel and K. Lee, Eds. Morgan Kaufmann Publishers, San Francisco, CA, pp.267-296, 1990.

[16]David K. Y. Yau, John C. S. Lui, Feng liang, Yeung Yam, "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles," *IEEE/ACM Trans. Netw.* 13, pp.29-42, February 2005.

[17]Svetlana Radosavac and John S. Baras, "*Detection and Classification of Network Intrusions using Hidden Markov Models*", 37th Conference on Information Sciences and Systems (CISS), Baltimore, March 2003.

[18]Krishna Nandivada, V.; Palsberg, J., "Timing analysis of TCP servers for surviving denial-of-service attacks," *Real Time and Embedded Technology and Applications Symposium, 2005. RTAS 2005. 11th IEEE* , pp. 541-549, 7-10 March 2005.

[19]Schmoyer, T.R.; Yu Xi Lim; Owen, H.L., "Wireless intrusion detection and response: a classic study using main-in-the-middle attack," *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol.2, pp. 883-888, 21-25 March 2004.

[20]Wei, W., Wang, B., and Towsley, D. 2002. Continuous-time hidden Markov models for network performance evaluation. *Perform. Eval.* 49, pp.129-146, 1-4 September 2002.

[21]J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless lan monitoring and its applications," in *WiSe '04*, ACM Press, 2004, pp. 70–79.

[22]Tomko, A.A.; Rieser, C.J.; Buell, L.H., "Physical-Layer Intrusion Detection in Wireless Networks," *Military Communications Conference. MILCOM 2006. IEEE*, pp.1-7, 23-25 October 2006.

[23]WaveLink, "Rogue Access Point Detection – Automatically detect and mange wireless threats to your network", Available: http://www.nowire.se/produktblad/Proxim/Rogue_Access_Point_Detection.pdf.

[24]Bahl, P., Chandra, R., Padhye, J., Ravindranath, L., Singh, M., Wolman, A., and Zill, B. 2006. Enhancing the security of corporate Wi-Fi networks using DAIR. In *Proceedings of the 4th international Conference on Mobile Systems, Applications and Services* (Uppsala, Sweden), New York, NY, pp.1-14, MobiSys '06. ACM, June 2006.

[25]Chirumamilla, Mohan. K and Ramamurthy. B, "Agent based intrusion detection and response system for wireless LANs," Communications, 2003. *ICC '03. IEEE International Conference on,* vol. 1, pp.492-296, 11-15 May 2003.

[26]S. Zanero, "Improving Self Organizing map performance for Network Intrusion Detection", Available: http://home.dei.polimi.it/zanero/papers/ids-perform.pdf.

[27]A. Arnes, F. Valeur, G. Vigna and R. Kemmerer, "Using Hidden Markov Models to evaluate the risks of intrusions," Available: http://www.cs.ucsb.edu/~vigna/publications/2006_arnes_valeur_vigna_kemmerer_RAID.pdf .

[28] Salamatian, K. and Vaton, S. 2001. Hidden Markov modeling for network communication channels. *SIGMETRICS Perform. Eval. Rev.* 29, pp.92-101, Jun. 2001.

VITA

GAYATHRI SHIVARAJ

gshiv001@odu.edu ♦ P: 757-597-1757 ♦ 1055W 48th Street, Apt 30, Norfolk, VA 23508

- **OBJECTIVE:**

To utilize my skills which include organizational and business skills, data processing and analysis, database administration, management analysis, operational research, and technical writing in Software Engineering.

- **TECHNICAL SKILLS:**

Programming Languages:
C, C++, JAVA, LINUX , HTML, XML, SQL, J2EE, JavaScript, Socket Programming.
Operating Systems:
Windows, UNIX, Macintosh
Other:
MS Access, MATLAB, NS-2, LATEX, compiler and language design, user interfaces, databases, client/server, telecommunications, debugging, reporting, and optimization, Intel 8086 Assembler, MS/DOS.

- **EDUCATION:**

Masters of Science in Electrical Engineering, *Old Dominion University,* Norfolk, VA.
Graduation date: May 2009.
Bachelors of Science, Electronics & Communications, *Muffakham Jah College of Engineering and Technology,* Hyderabad, Andhra Pradesh, India.
Graduation date: May 2006.

- **COURSE WORK:**

Electrical and Computer Engineering: Computer Networks, Digital Communications, Computer Communication Networks, Wireless Communication Networks, Linear systems, Statistical analysis and Simulations, Network Security – Rogue access point detection (Independent study), Engineering Systems Modeling, Linear Systems.
Submitted and Published a Conference Paper in MILCOM '08, San diego, California, "A HIDDEN MARKOV MODEL BASED APPROACH TO DETECT ROGUE ACCESS".

- **EMPLOYMENT HISTORY:**

Software Engineer.

1. Responsible for supporting an entire banking application, which is developed on different Operating systems like Unix, Sun-Solaris, AIX etc.
2. Experience in maintaining various databases using SQL.
3. Experience in managing web sites and documentation, using HTML, to manage various projects and also user documentation.

- **NON-TECHNICAL SKILLS:**

Activities and Awards:

1. Organized and participated in the prestigious **STAC** (Student Technical Awareness Conference) and **SPAC** (Student Professional Awareness Conference), conducted by the **IEEE** student chapter of Osmania University.
2. Chief Organizer of the technical paper conferences of **IEEE** student chapter of Osmania University.
3. Received Travel Grant for the conference paper published in MILCOM '08, California.

- **REFERENCES:** Available upon request.