Research on Intrusion Detection and Response: A Survey

Peyman Kabiri and Ali A. Ghorbani
(Corresponding author: Ali A. Ghorbani)

Faculty of Computer Science, University of New Brunswick,
Fredericton, NB, E3B 5A3, Canada
(Email: {kabiri, ghorbani}@unb.ca)

(Received June 15, 2005; revised and accepted July 4, 2005)

Abstract

With recent advances in network based technology and increased dependability of our every day life on this technology, assuring reliable operation of network based systems is very important. During recent years, number of attacks on networks has dramatically increased and consequently interest in network intrusion detection has increased among the researchers. This paper provides a review on current trends in intrusion detection together with a study on technologies implemented by some researchers in this research area. Honey pots are effective detection tools to sense attacks such as port or email scanning activities in the network. Some features and applications of honey pots are explained in this paper.

Keywords: Detection methods, honey pots, intrusion detection, network security

1 Introduction

In the past two decades with the rapid progress in the Internet based technology, new application areas for computer network have emerged. At the same time, wide spread progress in the Local Area Network (LAN) and Wide Area Network (WAN) application areas in business, financial, industry, security and healthcare sectors made us more dependent on the computer networks. All of these application areas made the network an attractive target for the abuse and a big vulnerability for the community. A fun to do job or a challenge to win action for some people became a nightmare for the others. In many cases malicious acts made this nightmare to become a reality.

In addition to the hacking, new entities like worms, Trojans and viruses introduced more panic into the networked society. As the current situation is a relatively new phenomenon, network defenses are weak. However, due to the popularity of the computer networks, their connectivity and our ever growing dependency on them, realization of the threat can have devastating consequences. Securing such an important infrastructure has become the priority one research area for many researchers.

Aim of this paper is to review the current trends in Intrusion Detection Systems (IDS) and to analyze some current problems that exist in this research area. In comparison to some mature and well settled research areas, IDS is a young field of research. However, due to its mission critical nature, it has attracted significant attention towards itself. Density of research on this subject is constantly rising and everyday more researchers are engaged in this field of work. The threat of a new wave of cyber or network attacks is not just a probability that should be considered, but it is an accepted fact that can occur at any time. The current trend for the IDS is far from a reliable protective system, but instead the main idea is to make it possible to detect novel network attacks.

One of the major concerns is to make sure that in case of an intrusion attempt, the system is able to detect and to report it. Once the detection is reliable, next step would be to protect the network (response). In other words, the IDS system will be upgraded to an Intrusion Detection and Response System (IDRS). However, no part of the IDS is currently at a fully reliable level. Even though researchers are concurrently engaged in working on both detection and respond sides of the system. A major problem in the IDS is the guarantee for the intrusion detection. This is the reason why in many cases IDSs are used together with a human expert. In this way, IDS is actually helping the network security officer and it is not reliable enough to be trusted on its own. The reason is the inability of IDS systems to detect the new or altered attack patterns. Although the latest generation of the detection techniques has significantly improved the detection rate, still there is a long way to go.

There are two major approaches for detecting intrusions, signature-based and anomaly-based intrusion detection. In the first approach, attack patterns or the

behavior of the intruder is modeled (attack signature is modeled). Here the system will signal the intrusion once a match is detected. However, in the second approach normal behavior of the network is modeled. In this approach, the system will raise the alarm once the behavior of the network does not match with its normal behavior. There is another Intrusion Detection (ID) approach that is called specification-based intrusion detection. In this approach, the normal behavior (expected behavior) of the host is specified and consequently modeled. In this approach, as a direct price for the security, freedom of operation for the host is limited. In this paper, these approaches will be briefly discussed and compared.

The idea of having an intruder accessing the system without even being able to notice it is the worst nightmare for any network security officer. Since the current ID technology is not accurate enough to provide a reliable detection, heuristic methodologies can be a way out. As for the last line of defense, and in order to reduce the number of undetected intrusions, heuristic methods such as Honey Pots (HP) can be deployed. HPs can be installed on any system and act as trap or decoy for a resource.

Another major problem in this research area is the speed of detection. Computer networks have a dynamic nature in a sense that information and data within them are continuously changing. Therefore, detecting an intrusion accurately and promptly, the system has to operate in real time. Operating in real time is not just to perform the detection in real time, but is to adapt to the new dynamics in the network. Real time operating IDS is an active research area pursued by many researchers. Most of the research works are aimed to introduce the most time efficient methodologies. The goal is to make the implemented methods suitable for the real time implementation.

From a different perspective, two approaches can be envisaged in implementing an IDS. In this classification, IDS can be either host based or network based. In the host based IDS, system will only protect its own local machine (its host). On the other hand, in the network based IDS, the ID process is somehow distributed along the network. In this approach where the agent based technology is widely implemented, a distributed system will protect the network as a whole. In this architecture IDS might control or monitor network firewalls, network routers or network switches as well as the client machines.

The main emphasis of this paper is on the detection part of the intrusion detection and response problem. Researchers have pursued different approaches or a combination of different approaches to solve this problem. Each approach has its own theory and presumptions. This is so because there is no exact behavioral model for the legitimate user, the intruder or the network itself.

Rest of this paper is organized as follows: In Section 2, intrusion detection methodology and related theories are explained. Section 3 presents the system modeling approaches. In Section 4, different trends in IDS design are presented. Section 5 describes the feature selection/extraction methods implemented in this area. In Section 6, application of honey pots in the network security will be discussed. Finally, conclusions and future work are given in Section 7 and Section 8.

2 Intrusion Detection

The first step in securing a networked system is to detect the attack. Even if the system cannot prevent the intruder from getting into the system, noticing the intrusion will provide the security officer with valuable information. The Intrusion Detection (ID) can be considered to be the first line of defense for any security system.

2.1 Artificial Intelligence and Intrusion Detection

Application of the artificial intelligence is widely used for the ID purpose. Researchers have proposed several approaches in this regard. Some of the researchers are more interested in applying rule based methods to detect the intrusion. Data mining using the association rule is also one of the approaches used by some researchers to solve the intrusion detection problem. Researchers such as Barbara et al. [4, 5], Yoshido [43] and Lee et al. [30] have used these methods.

Others have proposed application of the fuzzy logic concept into the intrusion detection problem area. Works reported by Dickerson et al. [16], Bridges et al. [8] and Botha et al. [7] are examples of those researchers that follow this approach. Some researchers even used a multidisciplinary approach, for example, Gomez et al. [18] have combined fuzzy logic, genetic algorithm and association rule techniques in their work. Cho [12] reports a work where fuzzy logic and Hidden Markov Model (HMM) have been deployed together to detect intrusions. In this approach HMM is used for the dimensionality reduction. Due to its nature, the data mining approach is widely appreciated in this field of research.

Some researchers have tried to use the Bayesian methodology to solve the intrusion detection problem. The main idea behind this approach is the unique feature of the Bayesian methodology. For a given consequence, using the probability calculations Bayesian methodology can move back in time and find the cause of the events. This feature is suitable for finding the reason for a particular anomaly in the network behavior. Using Bayesian algorithm, system can somehow move back in time and find the cause for the events. This algorithm is sometimes used for the clustering purposes as well. Reported works from researchers such as Bulatovic et al. [9], Barbara et al. [5] and Bilodeau et al. [6] are examples of this approach.

Although using the Bayesian for the intrusion detection or intruder behavior prediction can be very appealing, however, there are some issues that one should be concerned about them. Since the accuracy of this method

is dependent on certain presumptions, distancing from those presumptions will decrease its accuracy. Usually these presumptions are based on the behavioral model of the target system. Selecting an inaccurate model may lead to an inaccurate detection system. Therefore, selecting an accurate model is the first step towards solving the problem. Unfortunately due to the complexity of the behavioral model within this system finding such a model is a very difficult task. This paper will address the system modeling in the following section.

Researchers such as Zanero et al. [44], Kayacik et al. [23] and Lei et al. [32] find the Artificial Neural Network (ANN) approach more appealing. These researchers had to overcome the curse of dimensionality for the complex systems problem. A suitable method is the Kohonen's Self Organizing features Map (SOM) that they have proposed. Hu et al. [20] reports an improvement to the SOM approach used by Kayacik et al. [23], where the Support Vector Machine (SVM) method has been implemented to improve SOM. Using SOM will significantly improve the sensitivity of the model to the population of the input features. Zanero et al. [44] use the SOM to compress payload of every packet into one byte.

The main goal of using the ANN approach is to provide an unsupervised classification method to overcome the curse of dimensionality for a large number of input features. Since the system is complex and input features are numerous, clustering the events can be a very time consuming task. Using the Principle Component Analysis (PCA) or Singular Value Decomposition (SVD) methods can be an alternative solution [2]. However, if not used properly both of these methods can become computationally expensive algorithms. At the same time, reducing the number of features will lead to a less accurate model and consequently it will reduce the detection accuracy.

In the computer networks intrusion detection problem area, the size of the feature space is obviously very large. Once the dimensions of the feature space are multiplied by the number of samples in the feature space, the result will surely present a very large number. This is why some researchers either select a small sampling time window or reduce the dimensionality of the feature space. Since the processing time is an important factor in the timely detection of the intrusion, the efficiency of the deployed algorithms is very important. Time constraint may sometimes force us to have the less important features pruned (dimensionality reduction). However, the pruning approach is not always possible. Implementing data mining methodology, some researchers have proposed new data reduction approaches. Data compression can be considered to be an alternative approach to solve the high dimensionality problem. Generation of association rules as it was proposed by Lee et al. [30, 31] is an alternative to reduce the size of the input data (Rule based approach).

Size and dimensionality of the feature space are two major problems in IDS development. At the same time, methods such as Bayesian and HMM that use statistical or probability calculations can be very time consuming. Besides the dimensionality reduction or the data compression methods, there are two other methods that can deal with the problem of computation time. These methods are explained in the following subsections.

2.2 Embedded Programming and Intrusion Detection

One approach is to preprocess the network information using a preprocessor hardware (front-end processor). In this method some parts of the processing is performed prior to the IDS. This preprocess will significantly reduce the processing load on the IDS and consequently the main CPU. Otey et al. [37] have reported a similar work by programming the Network Interface Card (NIC). This approach can have many properties including lower computational traffic and higher performance for the main processor. Implementing this approach will make it easier to detect variety of attacks such as Denial of Service (DoS) attack. This is because the NIC is performing the major part of the processing while the main processor only monitors the NIC operation.

2.3 Agent Based Intrusion Detection

The second approach is the distributed or the agent based computing. In this approach not only the workload will be divided between the individual processors, but also the IDS will be able to obtain an overall knowledge of the networks working condition. Having an overall view of the network will help the IDS to detect the intrusion more accurately and at the same time it can respond to the threats more effectively. In this approach, servers can communicate with one another and can alarm each other.

In order to respond to an attack, sometimes it can be sufficient enough to disconnect a subnet. In this type of system in order to contain a threat, the distributed IDS can order severs, routers or network switches to disconnect a host or a subnet. One of the concerns with this type of system is the extra workload that the IDS will enforce on the network infrastructure. The communication between the different hosts and servers in the network can produce a significant traffic in the network. The distributed approach can increase the workload of the network layers within the hosts or servers and consequently it may slow them down.

There are two approaches in implementing an agent based technology. In the first approach, autonomous distributed agents are used to both monitor the system and communicate with other agents in the network. A Multi-agent based system will enjoy a better perception of the world surrounding it. Zhang et al. [46] report implementing a multi-agent based IDS where they have considered four types of agents: Basic agent, Coordination agent, Global Coordination agent, Interface agents. Each one of these agents performs a different task and has its own subcategories. For example, the basic agent includes:

Workstation agents, Network segment agents and Public server agents. These subcategory agents respectively work on the workstations of the network, as well as, the subnet level and public server level (Mail agent or FTP agent). In this way, the complex system with breakdown into much simpler systems and will become easier to manage.

In the second approach, mobile agents are used to travel through the network and collect information or to perform some tasks. Foo et al. [17] report an IDS development work [15] using mobile agents. They use the Mitsubishi's Concordia platform in their work to develop a mobile agent based IDS. Using the mobile agent, their IDS performs both the port scanning and the integrity checks on the critical files of the system. The proposed agent based IDS will raise the alarm if it detects any alteration on the critical files of the system. Mobile agents can be sent to other systems to monitor health of the target system and to collect information.

Luo et al. [33] introduce a new Mobile Agent Distributed IDS (MADIDS). Authors address number of deficiencies that exist in distributed IDSs: "The overload of data transmission", "The computation bottleneck of the central processing module" and "The delay of network transmission". Paper reports that one of the main goals of the system is to improve the performance of the IDS in regard to speed and network traffic.

In a work reported by Ramachandran et al. [38] the idea of neighborhood-watch is implemented for the network security. There are three different types of agents in three different layers. All the agents are defined in PERL (Practical Extraction and Report Language). In the front line (bottom layer) there is a Cop agent that is a mobile agent. There are different types of Cop agents dependent on their assignments. A Cop agent is responsible for collecting data from various sites and reporting them to its respective detective agent. In this system, each site will store all the important security information about its neighbors. This information includes checksum of critical data files and system binaries, etc. It will also store a list of its neighbors in the neighborhood. There are neighbors (hosts) within each neighborhood (subnet) whom can be inspected by the mobile agents called Cops. By voting among themselves, neighbors will decide on the course of action they intend to follow. This concept will be discussed in more detail in the following sections.

2.4 Software Engineering and Intrusion Detection

As the complexity of the IDS increases, the problem of developing the IDS becomes more and more difficult. A programming language dedicated to developing IDSs can be useful for the developer community. Such a programming language with its special components will improve the programming standard for the IDS code. IDS developers can enjoy the benefits of a new language dedicated to the IDS development. Such a language will improve both the programming speed and the quality of the final code.

In a paper by Vigna et al. [41] the main attention is focused on the software engineering aspect of the IDS. Issues such as object-oriented programming, component reusability and the programming language for the IDS are discussed in this paper. A new framework called State Transition Analysis Technique (STAT) is introduced in this paper. In their implemented framework, Vigna et al. [41] propose a type of state machine system called STAT that follows the state transition of the attack patterns. This framework is for developing signature based IDSs (The concept of the signature based IDS will be discussed later in this paper). There is a STAT-Response class that holds response modules. These response modules include library of actions that are associated with the pattern of the attack scenarios. All together, this language will produce an encapsulated object-oriented code with a high reusability in the code. There is an event provider module that will provide the framework with the events occurring on the network.

Another approach in programming languages for the IDS is to provide means to follow the state change in the system. In this way, the IDS will have the ability to have its behavior altered if necessary. Including this feature in the IDS will make it adaptive and reconfigurable. Possibility to alter the behavior of the IDS will provide us with a dynamically reconfigurable IDS. In a reported work, Sekar et al. [39] have implemented a *State Machine Language (SML)* approach based on the *Extended Finite State Automata (EFSA)* to model the correct or expected behavior of the network. Using a well designed program in SML, the state machine will be able to follow up with the events within the network and to produce appropriate outputs. If no irregularities detected, then the anomaly detection part of the process will analyze the outputs and will detect the anomalies.

There are two approaches in implementing an IDS. In the first approach, IDS is implemented in the form of software that is deployed on a server or a host. In this approach the final produce is not a physical object but it is software. In the second approach the IDS is built as a product with its own hardware platform (IDS appliance). In this type of IDS, once the product is installed on the network it will connect itself to the network and will start monitoring and analyzing the network. IDS can perform its duties in a way transparent to the network. Such approaches could help the IDS to perform the intrusion detection in a more successful and non-intrusive way. At the same time, this type of products are easier to install and will introduce minimum overhead on the network. Thus, their price might be higher.

2.5 Some Selected Papers

This section will describe selected papers in different research areas of the IDS technology.

2.5.1 Bayesian (Statistical) Approach

As an example for the implementation of the Bayesian method in IDS, Barbara et al. [5] report a work on the subject of intrusion detection for the anomaly detection. Authors report similar categories (misuse and anomaly detection for intrusion detection), they also report the same features for these two methodologies. In order to be able to handle unknown attacks they have selected the anomaly detection method. Their aim is to improve the detection and false alarm rates generated by the system. Their report indicates that this work is the continuation of an ongoing research based on "an anomaly detection system called Audit Data Analysis and Mining" (ADAM). Their approach is mainly data mining oriented but in this paper the reported work is related to the pseudo-Bayes estimators. The application for these estimators is to estimate the priori and posteriori probabilities of new attacks. In this work, Naive-Bayesian classifier is used to classify network instances. They also claim that due to the properties of the pseudo-Bayes estimators, system won't need any priori knowledge regarding the new attack patterns.

ADAM consists of three parts. Part one is the preprocessor and its job is to collect data from the TCP/IP traffic data (network sniffer). The second part is the data mining engine that extracts association rules from the collected data. Data mining engines main job is to search for unexpected behaviors. ADAM works in two modes: Training and detection modes. The last part of the system is the classification engine and its task is to classify the association rules into two classes: Normal and abnormal. Abnormal classes can be later linked to some attacks.

Authors report two main advantages for the system, first the ability to work in real time (online data mining operation) and then the strategy of anomaly detection of the system. In their system, rules depict behavior models. These rules are saved in a database and constantly monitored. If a rule is a new rule and not yet registered in the database (anomaly) and its occurrences have reached to a threshold value, then it will be labeled by the system as a suspicious event. The mining engine works in three levels: single level, domain level and feature level.

Single level mining engine works in two different modes: static mining and dynamic mining. The first one is for the normal operation time of the system when a profile is made for the system behavior. The second one however "uses a sliding window method that implements incremental, on-line associated rule mining"[5].

In the domain level mining engine, the source and destination IPs are monitored. The reported system may find it suspicious if both the source and destination IP's come from the same subnet. In the feature selection engine, a windowing technique is implemented to record instances of the network (every window is 3 seconds wide). In this way, system collects snap shots from the network behavior and then analyzes them.

There is also a second slower sampling rate that is every 24 hours to detect the slow occurring but long lasting anomalies. Then the system will apply the domain level mining methods on them to capture the rules and extract features. In the reported work, a selected number of attributes in the training data were reported to characterize classes. These classes reflect properties resulted during different levels of the data mining. Classifier is trained using the training data and later on is tested using the test data.

In the reported work, a pseudo Bayesian classifier is used for the classification. The estimation part of this classifier has the smoothing feature. The pseudo Bayesian estimator is a popular method in discrete multivariate analysis. In the reported work, Barbara et al. [5] use Dirichlet distribution probability density function as the kernel of the likelihood function. This method is used to estimate cell values for the tables with large number of sampling zeros. In these tables, it may also happen that due to repeated sampling, some cells show more zeros than the others (density of zeros) and this is when the Dirichlet method will help us. The final stage of classification is carried out using the Naive Bayesian classification.

One of the most interesting parts of this research is the use of Naive Bayesian classifier. In the description of the classifier, Barbara et al. have used the Dirichlet distribution to obtain the probability density function for the classifier. Dirichlet distribution [6] is a good choice for this type of problem. Dirichlet distribution and Gamma distribution are time related. For example, Gamma distribution [6] will give an estimate for the time one have to wait ("waiting time in a Poisson process" [6]) before getting at least n successes. Bilodeau et al. in their book [6] proposed the following formula for the probability density function using Gamma estimation:

$$f_n(t) = \lambda e^{-\lambda t}(\lambda t)^{n-1}/(n-1)!, t > 0 \qquad (1)$$

In comparison to the Gamma distribution, Bilodeau et al. in their book [6] have described the Dirichlet distribution as "simply the proportion of time waited". *Analysis*: As time and its effects on the outcomes of any IDS is subject to a great importance in intrusion detection, addressing this issue gives a big advantage to this paper. The concept is very much into the linear algebra's subject area and needs further study. At the same time by looking at the formulas presented in either [5] or [6] reader can expect a high computation processing load for performing multiple multiplications (unless we can somehow go around this problem).

There is still one question that remains to be answered and that is: "Can one be sure that input parameters to an IDS are independent from one another?" Dependent on the answer that might be yes or no, the method of approach can be different. We are doubtful about taking the parameters as independent (or conditional independent) parameters. This is because they serve the same purpose that is intrusion. However, on the contrary it can not necessarily mean that they are not dependent either, because not all of the activities in the network are intrusions

and most of them are random legitimate activities. From our point of view, this subject deserves more study. This is so because during the design stage understanding the statistical nature of these events will help us to build the optimum model of the system.

Barbara et al. [5] in their paper, present results using two configurations: In the first configuration, given training data after Naive-Bayesian Classifier detected the intrusion, system will remove it from the DARPA 1998 training data and then will apply the classifier on the DARPA 1999 test data. In the second approach however, the DARPA 1999 training data is selected with the same test data (DARPA 1999). Then both the test and training data are introduced to the Naive Bayesian classifier and the outcome is analyzed (using the test data). The presented results are satisfactory but although they present a good research work, there is a concern with regard to the test environment. The problem arises when Barbara et al. [5] say: "To better evaluate the performance of pseudo-Bayes estimator, **we pick a set of attacks that behave very differently**, while for the attacks that share some similarities, we only select one candidates to represent the rest". In their conclusions they also talk about the problem of detecting attacks similar in nature (*Analysis*: can we translate this to: dependent input variables?). *Analysis*: The presented results confirmed our ambitions regarding the choice of assuming input parameters from the network as either independent or dependent parameters! Since a random variable version of the Bayes estimator is implemented in their work and due to the following two assumptions in this method:

1) The multinomial distribution assumption in the Bayes estimator.

2) The assumption for the Naive Bayesian is that the parameters are conditional independent.

Once the behavior of the anomalies is similar, the proposed classifier will misclassify the attacks as it is evident in the reported results. Nevertheless this paper presents a good research work in intrusion detection.

2.5.2 Fuzzy Logic Approach

As an example for the fuzzy logic based approach, Dickerson et al. [16] report a research based on the fuzzy logic concept. The paper reports a Fuzzy Intrusion Recognition Engine (FIRE) for detecting malicious intrusion activities. In the reported work, the anomaly based Intrusion Detection System (IDS) is implemented using both the fuzzy logic and the data mining techniques. The fuzzy logic part of the system is mainly responsible for both handling the large number of input parameters and dealing with the inexactness of the input data.

In the reported work, a Network Data Collection (NDC) module is implemented to take samples from the network (using TCP packet data) with 15 minutes intervals. NDC is a kind of network data sniffer and recorder system that is responsible for reading packets off the wire and storing them on the disk. The sample size is so large that authors were forced to use data mining technique to create an aggregated key composed of IP source, IP destination and destination port fields to reduce the data size. In this work, system tracks the statistical variance of the packet counts searching for any unusual increases in their number. Once any unusual increase is detected, it means that someone is scanning the network with small number of packets.

There are three fuzzy characteristics used in this work: COUNT, UNIQUENESS and VARIANCE. The implemented fuzzy inference engine uses five fuzzy sets for each data element (LOW, MEDIUM-LOW, MEDIUM, MEDIUM-HIGH and HIGH) and appropriate fuzzy rules to detect the intrusion. In their report, authors do not indicate that how did they derive their fuzzy set. The fuzzy set is a very important issue for the fuzzy inference engine and in some cases genetic algorithm approach can be implemented to select the best combination. The proposed system is tested using data collected from the local area network in the college of Engineering at Iowa State University and results are reported in this paper. The reported results are descriptive and not numerical therefore it is difficult to evaluate the performance of the reported work.

Gomez et al. [18] report a work based on the fuzzy logic concept. This work is dedicated to the network intrusion detection problem. The dataset for this work is KDD-cup'99 and 1998 DARPA datasets. In this work, the Genetic Algorithm (GA) is used to optimize the fuzzy rules so that they can better fit to the purpose. In this approach, fuzzy sets are normalized to fit within the boundary of 0.0 to 1.0.

The fitness value for the GA is calculated using the confidence weights of the system. This process is very similar to the way uncertainty problem is handled in the expert systems. Later on in their paper, a comparison has been made between the rules for the normal and abnormal behavior (there are two main sets of rules in the system, one is for the normal and the other one is for the abnormal behaviors).

In a graph presented in this paper, the false alarm rate and the detection rate of the system were input parameters and three curves for Normal rule, Abnormal rule and Normal-Abnormal rules (one with the confidence a and the other one with $1 - a$) were plotted. The graph was showing a higher detection and lower false alarm rates for using only abnormal fuzzy rules. The system was tested using only 1% of the original 1998 DARPA datasets where 10.63% false alarms and 95.47% detection rate was reported. Authors mention that this abstraction is possible since the normalization process will produce a uniform distribution. *Analysis*: Selecting the 1% ratio of the whole dataset for the training can be an indication that high computational power is required for this task.

In another reported work in this area, Botha et al. report a work [7] to detect the intrusion using the user

behavior and the fuzzy logic methodology. In this paper, intrusion detection algorithms are similar to the two earlier approaches introduced in previous papers. The overall view of the authors is to consider six different generic phases for an intrusion into a network. The goal of this system is to track and monitor the current state of the user's behavioral profile considering these categories. These six phases are:

1) Probing phase. Intruder collects information regarding the operating system, firewall and the user profile. Knowing this information will narrow intruder's options in finding the weaknesses within the system (Probing command and illegal firewall access).

2) Gaining initial access phase. This phase includes the following parameters: Invalid password attempt, user terminal (network address) and user networking hours.

3) Gaining full system access. In this phase the following activities will be encountered: Illegal password file access attempt, illegal file/directory access attempt, illegal application access.

4) Performing the hacking attempt. In this phase intruder is going to use system facilities and information (Intruder's action).

5) Covering hacking tracks. Here the intruder will erase all the track or clues leading to the exposure of his access routes and identity (Audit log access).

6) Modifying utilities to ensure future access. In this phase, the intruder will create a backdoor in the system for himself to use it for his future access (Creating user account).

In this paper, it is assumed (authors reason was the lack of data) that the model for the transition from one state to the other is linear. In other words, if anyone fails to access the system out of the regular working hours, then IDS will be 33.3% certain that this was an intrusion attempt. There is a separate membership function assigned to each one of the inputs to the system. Predefined rules together with output from the aforementioned functions are used by the fuzzy inference engine for deriving conclusions. Results reported using only 12 test subjects that looks to be a small number of test cases.

2.5.3 Data Mining Approach

In the data mining approach, Lee et al. [31] report a work based on data mining concept where initially two main usual classes of IDS are described and compared. Later, authors have explained their way of solving problems with the system and bringing it up to where it is now. Their approach is a rule-based approach (using machine learning techniques). In their proposed system, anomalies are detected using predefined rules. However, the system supervisor should know the behavior pattern for a certain anomaly in order to be able to update the system with the appropriate rules. This is how the system becomes adaptive. Authors have designed, implemented and tested several rule sets for various attack patterns. The rule generation methodology implemented in this work is interesting. They define an association rule (item set) with the following generic form: $X \rightarrow Y, c, s$ where X and Y are the item sets for the rule and $X \cap Y = \emptyset$ is the relation between them. s=$support(X \cup Y)$ where s is the support value for the rule and c=$\frac{support(X \cup Y)}{support(X)}$ is the confidence for the rule. System keeps these rules for a period of time and uses them as the pattern for the event and behavior model for the users. As an example, Lee et al. [30, 31] say:

"an association rule for the shell command history file (which is a stream of command and their arguments) of a user is: $trn \rightarrow rec.humor, 0.3, 0.1$, which indicates that 30% of the time when user invokes trn, he or she is reading the news in $rec.humor$, and reading this newsgroup accounts for 10% activities recorded in his or her command history file."

There is another rule called frequent episode rule: $X, Y \rightarrow Z, c, s, window$ where X and Y are the item sets for the rule and $X \cap Y = \emptyset$ is the relation between them. s=$support(X \cup Y \cup Z)$ where s is the support value for the rule and c=$\frac{support(X \cup Y \cup Z)}{support(X \cup Y)}$ is the confidence for the rule and $window$ is the sampling window interval.

Analysis: Their idea for tracking users sounds very interesting. As it is explained in the paper, applying proper subintervals, system will reduce the length of the user records. At the same time, system will keep the historical records for the activities in its database (data reduction). Using the user records, system will generate a rule set for the activities within the network. At this stage, system can notice the irregularities and identify them (if they are known). Several test scenarios where presented.

Since for the test purposes no standard datasets such as DARPA was used, it is hard to evaluate and compare their results. However, the proposed rule based approach is implemented in a good way.

There is an abstraction on the anomaly detection concept in their reported work. In the report [30] authors say: "Anomaly detection is about establishing the normal usage patterns from the audit data".

Their viewpoint seems to be the following: Anomaly detection is to detect any **known** anomaly (or a famous anomaly pattern) in the network. However, we are not necessarily agreed with them on the **known** anomaly or the signature based approach and would rather to use **any automatically detected intrusive** anomaly detection approach. Adaptability of their reported system requires that someone always keep the system rule sets up to date. It could be a big challenge to include an automated adaptation feature in the IDS.

Lee et al. in another paper [28] report a work to improve and continue their earlier work in the field of intru-

sion detection. In their new approach, they have implemented their system in a distributed configuration. This will help them to break down their workload and perform a type of parallel processing. In this way, they can also perform sophisticated tasks and implement complicated algorithms. Analyzing their work and considering their background in rule based approach; one can easily get the idea of the "Black Board" strategy as it is in the expert systems, out of their work.

They have also indicated that they are very much interested in "Black Box" modeling of the system. This is a great idea and honestly speaking this is the idea in our minds as well. This is because attacks are not in a static model and every now and then a novel attack pattern emerges. A black box approach to this problem will provide the IDS with the ability to detect the intrusion without necessarily knowing its type/category or name. Lee et al. [28] noted in their report that

"A major design goal of CIDF is that IDAR systems can be treated as "black boxes" that produce and consume intrusion-related information". Where CIDF and IDAR respectively stand for "Common Intrusion Detection Framework" and "Intrusion Detection Analysis and Response".

Considering the above, they have also noted in the earlier parts of their report that: "we need to first select and construct the right set of system features that may contain evidence (indicators) of normal or intrusions. In fact, feature selection/construction is the most challenging problem in building IDS, regardless the development approach in use."[28] that is a very true statement and it is important to find right features. There are some issues to bring up in this regard. These issues will be discussed in the following.

In the experiments section of the reported work, authors report an experiment where in a simulated SYN flood attack the IDS has not only detected the attack but has sent agents to the slave computers (those who where attacking the network or the server) to successfully kill the malicious agents there. The idea seems fine, but what about the legal and privacy issues? Is it legal to send agents to people's computers without their consent? There should be a legal solution to the privacy problem before implementing such strategies. This approach can be feasible for the network of an organization, but not over the internet.

Analysis: This approach seems reasonable but there are some issues that need to be addressed:

1) The reported work is heavily counting on the connectivity or the availability of the network structure for their work. In some occasions, this cannot be expected. This is because in some DOS attacks not the server but the network switches might become saturated, which means that there will be no means by which these distributed systems can communicate with one another.

2) The feature detection part has to be automated, this is because different attack strategies may have different features and in an adaptive system feature extraction has to be automated. However, authors in their implementation part of the report still report that human inspection is required in their system.

3) We believe in the Black Box (BB) approach for this type of problems. It is also evident that modeling such a huge and complicated system needs both a great computational power and a large memory space. Nevertheless, one has to accept the fact that some times cost is high! The question is not the cost but on the other hand, it is about the possibility. In many occasions, learning speed in BB modeling is so slow that it is practically impossible to use it in the real world applications. However, if possible to implement, a BB model can never tell you how or why this situation is an attack! It just knows that this is an attack (seems like one or behaves like one)!

No numerical results are presented in this report. Just the experimental environment and the experiments were described.

2.5.4 Different Tends in Data Mining Approach

In another group of the fuzzy logic and Genetic Algorithms (GA) related papers that are related to IDS concept, the one to start with is a work from Bridges et al. [8]. They report a work where fuzzy logic is used to model the uncertainties in the network behavior. The GA's role here is to optimize the membership functions of the fuzzy logic engine. Authors also report that they have implemented standard S, PI, and Z functions in their work as well. This will make the membership function to look different from just some overlapped triangles. Here, the triangles will turn into half sine waves. Their approach is an anomaly-based approach. They are using expert systems and their approach is rule-based. Association rules and their corresponding confidence and support factors are also implemented in the system. Their reported result shows that by tuning fuzzy logics membership function, GA's optimization process is improving the performance of the IDS (improves its feature extraction capabilities). In the reported paper, fuzzy results were compared versus results from a non-fuzzy system using a diagram. The depicted diagram indicates less false positive error rate for the fuzzy based methodology. The method is well defined and as it is indicated in the paper, the work is an ongoing work and needs further follow up.

In another reported work by Barbara et al. [4] reports the same work as it is previously mentioned in this report [5] and reports other researchers approaches in this area. He is not satisfied with the quality of the result reported by Lee et al. [30]. However, two papers from Lee et al. are referenced in their paper. Regarding the weaknesses of their own method, they reason that it is due to inaccurate thresholds in their classification system. Authors are suggesting that in order to improve the accuracy and

the detection rate for the proposed system, one way can be to add more sensors to the system. This idea is similar to the one in the control systems area of research (so called multi-sensor data fusion). In their future work, authors goal is to avoid the dependency on the training data (probably because it is very difficult to obtain such a dataset) for the normal events.

As it is evident in this last set of reported works, the fuzzy logic or Bayesian estimator based works can be included under either their own name or under the data mining category name. This is because the data mining work area is a multi-disciplinary area of research.

Yoshida [43] in his paper reports a new approach to the IDS design. In this paper, the author indicates that his/her goal is to provide system with the ability to detect newly encountered attacks. However, this claim in the paper is not supported by the experimental results. Yoshida's report is mainly descriptive and it talks about the new approach without showing any proof of its performance.

Yoshido explains that application of the APRIORI algorithm that mines the association rules from the given dataset is most popular among the researchers in data mining research area. Yoshido also believes that "the result of APRIORI algorithm involves association rules with contradiction"[43]. He also indicates that the result of this algorithm is noisy and in order to use it within an IDS, the result needs post-processing. As for his proof, he provides an example where in a given database there are two rules such that: Rule X has 100 supporting and 200 contracting data items in the database and rule Y which has 99 supporting and no contracting data items. Given MinSup (Minimum support) value equal to 100, APRIORI algorithm will only find the rule X. If it is desired to have the Y rule as well, then the MinSup value should be decreased which in turn will lead to a higher noise in the result. In order to improve the results, Yoshido proposes a Graph Based Induction (GBI) approach using Entropy based data mining. The GBI algorithm is as follows: First the input graph is contracted. Here "Every occurrence of the extracted sub-graph in the input graph is replaced by a single node in step 1"[43]. In the second step, the contracted graph is analyzed and consequently every subgraph consisting of two linked nodes that are called a pair is extracted. Finally satisfying a certain criteria the best pair is selected. Later on "The selected pair is expanded to the original graph, and added to the set of extracted sub-graphs"[43]. For calculation of the Entropy he uses the following formulas.

$$\begin{aligned} & Information\ gain(D,T) \\ = \ & Entropy(D) - \sum_{G_i \epsilon G} \frac{|G_i|}{|D|} Entropy(G_i) \end{aligned} \quad (2)$$

Where T is the new test dataset and D is the original dataset that is going to be classified. The Entropy can be calculated using the following formula:

$$Entropy(D) = \sum_{i=1}^{n} -p_i \log_2 p_i \quad (3)$$

The G_i is a subset of D classified by the test T and p_i is the probability of class i.

Cabrera et al. [11] report a work in continuation of their earlier work [10] where the feasibility of their approach was studied. Authors use the Simple Network Management Protocol (SNMP) to build an IDS system. They separate their approach from the common approaches in the network security area by saying:

"IDSs either rely on audit records collected from hosts (host-based IDSs) or on raw packet traffic collected from the communication medium (network-based IDSs). SNMP-based NMSs on the other hand rely on MIB variables to set traps and perform polling"[10].

Later, paper explains that although these two approaches do not have much in common, SNMP-based Network Management Systems (NMS) relying on the Management Information Base (MIB) variables can help the IDS to set traps and perform polling. This will enable us to design a distributed IDS. Authors intention is to use MIB variables to improve the detection rate of the IDS especially for those attacks that are difficult to detect. A SNMP-friendly IDS can use the MIB to cover a wide spectrum of security violations. They also believe that "MIB variables can be used not only to characterize security violations, but also to characterize precursors to security violations"[10]. Authors say that the idea of proactive IDS is about predicting the intrusion attack before it actually reaches to its final stage.

Cabrera et al. mainly focus on the Distributed Denial Of Service (DDOS) attack. In this type of attack, initially a master node will install a slave program in the target clients of the network. Then after awhile it orders them to start the attack by sending a message to them. In this system, slaves will generate an artificial traffic by which they will cause network congestion and will bring the network into halt. Cabrera et al. have characterized their proposed system into two categories:

1) **Temporal Rules:**

   In this category in the detection rule, the antecedent and the consequence will appear in a correct order in distinct time instances (first antecedent followed by the consequence). The time series analysis in this work will deal with the design of the IDS.

2) **Report incoming danger:**

   If the antecedent is true then after a certain time delay the attack will commence.

Extraction of the temporal rules is an off-line operation that implements data mining methodologies. Extraction of the rules is performed in four stages where a large dataset from the network status evolutions to the history of security violations are analyzed. These stages are as follow:

Step 1 Extracting the effective parameters/variables at the target side within the dataset. It is very important to know where to look for the clues.

Step 2 Extracting the key parameters/variables at the intruder side within the dataset. IDS should be able to model the behavior of the intruder and these variables are used to detect the current state of the intrusion process. This information may derive from statistical casualty tests on some candidate variables plus variables from step 1.

Step 3 Determining the evolution of the intrusion process using the variables derived from step 2 and comparing them versus normal state of the network. It is clear that this work follows the anomaly detection approach.

Step 4 In this stage the events extracted in the step 3 are being verified to see if they are consistently followed by the security violations observed in variables extracted in step 1.

In their description of this type of attacks, authors depict a timing diagram for a five stage transfer to the final network saturation in DDOS. These steps are: Master initiates installation of slaves (T0), Master completes installation of slaves (T1), Master commands the slave to initiate the attack (T2), Slaves start sending disabling network traffic to the target (T3), Disabling network traffic reaches the Target (T4), The target is shut down (T5).

At this time chart, T0 is the start of the attack and T5 is when the network will go down. The time-period between T1-T2 is solely dependent on human factor and on when the master will decide to order the slave to start the attack. Considering this chart and by using the NMS within the IDS, the system might be able to predict or react to the attacks.

Authors of the paper have prepared a test rig for the intrusion attack simulation and have carried out few interesting experiments on their test rig. The results are monitored and recorded. In this way, they can investigate the behavior of their IDS and study the results. Their main emphasis is on the data extracted from the MIB variables. They have included few charts from the MIB variables within the test period in their paper and have analyzed them. In intervals of 2 hours and sample rate of 5 seconds, 91 MIB variables corresponding to 5 MIB groups are collected by the NMS. These charts are synchronized so that they can be studied. Charts will provide us with an understanding of the behavior of the system during the normal and under attack periods. Since these charts are synchronized, one can easily relate the sequence of the events from one variable to the other.

Later on in their paper, authors explain how to extract rules from this dataset. In their description they have assumed that the sampling interval is constant i.e. samples are taken in equal time intervals. The result is a multivariate time series. Among different definitions in the paper, two of them seem very interesting and they are explained in below:

Causal rule "If A and B are two events, define $A \stackrel{\tau}{\Rightarrow} B$ as the rule: If A occurs, then B occurs within time τ. We say that $A \stackrel{\tau}{\Rightarrow} B$ is a causal rule"[11].

Precursor rule "If A and B are two events, define $A \stackrel{\tau}{\Leftarrow} B$ as the rule: If B occurs, then A occurred not earlier than τ time units before B. We say that $A \stackrel{\tau}{\Leftarrow} B$ is a precursor rule"[11].

Both of these rules are special cases of the temporal rules. As an indication of the certainty level for correctness of the rules, each one of these rules can be associated with a confidence factor. Authors mention that precursor rules are mined, but only causal rules were applied. Three problems for the rule extraction are addressed in this report and later on solutions have been suggested [11].

3 Modeling the Network as a System

The goal of finding a model for the network is to define the normal behavior and consequently anomaly in the behavior of the system. In the current literature, authors have defined the normal behavior of the network with regard to their own view points and no generic definitions are necessarily provided. A generic definition for the normal behavior and anomaly is proposed in below.

Generic definition of the normal behavior of the system (network): The most frequent behavior of (events within) the system during a certain time period is called the normal behavior of the system. This behavior is the dominant behavior within the system and is the most frequently repeated one.

Generic definition of the anomaly within the system (network): The least frequent behavior of (event within) the system during a certain time period is called anomaly or abnormal behavior. The repeating period for an anomaly event has a very long repeat period and its interval is close to the infinity.

The most and the least frequent events will have respectively the lowest and the highest variances among all the other events. Therefore, effective parameters will be: the duration of the time period, the frequency and the variance of the events within that time frame.

As it is clear from the literature, researchers have followed different approaches to improve accuracy and performance of their proposed IDS. However, the execution time constraint is always an obstacle or a challenge to overcome. Modeling a dynamic and complex system such as the network is very difficult. Thus, abstraction and partial modeling can be a good solution. This is why some researchers have chosen to separate different parts of the network and model them individually. The whole network can be divided into three different segments: host, user and the network environment. The user itself can

be divided into two parts: legitimate user and malicious user (intruder). Different researchers have selected either of these groups. Assigning a behavioral model to either of these groups, one can derive a model of the legitimate or anomaly behavior for them.

To model the host, it is required to monitor the system within an intrusion free working environment for a while. Using the collected data, it would be possible to derive a model for the normal behavior of the host. Any deviation from this model can be considered as an anomaly behavior and can be used for the intrusion detection. Usually there is a threshold value that determines the acceptable tolerance for any deviation from this model. Any activity that subjects the system to a deviation larger than a threshold value from its normal behavior model can be considered as an anomaly.

Another approach is to monitor the system for a period of time and then assign a baseline to the systems parameters. In this approach, crossing the baseline denotes an anomaly behavior. It is also possible to assign a normal behavior model to a host and to consider any other behavior an anomaly. However, this approach will require applying limitation on the system that might not be desirable. This approach might be suitable for cases where system performs highly repetitive tasks and within a well defined work area. It also requires deep knowledge of the system. The approach is a specification based approach to the ID and Section 4.1 will provide a more detailed discussion on the specification based ID. Sekar et al. [39] report a work in this area where a state machine based IDS is implemented to follow state transitions within the system. In this work, system is expected to behave in a certain way and IDS will respond to any abnormal state transitions.

User modeling can be an alternative method to tackle the ID problem. In this method, one can decide whether to model the legitimate user. The anomalous behavior is usually (but not necessarily) an indication of an intruder user. This approach can be used to model the intruder and to monitor his actions and his progress. Determining the normal behavior of the legitimate user can also be specification based that will lead to limiting the user within a certain boundary. Since the user is a human being and humans can be unpredictable, normal behavior modeling of the user can be a very difficult task. The same can be true with building a behavior model for the intruder. Intruder as an intelligent human being, who is aware of the usual behavior for the intruders, can slightly alter his intrusion approach and fool the system. Modeling the intruder can be a better alternative since it can be assumed that intruders are a small subset of the user community and with a known attribute (known goal) i.e. intrusion. On the contrary, legitimate users are a much larger subset of the overall users and their attribute(s) can be diverse i.e. they might have different interests and different goals.

In a reported work, Vigna et al. [41] implement a State Transition Analysis Technique (STAT) to model attack scenarios (intruder behavior modeling). Their work is especially interesting since STAT framework has an extension process that includes the extension of the attack modeling language. Therefore, using this modeling language it would be possible to model different attack scenarios.

In another reported work, Botha et al. report a work [7] to detect the intrusion using the malicious user behavior and the fuzzy logic. The overall view of the authors is to consider six different generic phases for an intrusion into a network. The goal of this system is to track and to monitor the current state of user's behavioral profile considering these categories. These six phases are respectively: probing, gaining initial access, illegal password file access attempt, performing the hacking attempt, covering hacking tracks and finally modifying utilities to ensure future access. One of the short comings of this work is an assumption that is made by the authors. They have reported that due to the lack of data, the model for the transition from one state to the other is assumed to be linear. However, in the real world this transition is nonlinear.

The network environment itself can be considered for the modeling. In this case, transactions between members of the network can be monitored. Agent based distributed systems are the main contributors to this approach. Nevertheless, this approach is very complicated and the target system is complex. In this approach the relationship between the members of a network play an important role. Distributed processing will improve the time constraints for the processing of the information within such an environment. A work reported by Lee et al. [28] is an example of this approach. In the reported work data mining techniques and a Common Intrusion Detection Framework (CIDF) are deployed to build a distributed IDS. In this work connection/section records are used as features.

4 Some Trends in IDS Design

Before getting started with describing trends in the IDS design, it should be noted that IDS has a classifier kernel. The kernel of the IDS is responsible for classifying the acquired features into two groups namely normal and anomaly, where the anomaly pattern is likely to be an attack. Nevertheless, there are occasions where a legitimate use of the network resources may lead to a positive classification result for the anomaly or signature based intrusion detection. As a result of this wrong classification, IDS will wrongly raise the alarm and will signal an attack. This is a common problem with the IDS and is called False Positive (FP). One of the parameters to measure the quality of an IDS is the number of its FP alarms. The smaller is the number of false positives, the better is the IDS.

4.1 Signature Based, Anomaly Based and Specification Based IDS

Signature based intrusion detection (misuse detection) is one of the commonly used and yet accurate methods of intrusion detection. Once a new attack is launched, the attack pattern is carefully studied and a signature is defined for it. The signature can be a name (in characters) within the body of the attack code, the targeted resources during the attack or the way these resources are targeted (attack pattern). Studying the attack pattern, security specialists can design a defense against that attack. Later on, using the proposed defense method, the IDS is updated accordingly to recognize the new attack patterns and to response to them. This approach is very efficient for the known attacks and produces small number of FP alarms. However, as the main short coming of this approach, it is not capable of detecting novel attacks. Once the attack pattern is slightly altered, this approach will not detect the altered versions of the old attacks. Thus, this approach is only efficient in detecting previously known attacks. There is another approach for detecting the novel and unseen attacks that follows.

Another widely used ID method is the anomaly detection approach [35, 34, 22, 26]. The basic idea behind this approach is to learn the usual behavioral pattern of the network. Consequently the attack is suspected (detected) once the network behaves out of its regular way (anomaly). However, network regular behavior is not similar for different networks. The network behavior is dependent on the date or the working conditions in the organization where the network is installed. The regular behavior model for the network can be variable. Considering these working conditions, the degree of freedom for the problem is large. One way to solve this problem is to make the IDS adaptable to the network environment where it is going to be installed. To do so, IDS will start to monitor and record the network behavior just after its deployment.

Assuming the recorded pattern as the regular pattern for the network, IDS will use it as the normal behavior of the network and will set a baseline. Once the network pattern deviates from this baseline pattern by more than a threshold value, it denotes an anomaly. As it was mentioned earlier, not every anomaly indicates an intrusion. This is especially true in this case, where the system is very dynamic. Thus, it is not clear if the detected anomaly should be assumed to be an intrusion or not. As a direct result of this uncertainty, anomaly based IDS will produce high FP alarms. As a remedy to this problem there should be a pruning system to detect FP alarms and cancel them. Keeping this shortcoming in mind this approach has a big benefit, that is, it is capable of detecting novel attacks or new releases of the old attacks.

One of the problems in this field of research is finding either the right features or the right relation between certain features to monitor. May be sometime in future, the anomaly detection methodology becomes mature enough not to require a baseline anymore. Currently many commercial ID systems use a hybrid approach where anomaly based intrusion detection is used together with the signature based intrusion detection method. Using the signature based ID methods system can accurately identify the known attacks with low FP alarms. If any unknown intrusions occur then anomaly detection based ID methods can detect the intrusion and raise the alarm. Using the anomaly detection based ID, the signature based methods can also be used to refine the FP alarms raised by this method. This approach will result in increasing the accuracy and reliability of the IDS while keeping the number of FP alarms low.

A recently introduced approach is the specification based intrusion detection approach. Some reported works emphasize only on the signature (misuse) based and anomaly based intrusion detection approaches [16, 12, 42, 22]. However, there are others who talk about all three of the approaches. The specification constraint in this approach is used for reducing the number of FP alarms [40, 39].

Implementation of the anomaly based IDS requires in depth knowledge of the system. The specification constraints are extracted by the human expert manually. Although specifying critical resources of the system and their utilization may improve the security, there might always be some points missing in this process that may affect the system utilization. Specification based is not just applicable to the host systems but they can also be applied on the users as well. A legitimate user is expected to behave in a certain way, or it can be specific that a user should behave in this manner. This decision will improve the security but with the expense of a less attractive user interface. Limiting the user actions and freedom may lead to making the application look less appealing to some users. It is expected to get better results by applying specification based ID methods on the system itself.

4.2 Network Based IDS and Host Based IDS

As it was mentioned earlier in the introduction section of this article, network based and host based systems are two categories of the IDSs. The network based IDS is responsible to protect the entire environment of the network from the intrusion. This task asks for full knowledge of the system status and monitoring both the components of the network and the transactions between them. Agent technology plays a key role in this strategy. Network is the infrastructure for a distributed system. Therefore, agents are a natural choice for this approach. Collecting information within the network and processing them, responding to the requests and commands of the kernel of the IDS or working as an individual, all can be accomplished using agent based technology. The network based IDS is capable of accessing the network routers and instructing them to perform tasks. Using this feature, system can ask

the router to disconnect a terminal or a subnet that has become a security threat.

There are several reported works in this area. In a paper by Foo et al. [17] authors investigate the implementation of the mobile agents in the IDS area. In this way, they intend to improve the speed of program development and update for the IDS. In another paper by Zhang et al. [46], a new architecture for a multi-agent based IDS is proposed. Paper divides the security threats into two main groups: those that come from Insider Intruders and others that come from Outsider Intruders. Paper also categorizes IDSs into three categories: host-based, network-based and router-based IDSs. Authors believe that in a multi-agent based IDS, the system should be able to have a perception of the world surrounding it. Finally, paper proposes a model with a network architecture consisting of four types of agents: basic agent, coordination agent, global coordination agent, interface agent. Console consists of two agents, a global coordination agent and an interface agent. The global coordination agent is responsible for all of the coordination agents in the system. This includes receiving reports and sending instructions to them. The interface agent can provide information for the administrator in the form of Graphical User Interface (GUI). It can also receive control commands in the form of GUI.

In a paper by Luo et al. [33] a new Mobile Agent Distributed IDS (MADIDS) has been introduced. Paper reports that one of the main goals of the system is to improve the performance of the IDS in regard to speed and network traffic. MADIDS consist of four parts: Event Generation Agent, Event Analysis Agent, Event Tracking Agent and Agent server. Data is transferred by the Generalized Intrusion Detection Object (GIDO). Event generators are responsible for collecting data and converting them to the appropriate format. Event analyzers are responsible for analyzing the events and generating GIDOs. Response units will process GIDOs. Events and GIDOs are store in event servers (databases). Distributed computing on different computers will significantly improve MADIDSs processing performance.

In a work reported by Ramachandran et al. [38] the idea of neighborhood watch is implemented for the network security. There are three different types of agents in three different layers. There are different types of Cop agents dependent on their assignments. A Cop is responsible for collecting data from various sites and reporting them to its respective detective agent. The detective agent is responsible for analyzing the reports received from the Cop agent. There is a Chief agent on the top layer who will have all the detectives reporting to him. Chief is responsible for the security of both the host and the neighbors. There might be a Chief agent monitoring a number of other Chiefs. Chief will monitor and study the reports from the detectives. If a Chief notices that security of a site has been compromised, then it will select either of the actions: Chief decides to further monitor that site or will order other sites to protect themselves from that site.

Ramachandran et al. [38] have proposed this approach with the intension to distribute the decision making and the workload of the IDS. The detective agent is responsible for analyzing the reports received from the Cop agent. There is a Chief agent on the top layer who will have all the detectives reporting to him. Chief is responsible for the security of both the host as well as the neighbors. There might be a Chief monitoring a number of other Chiefs. Chief will monitor and study the reports from the detectives. If a Chief notices that security of a site has been compromised, then it will either decide to further monitor that site or it will order other sites to protect themselves from that site. Paper reports that every now and then sites start to check each other to determine the security level of their neighbor. The result will be reported to the Detectives.

Mukkamala et al. [34] believe that IDS has two categories: host based IDS and network based IDS. They define these two types as follows: "A host based IDS monitors all the activities on a single information system host. It ensures none of the information system security policies are being violated. A network IDS monitors activities on a whole network and analyzes traffic for potential security breaches or violations."

The host based IDS is only installed on a single host/terminal and is responsible for monitoring the status of that terminal/server only. This type of IDS is responsible for the security of its host and will monitor all the network activities in that host [23]. One of the problems with the host based IDS is the high processing overhead that they impose on their host. This overhead will slowdown the host and therefore it is not welcomed. This approach is quite popular among the researchers.

4.3 Different Approaches to IDS Design

An active IDS will provide a predefined response to the detected intrusions. The passive IDS is only responsible for monitoring the system and to inform the administrator once an intrusion occurs or to produce an advance warning. The response concept is related to the active IDS. This response can be a reaction to a security breach in the system or a preemptive response to avoid a security breach. One of the main goals of any active IDS is to prevent the security breach and not just to respond to the threat.

Continuing their earlier work [10], Cabrera et al. [11] report a more advanced work where the feasibility of their approach is studied. Authors use the Simple Network Management Protocol (SNMP) to build an IDS system. In this report, authors report that the idea of proactive IDS is about predicting the intrusion attack before it actually reaches to its final stage. Proactive IDS is a system that reacts to the imposed threats, and in response it will apply predefined defensive routines within the system.

There are two types of product lines in the commercial IDS industry. In one type of production, the IDS is

produced in the form of a software package. In order to protect a host, the IDS software has to be installed on that host. Once the IDS is installed, it will access the network modules/ports of that host and will gain control over them. Later on, using its control over the system, IDS will monitor the network transactions and will respond to the threats. Although it can be used for the distributed IDS as well, this approach is more suitable for the host based IDS than the network based IDS.

However, industry has shown a great interest in another approach as well. In this approach, the whole IDS product is included in one box (IDS appliance). Both the hardware and the software modules are inside that box. Other hosts/servers can communicate with the IDS using the network infrastructure. Network administrator can update the IDS with new policies using a terminal with a network connection. Products from companies such as CISCO (CISCI IDS Sensors) and Mazu (Mazu Enforcer) are examples of this type of approach.

The reason for the IDS appliance approach being attractive to the market and consequently to the IDS industry, is its ease of installation and flexible deployment. At the same time, administrators do not need to worry about the high computing overhead exerted on the host machines by the IDS. Once a network is targeted, the first attack is aimed on the IDS itself. Thus, in the case of the host based IDS, both the host and the IDS will go under attack. This situation will increase the computing overhead on the host machine at the same time will reduce its response time. With the appliance IDS this problem is solved. Another benefit of this approach is for the manufacturer. Producing the IDS in the form of an appliance will improve the security measures for the product reengineering as well. It is easy to crack a software and make illegal copies out of it, however, following this approach wont be feasible for the appliance IDS. The hardware implementation of the appliance will make it harder and more expensive to make a copy of it. The drawback for this approach is the cost of production. Using the hardware components will increase the production cost and consequently the price of the product. Another benefit of this approach is the guarantee for the optimum hardware setup and performance for the IDS. This is because, the hardware platform and the software setup is already completed and tested by the manufacturer that is familiar with the system.

5 Where to Look for the Features?

Desired features for the IDS depend on both the methodology and the modeling approach used in building the IDS. These features are usually numerous. Thus considering the volume of data, processing all of them will take quiet awhile. In order to speed-up the process, these features are usually preprocessed to reduce their size, while increasing their information value. There are numerous approaches reported in this area. Most of the reported research is concerned with the header of the packets. However, recently researchers have valued the body or the payload of the packets as well. This part of the packets was usually disregarded due to its large volume and the extensive processing time required for processing them. Researchers such as Lee et al. [30, 31, 28] with data mining interests tackle the problem using the association rules. However, they extract these rules using the information in the header of the packets. Zanero et al. [44] report a work where TCP/IP header and packets payload are used to extract features. In this work, using an unsupervised clustering algorithm, the payload is compressed into a single byte. In their work, Zanero et al. have used SOM for the classification of packets. In another reported work, Lei et al. [32] report Improved Competitive Learning Network (ICLN) method based on the SOM but 75% faster (experimented on the KDD-99 dataset).

Just extracting features is not useful for the ID. Extraction should be followed with a second stage where patterns are produced using the extracted features. Using these patterns, intelligent kernel of the IDS will analyze the working condition of the network and will raise the alarm if necessary. Employing pattern extraction method, one should consider both the importance of the features and the relation between them in the feature space. The resultant pattern is somehow a compressed and abstracted version of the feature space. One of essential questions is how to determine the information value of the features or how to find the relations and evaluate the importance of the relations between the features? Usually statistical methods [6] are used for this purpose. Data mining methods are also commonly used in this area [30, 31, 28, 29]. These approaches are not limited to using the association rule approach but other methods such as ANN [44, 32, 23, 20], Bayesian [5], Fuzzy Logic [8], Genetic Algorithms (GA) [8, 35, 22], HMM [12, 3, 42] and SVM [23, 20] methods have attracted much attention as well.

6 Honey Pot (HP)

Despite its effectiveness, not until recent years this approach has been taken seriously within the academia. However, recently research in this area has gained some momentum. Maybe this lag was due to the fact that there is no theoretical concept involved in the HP approach, it is just a deception. HP is mainly a heuristic approach and is based on the concept of bait and trap. Nevertheless, industry sector is very attracted to this concept. There are a number of products available that use the HP to trap undetected intrusion attempts. Generally speaking, HP is a deception based approach to detect actions of a deceitful enemy (the intruder). The HP concept has attracted much attention over the internet and there are numerous sites dedicated to this concept [19]. Some HP based frameworks include:

- Honeyd project with GNU General Public License [21] that creates virtual hosts on a network.

- Honeynet [36] project, is a project defined with the goal of building a virtual Honey net.
- Specter [13] is a commercial product and supports HP for different resources within the system. One of the important features for this commercial product is the wide spectrum of operating systems that it covers.
- Project Honeypot [1] is aimed to protect websites and email servers from the spammers.
- Back Office Friendly (BOF) [14] is a free software that deploys HP for various services in the system e.g. SMTP, POP3, HTTP, FTP, Telnet and IMAP2.
- Many other research projects in this area are listed in Honeypots.net site [19] that are out of the scope of this report.

The overall idea behind the HP technology is to lay a trap and bait and wait for the hunt to fall into it. This is in addition to all the other detection tools deployed to catch the hunt. Here HP is used as a supplement to the IDS to detect the intrusion where IDS was unable to do so. In this way, the probability of missing some attacks undetected is reduced. Other than detecting attacks missed by the IDS, HP can help the IDS in many other ways.

The main assumption in implementing a HP, is that, no one would ever use the selected resource for the HP. Therefore, any attempt to use those resources is expected to be malicious and should be monitored. Some benefits of the HP are listed in below:

- HP will keep the attacker preoccupied. In this method, after detecting the intrusion attempt, system will alert the security officer immediately while keeping the attacker busy. This approach has several benefits that are listed in below:
  1) By selecting correct resources for the HP to emulate as bait, one can distinguish different interests of the attackers. Having the intruder fallen in the trap, HP can start studying the opponent. HP will let the intruder to navigate through the emulated environment and will log its actions. HP or the security officer can gain much information by studying logged actions of the attacker, the targeted resources and attackers information regarding the system such as username, password, etc.
  2) Keeping the attacker busy. HP will buy more time for the response system or the security officer/administrator of the system to come-up with a proper response to the attack attempt. Response latency time is very important since attacks might be too fast for the response system to react to them. At the same time, providing the system administrator with sufficient time might help in finding the root of the connection that the attacker is using. Tracking the attackers can be very difficult and it requires a great deal of experience and time to trace them back to their origin.
  3) Another benefit gained by HP keeping the attacker busy is to waste attackers time and to prevent him compromising other resources with a fast speed. For example, in a port scanning scenario keeping the attacker waiting for a reply may significantly slowdown the whole process. HP can do this in a different way as well, that is, by emulating a working environment, HP can convince the intruder that he is in a real system and let him try to perform the intrusion. The more convincing is the HP, the more time the intruder will spend in the environment.
- Within an IDS guarded system, HP will detect unnoticed attacks. In this way, HP will increase the reliability of the IDS. The detection offered by the HP is independent of the type of attack that is enforced on the system.
- Another benefit for using a HP in a system is with respect to the overhead on the processor. In the IDS approach every packet transaction in the system has to be monitored and analyzed. Thus, the IDS approach will generate a high computational power demand within the system. The same is true with the data transfer within the system. Due to the size of the packets transferred within the system, data transfer time will consume a large part of the processing time. In addition to the processing time, data transfer will consume different data transfer related resources of the system. Using the HP will reduce the processing and resource consumption overhead on the system. The processing time will be provided for the HP process only when the HP monitored resources within the system are utilized. In this way, the overhead enforced on the host system by the HP during the HPs idle period will be negligible. This scenario is very similar to the differences between the interrupt based and the polling based IO handling methods in the computer hardware design.

In order to improve the performance of the HP operation and to increase the hit probability of the HP, usually a large number of HPs are placed in a network. These HPs are close to the important resources in the network and operate both as a guardian and a decoy for them. A group of HPs that are distributed in a system is called a Honey Pot farm.

Nowadays, intruders aware of the HP technology try to avoid the HPs or even take advantage of them. To do so, they have implemented tools to detect the HP and once detected, they disengage the HP. At the same time, HPs are usually deployed to protect a resource or a data. Once

the HP is detected, it is most probable to find a valuable resource in its neighborhood.

Zhang et al. [45] report an investigation on the HP concept. In the reported paper, HP concept is investigated and HP and honey nets are described. Data control and capture for the HP are illustrated. Authors categorize the HPs into four categories: Prevention, Detection, Reaction and Research. By the prevention, Zhang et al. mean that the prevention HP will delay the intrusion by diverting intruders attention to the HP. As for the detection HP, it can generate alert once the attack is detected. Authors also believe that HP can not be used individually and it is a supplement to the IDS. They define the reaction HP to be a type of companion system and trial environment for test systems vulnerabilities. The research honey pot is to log and study the opponent (the intruder) and to report the result. This type of HP will act in a very flexible way and will provide the intruder with a vast maneuver space. The main goal of this type of HP is to determine the purpose and the goal of the intruder. A HP can hold any or a combination of these categories.

Kuwatly et al. [27] in their paper, divide HPs into two categories i.e. Low-interaction HPs and High-interaction HPs. Low-interaction HP are those that have limited interaction capabilities and can emulate certain protocols such as FTP e.g. Specter [13], Honeyd [21].

As the HP technology improves so does the anti-HP technology. Therefore, a never ending battle is already started. Honey pots have to be improved constantly otherwise they themselves will become a weak point in the system. In other words, protecting or hiding something shows that it is important to us. Thus, by knowing what resources HP is protecting, the intruder can identify the environment and the valuables in the system in a better and more efficient way [25]. Authors propose a design for a dynamic honey pot, capable of changing configuration to match the dynamic and ever changing environment of a network.

Khattab et al. [24] propose roaming HPs for service-level DoS attacks (physical roaming). The proposed mechanism allows the HP to randomly move its position within a server pool. Interesting beneficial features in this work are the filtering effect and connection-dropping. The filtering effect is when the idle server that is acting as a HP detects addresses of the attackers and filters them out or blacklists them. The connection-dropping occurs at a random displacement time when the server is switching from idle (HP mode) to active. At this time server drops all the connections (attacker connections). This connection dropping in turn will open space for the legitimate requests before a new wave of attacks start again. In their paper, authors define the logically roaming honey pots in the following way.

Logically roaming honey pots are similar to the IP hopping, where legitimate clients coordinated by the servers randomly change the destination address in their packets. In this way, the unwelcome traffic that is not updated with the correct destination address will be rendered. They also claim that although logical roaming of the HPs is more cost effective, the physical roaming is still necessary to protect the network against the internal attacks.

7 Conclusions

Considering the surveyed literature, it is clear that in order to be able to secure a network against the novel attacks, the anomaly based intrusion detection is the best way out. However, due to its immaturity there are still problems with respect to its reliability. These problems will lead to high false positives in any anomaly-based IDS. In order to solve this problem, usually a hybrid approach is used. In the hybrid approach, the signature-based approach is used together with the anomaly-based approach. In this way, the second approach is mostly used for the novel tactics while the accuracy of the first approach (signature based approach) will provide a reliable detection for the known attacks. Specification-based approach is only good when system specifications and details are known and applying limitations on the user is acceptable. The generic definition of the normal behavior and the anomaly behavior in the system are presented in this paper. The intension for introducing these generic definitions was to help researchers to converge on the definition of the normal behavior of the network.

In network-based IDS, agent based systems play an essential role. In such systems a distributed processing architecture is a must and system has to collect information from different components within the network. Implementing such architecture, one should avoid increasing the network traffic.

Large volume of data and non-deterministic normal behavior of the network are two major challenges in IDS design. As the volume of data using the header of the packets is already very large, using information in the payload will make the process even slower. However, there are works reported by some researchers in this area that show good progress in using packets payload for the analysis.

The intrusion detection products were analyzed with respect to the software or appliance based production and the benefits of either of the designs were discussed. Building hardware appliances can be more difficult for companies with lower development budget. However, appliance based IDSs are more appreciated in the market. From the consumer point of view, appliance based IDS is easier to install and to maintain. In manufacturers view, appliance based IDS is a more secure design to manufacture but as the same time more expensive to produce.

Another aspect of the IDS design is the issue of the missed attacks. If some attacks are not detected by the IDS, there are no means to notice them. This is especially the case with the novel attacks. In addition to all other benefits, HP technology can help to expose these attacks. The accuracy of the HP technology depends on the number of HPs distributed in the system (population of the HP farm). The larger the population of the HPs the more

accurate is the detection rate. Increasing the accuracy is not the only benefit for implementing the HP technology, but it can be used for other purposes as well. For example, HP can be used for studying or slowing down the intruder.

8 Future Work

As for the future work, intension is to produce an IDS capable of anomaly and signature based intrusion detection. There are two options in front of us, i.e. host based or network based IDS. The host based IDS can be easier to implement, though the network based IDS needs more time and effort for its implementation and design. In return, the network based IDS will provide a more reliable and more accurate IDS. The network IDS needs to have environment awareness. Thus, the network based IDS needs special sensors for its work. Agent based technology is one of the essential blocks in this distributed architecture design methodology.

The selected approach for our future work is the network based software product. However, the host based approach will be considered as well. The project timeframe and the budget are main issues with regard to this decision. Nevertheless, accepting the expenses, it is always possible to convert a software based IDS to the appliance version of it.

From the theoretical point of view, it is intended to improve the accuracy of the anomaly based intrusion detection. One way to do so is to use the payload of the packets. Therefore, it is necessary to envisage a method either to reduce the size of the data or to process the data more quickly. The main idea is to find a method to handle high volume of data with less information loss. For the same reason, features should be evaluated with respect to their information value. In this way, every feature will be associated with a coefficient of importance that determines its overall effectiveness in comparison to the other features. Efficient algorithms and programs can provide a great help for this purpose.

Acknowledgement

This work was funded by the Atlantic Canada Opportunity Agency (ACOA) through the Atlantic Innovation Fund (AIF) to Dr. Ali A. Ghorbani.

References

[1] Unspam; LLC a Chicago-based anti-spam company. "Website for the project honeypot,". http://www.projecthoneypot.org/.

[2] M. Analoui, A. Mirzaei, and P. Kabiri, "Intrusion detection using multivariate analysis of variance algorithm," in *Third International Conference on Systems, Signals & Devices SSD05*, vol. 3, Sousse, Tunisia, Mar. 2005. IEEE.

[3] A. Zhong and C. F. Jia, "Study on the applications of hidden markov models to computer intrusion detection," in *Proceedings of Fifth World Congress on Intelligent Control and Automation WCICA*, vol. 5, pp. 4352–4356. IEEE, June 2004.

[4] D. Barbara, J. Couto, S. Jajodia, and N. Wu, "Special section on data mining for intrusion detection and threat analysis: Adam: a testbed for exploring the use of data mining in intrusion detection," *ACM SIGMOD Record*, vol. 30, pp. 15–24, Dec. 2001.

[5] D. Barbara, N. Wu, and S. Jajodia, "Detecting novel network intrusions using bayes estimators," in *Proceedings of the First SIAM International Conference on Data Mining (SDM 2001)*, Chicago, USA, Apr. 2001.

[6] M. Bilodeau and D. Brenner, *Theory of multivariate statistics*. Springer - Verlag : New York, 1999. Electronic edition at ebrary, Inc.

[7] M. Botha and R. von Solms, "Utilising fuzzy logic and trend analysis for effective intrusion detection," *Computers & Security*, vol. 22, no. 5, pp. 423–434, 2003.

[8] Susan M. Bridges and M. Vaughn Rayford, "Fuzzy data mining and genetic algorithms applied to intrusion detection," in *Proceedings of the Twenty-third National Information Systems Security Conference*. National Institute of Standards and Technology, Oct. 2000.

[9] D. Bulatovic and D. Velasevic, "A distributed intrusion detection system based on bayesian alarm networks," *Lecture Notes in Computer Science (Secure Networking CQRE (Secure) 1999)*, vol. 1740, pp. 219–228, 1999.

[10] J. Cabrera, L. Lewis, X. Qin, W. Lee, R. Prasanth, B. Ravichandran, and R. Mehra, "Proactive detection of distributed denial of service attacks using mib traffic variables - a feasibility study," in *Proceedings of the 7th IFIP/IEEE International Symposium on Integrated Network Management*, pp. 609–622, Seattle, WA, May 2001.

[11] Joao B. D. Cabrera, L. Lewis, X. Qin, W. Lee, and Raman K. Mehra, "Proactive intrusion detection and distributed denial of service attacksa case study in security management," *Journal of Network and Systems Management*, vol. 10, pp. 225–254, 2002.

[12] S. B. Cho, "Incorporating soft computing techniques into a probabilistic intrusion detection system," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART C: APPLICATIONS AND REVIEWS*, vol. 32, pp. 154–160, May 2002.

[13] NETSEC-Network Security Software Co. "Specter,". http://www.specter.com/.

[14] NFR Co. "Website of nfr co.,". http://www.nfr.net/.

[15] Mitsubushi Corporation. "Concordia mobile agent development kit,". Software, 1999.

[16] John E. Dickerson and Julie A. Dickerson, "Fuzzy network profiling for intrusion detection," in *Proceedings of NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society*, pp. 301–306, Atlanta, USA, July 2000.

[17] Simon Y. Foo and M. Arradondo, "Mobile agents for computer intrusion detection," in *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory*, pp. 517–521. IEEE, IEEE, 2004.

[18] J. Gomez and D. Dasgupta, "Evolving fuzzy classifiers for intrusion detection," in *Proceedings of the 2002 IEEE Workshop on the Information Assurance*, West Point, NY, USA, June 2001.

[19] honeypots.net. "Website for honeypot,". http://www.honeypots.net/.

[20] P. Z. Hu and Malcolm I. Heywood, "Predicting intrusions with local linear model," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 1780–1785. IEEE, IEEE, July 2003.

[21] Website is maintained by: Niels Provos. "Honeyd framework,". http://www.honeyd.org/.

[22] J. Guan, D. X. Liu, and B. G. Cui, "An induction learning approach for building intrusion detection models using genetic algorithms," in *Proceedings of Fifth World Congress on Intelligent Control and Automation WCICA*, vol. 5, pp. 4339–4342. IEEE, June 2004.

[23] H. Gunes Kayacik, A. Nur Zincir-Heywood, and Malcolm I. Heywood, "On the capability of an som based intrusion detection system," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 1808–1813. IEEE, IEEE, July 2003.

[24] Sherif M. Khattab, C. Sangpachatanaruk, D. Mosse, R. Melhem, and T. Znati, "Roaming honeypots for mitigating service-level denial-of-service attacks," in *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS04)*, pp. 328–337. IEEE, IEEE Computer Society, Mar. 2004.

[25] N. Krawetz, "Anti-honeypot technology," *IEEE SECURITY & PRIVACY*, vol. 2, pp. 76–79, Jan.-Feb. 2004.

[26] C. Kruegel, T. Toth, and E. Kirda, "Service specific anomaly detection for network intrusion detection," in *Proceedings of the 2002 ACM symposium on Applied computing*, pp. 201–208. ACM, Symposium on Applied Computing, ACM Press New York, NY, USA, Mar. 2002.

[27] I. Kuwatly, M. Sraj, Z. Al Masri, and H. Artail, "A dynamic honeypot design for intrusion detection," in *Proceedings of the IEEE/ACS International Conference on Pervasive Services (ICPS04)*, pp. 95–104. IEEE, IEEE Computer Society, July 2004.

[28] W. Lee, Rahul A. Nimbalkar, Kam K. Yee, Sunil B. Patil, Pragneshkumar H. Desai, Thuan T. Tran, and Salvatore J. Stolfo, "A data mining and cidf based approach for detecting novel and distributed intrusions," in *Proceedings of Recent Advances in Intrusion Detection, 3rd International Symposium, (RAID 2000)* (H. Debar, L. M., and S.F. Wu, eds.), pp. 49–65, Toulouse, France, October 2000. Lecture Notes in Computer Science, Springer-Verlag Heidelberg.

[29] W. Lee and Salvatore J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, pp. 227–261, Nov. 2000.

[30] W. Lee, Salvatore J. Stolfo, and Kui W. Mok, "Mining audit data to build intrusion detection models," in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD '98),*, New York, NY, USA, Aug. 1998.

[31] W. Lee, Salvatore J. Stolfo, and Kui W Mok, "Adaptive intrusion detection: A data mining approach," *Artificial Inteligence Review*, vol. 14, no. 6, pp. 533–567, 2000.

[32] J. Z. Lei and Ali Ghorbani, "Network intrusion detection using an improved competitive learning neural network," in *Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR04)*, pp. 190–197. IEEE-Computer Society, IEEE, May 2004.

[33] G. Luo, X. L. Lu, J. Li, and J. Zhang, "Madids: A novel distributed ids based on mobile agent," *ACM SIGOPS Operating Systems Review*, vol. 37, pp. 46–53, Jan. 2003.

[34] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in *International Joint Conference on Neural Networks IJCNN02*, vol. 2, pp. 1702–1707, Honolulu, HI USA, May 2002. IEEE, IEEE. Source: IEEE Xplore.

[35] F. Neri, "Comparing local search with respect to genetic evolution to detect intrusions in computer networks," in *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 238–243, Marseille, France, July 2000. IEEE, IEEE. Source: IEEE Xplore.

[36] Website of the Honeynet Project. "Honeynet project,". http://www.honeynet.org/.

[37] M. Otey, S. Parthasarathy, A. Ghoting, G. Li, S. Narravula, and D. Panda, "Towards nic based intrusion detection," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 723–728. ACM, ACM Press, NY, USA, Aug. 2003. Poster Session: Industrial/government track.

[38] G. Ramachandran and D. Hart, "A p2p intrusion detection system based on mobile agents," in *Proceedings of the 42nd annual Southeast regional conference*, pp. 185–190. ACM Press New York, NY, USA, Apr. 2004.

[39] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: a new approach for detecting

network intrusions," in *Proceedings of the 9th ACM conference on Computer and communication security*, pp. 265–274, Washington D.C., USA, Nov. 2002. ACM Press.

[40] T. Song, J. Alves-Foss, C. Ko, C. Zhang, and K. Levitt, "Using acl2 to verify security properties of specification-based intrusion detection systems," in *Fourth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2-2003)*, July 2003.

[41] G. Vigna, F. Valeur, and Richard A. Kemmerer, "Designing and implementing a family of intrusion detection systems," in *Proceedings of the 9th European software engineering conference held jointly with 10th ACM SIGSOFT international symposium on Foundations of software engineering*, pp. 88–97, Helsinki, Finland, 2003. Source: ACM Portal.

[42] N. Ye, "A markov chain model of temporal behavior for anomaly detection," in *Proceedings of the 2000 IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY, June 2000.

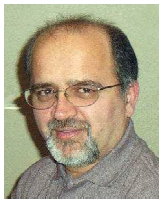
[43] Ken. Yoshida, "Entropy based intrusion detection," in *Proceedings of IEEE Pacific Rim Conference on Communications, Computers and signal Processing (PACRIM2003)*, vol. 2, pp. 840–843. IEEE, Aug. 2003. IEEE Explore.

[44] Ste. Zanero and Sergio M. Savaresi, "Unsupervised learning techniques for an intrusion detection system," in *Proceedings of the 2004 ACM symposium on Applied computing*, pp. 412–419, Nicosia, Cyprus, Mar. 2004. ACM Press.

[45] F. Zhang, S. Zhou, Z. Qin, and J. Liu, "Honeypot: a supplemented active defense system for network security," in *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT2003)*, pp. 231–235. IEEE, IEEE, Aug. 2003.

[46] R. Zhang, D. Qian, C. Ba, W. Wu, and X. Guo, "Multi-agent based intrusion detection architecture," in *Proceedings of 2001 IEEE International Conference on Computer Networks and Mobile Computing*, pp. 494–501, Oct. 2001.

Peyman Kabiri received his PhD in Computing (Robotics and Machine Learning) and MSc in Real time Systems from the Nottingham Trent University, Nottingham-UK respectively in years 2000 and 1996. He received his B.Eng. in Computer Hardware Engineering from Irans University of Science and Technology, Tehran-Iran in 1992. He is currently with the Faculty of Computer Science/ University of New Brunswick as project coordinator since September 2004. His previous academic positions were as follows: Assistant professor in Department of Computer Engineering Irans University of Science and Technology and Assistant Professor in Azad University central branch Faculty of Engineering both in Tehran. He was a reviewer in several conferences. His research interests include Machine Learning, Robotics and Network Intrusion Detection.



Ali A. Ghorbani (M95) received his PhD (1995) and Masters (1979) from the University of New Brunswick, and the George Washington University, Washington D.C., USA, respectively. He was on the faculty of the Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran, from 1987 to 1998. Since 1999 he has been at the faculty of Computer Science, University of New Brunswick (UNB), Fredericton, Canada, where he is currently a Professor of Computer Science. He is also a member of the Privacy, Security and Trust (PST) team at the National Research Council (NRC) of Canada.

He has held a variety of positions in academia for the past 24 years. His research originated in software development, where he designed and developed a number of large-scale systems. His current research focus is Neural Networks, Web intelligence, agent systems, complex adaptive systems, and trust and security. He established the Intelligent and Adaptive Systems (IAS) research group in 2002 at the faculty of Computer Science, UNB. The IAS group (http://ias.cs.unb.ca) pursues research on machine and statistical learning, data mining, intelligent agents and multiagent systems. The group is also home to R&D in Web intelligence, network security and application of multiagent systems to e-Health.

He authored more than 90 research papers in journals and conference proceedings and has edited four volumes. He is on the editorial board of the Computational Intelligence (CI), an international journal. He is also associate editor of the International Journal of Information Technology and Web Engineering.

Dr. Ghorbani a member of ACM, IEEE Computer Society, and ANNS.