Towards Cyber Defense: Research in Intrusion Detection and Intrusion Prevention Systems

Mohammad A. Faysel †, and Syed S. Haque †

†Department of Health Informatics, School of Health Related Professions
University of Medicine and Dentistry of New Jersey ,65 Bergen Street, Room 350, Newark, NJ 07107, U.S.A.

Summary
Cyber attack is one of the most rapidly growing threats to the world of cutting edge information technology. As new tools and techniques are emerging everyday to make information accessible over the Internet, so is their vulnerabilities. Cyber defense is inevitable in order to ensure reliable and secure communication and transmission of information. Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are the major technologies dominating in the area of cyber defense. Tremendous efforts have already been put in intrusion detection research for decades but intrusion prevention research is still in its infancy. This paper provides a comprehensive review of the current research in both Intrusion Detection Systems and recently emerged Intrusion Prevention Systems. Limitations of current research works in both fields are also discussed in conclusion.

Keywords:
cyber defenses, cyber security, intrusion detection system, intrusion prevention system, network security survey.

1. Introduction

Besides perimeter firewalls, Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are the major techniques widely used by organizations in various fields to defend cyber attacks. IDSs basically use system generated retrospective log files and apply specific detection techniques to determine attacks. On the other hand, IPSs not only detect attack using similar techniques as IDSs, they have the capabilities to take specific responses as well. Intrusion detection has been an active research field for decades and tremendous achievements have been accomplished by researchers in detecting intrusive incidents by applying various techniques. In response to rapid growth of cyber attacks, intrusion detection system by itself is not adequate but taking appropriate response at the same time have proven to be promising in defending cyber threats. As a result, there have been a new research field emerged recently, intrusion prevention system, which is still in its early infancy.

IPS research took its root from IDS research and some researchers define IPSs as combination of IDSs with added functionalities [1]. Although, research in IDSs and IPSs are very important in order to address cyber defense, very few survey papers include current research works in both fields together. In order to address this issue, this paper provides a comprehensive review of current research works in both of the areas. Based on detection techniques, each of the research works and its methodology used are carefully reviewed. Towards the end, this paper also discusses the limitations of current research in both IDS and IPS fields. Although, there are a number of research works carried out on both host based and network based IDSs, the scope this review paper is limited to the network based systems.

2. Research in Intrusion Detection System (IDS)

Intrusion Detection System has been an active research field for at least past 30 years. In a survey of nearly 15 years of IDSs research, Stefan Axelsson [2] describes and compares the early IDSs (up until 90's). Early IDSs mostly focused on detection of misuse based on user activities. Their detection capabilities were based on logs of UNIX system commands invoked by individual users.

IDSs are categorized into two categories based on detection techniques they use: misuse detection and anomaly detection. We will explore IDS research works involved in these two categories based on the methodologies they use.

2.1 Misuse Detection-based IDSs

Misuse detection technique is the most widespread approach used in the commercial world of IDSs. The basic idea is to use the knowledge of known attack patterns and

Manuscript received July 5, 2010
Manuscript revised July 20, 2010

apply this knowledge to identify attacks in various sources of data being monitored. Therefore, misuse detection based IDSs attempt to detect only known attacks based on predefined attack characteristics [3]. An attack may take place in different patterns and the accuracy of such IDSs solely depends on how well the knowledge of attack information is preprocessed and fed to the IDSs' detection engine. Well "crafted" expert knowledge of known attacks can enable misuse detection based IDSs to perform more accurately with a low false positives.

2.1.1 Signature based approach

Signature based approach of misuse detection works just similar to the existing anti-virus software. In this approach the semantic characteristics of an attack is analyzed and details is used to form attack signatures [4]. The attack signatures are formed in such a way that they can be searched using information in audit data logs produced by computer systems. A database of attack signatures is built based on well defined known attacks and the detection engine of an IDS compares string log data or audit data against the database to detect attack. Each time a new attack is discovered, the attack signature database has to be quickly updated accordingly for more up-to-date result and accuracy. There are various signature matching algorithms used in various signature based cyber attack detection systems.

Snort [5] is the most popular signature based lightweight network IDS which is available as open source. Snort can be configured in any of the three modes: packet sniffing mode which enables it to monitor and to display network traffic packets; network traffic logger mode in which Snort writes the network traffic log into a file; and IDS mode in which it has both intrusion detection and prevention capabilities in real time based on user defined known attack signatures. It uses a database consists of user defined attack signature rules and uses Boyer-Moore pattern matching algorithm against the database for each network traffic packet [5]. Unlike other signature based IDSs, Snort analyzes application layer of network traffic to detect specific pattern of well known attacks such as buffer overflow, port scan etc. [5]. When possible match is found, snort can alert the proper user, record the network packet information, and can take user defined actions such as dropping the packet etc. Although there are many open source products that can be combined with Snort to expand sophisticated intrusion prevention capabilities, its detection capability is limited to the attack signature rules provided in the database. Therefore, it is incapable of detecting novel or new attacks [6].

Haystack [7] was designed as one of the earliest signature based IDS mainly for monitoring multi-user system in the Air Force computing facility [2]. Primary prototype of the system was basically designed to detect only six specific types of attacks namely attempted break-ins, masquerade attacks, unauthorized penetration to the security control system, sensitive data leakage, denial of service attack and suspicious use of the system [2][7]. The security policies were converted into rules and stored in a database and each new session audit data was compared against the database to detect violation of predefined rules for misuse detection. Along with signature based misuse detection, this system also used an anomaly detection technique, which was based on behavior profile of each user's past actions and acceptable behavior of specific user group. However, it was a user profile based non-real time IDS with limited capabilities [2].

Network Flight Recorder (NFR) [8] is a commercially used powerful network based intrusion detection and analysis tool. It uses various signatures of known attacks to raise alarm in case any attack is detected. It differs from Snort in the way that NFR is more complete network monitoring and analysis system than Snort [5]. NFR uses its own scripting language called n-code for generating signatures and network packet analysis.

Bro [9], an open source UNIX based network intrusion detection system, uses a signature based approach [10]. It detects technique is based on detecting known attacks and events based on pre defined attack signatures and events and detects uncommon activities (failed connection attempts).

Bro system is composed of three distinct layers namely libpcap [11]- the packet capture library for packet filtering, event engine- which deals with the already filtered network packets and a policy script interpreter- which handles events generated by event engine [9]. Bro uses special Bro scripting language to design specific event handler which can take an action such as generating real-time alert message, logging the entry based on an event generated by the event engine. The Bro system is capable of preventing intrusive attack by taking proper defensive action such as blocking the attacking computer host or by terminating the TCP connection when instructed [9][12].

Abstraction based misuse detection system designed by Lin et al. [23] called, ARMD (Adaptable Real-time Misuse Detection System), used its own high level language called MuSigs for misuse signatures abstraction from audit log in the UNIX based environment. Using MuSigs misuses were represented into easily understandable simple abstract forms of signatures [13]. The system had the capabilities to interpret these abstract signatures to the detection mechanism. ARMD used a monitoring algorithm that checked for matching signature for a specific view in the given event history and if there was any match found, it would report it. Using ARMD part of the monitored system, real-time misuse could be detected while other part of the system could be monitored off-line [13].

Commercial vendors claim to incorporate both signature and anomaly based capabilities in their commercial IDSs but most of their systems are solely rely on signature based techniques in practice [10]. Kruegel et al. [10] applied decision tree approach for matching attack signatures instead of traditional signature matching technique such as the one used in Snort [5] and achieved improved detection speed.

Signature based approach is easy to implement efficiently and very popular in the commercial world [4]. They can operate with a high level of accuracy in detecting known attacks in real-time. However, signature based approach is incapable of detecting previously unknown or novel attacks. Moreover, signature database requires manual update of new type of attack discovered and human expert has to perform such task, which is time consuming. Therefore, there is a huge delay in the discovery of a new attack and development of the attack signature and propagating the signature in the attack signature database [14].

2.1.2 Rule based approach

Most of the widely used misuse detection systems use rule-based approach [3]. Such systems are built on a number of conditional if-then rules for their detection techniques. Rules are developed by analyzing attacks or misuses by experts and then transforming them into conditional rules which are later used by inference modules of IDSs to compare against monitoring data (usually logs) to detect any misuse.

A real time Intrusion-Detection Expert System (IDES) [15] [16] [17] is one of the classic rule-based misuse detection systems. The system uses multivariate methods to calculate summary statistics of the characteristics of user behavior from audit logs of user activities. Using these statistics, the system develops a profile of normal behavior for each category of user group based on user privileges for each group composed of users with same level of privilege [18]. The system then uses a statistical sub-system to monitor and compare user behavior against past behavior of that user and also against a rule based expert system composed of the expected normal behavior of the user group the user belongs to. If the monitored user activity deviates from expected behavior at a significant level, that is, if the user activity violates any rule in the expert system, then the user behavior will be considered as intrusion [17]. IDES had the capability to detect misuse or attack by authorized users who abuse their given privileges.

Production-based expert system toolset, known as P-BEST [19], is a rule-based expert system that is consisted of a rule interpreter and a set of various routines. User specified facts and rules are interpreted by the rule interpreter into a forward chaining expert system in P-BEST. P-BEST rule-based expert system is used for computer and network misuse detection [19]. Each time a new fact is added, the rule sets have to be reevaluated [18]. Defining a fact in P-BEST is a time consuming task [3]. P-BEST was initially developed for Multics Intrusion Detection and Alerting System (MIDAS) [20], which is based on heuristic technique. The EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances) [21] [22] also uses P-BEST expert system in its rule based signature detection subsystem. EMERALD is an environment for misuse and attack detection in a large scale network, which employs both anomaly and misuse detection techniques. However, anomaly detection is based on traditional method identifying deviation from normal behavior [22].

The continuation of IDES resulted in Next Generation Intrusion Detection System (NIDES) [18] [23] [24]. NIDES is built on client-server technique where log data of various hosts on a network are gathered to a specific host, where the rule based anomaly detection is performed as well as P-BEST expert system is used for misuse detection.

AT&T's ComputerWatch [25] is an audit log analysis tool and an IDS that uses pre specified rules and compares user activities against those rules for suspicious misuse detection.

2.1.3 State Transition Approach

Based on finite state machine, state transition approach is used in various computer misuse or attack detection research works [26] [27] [28] [29]. In this approach attacks or misuses are a series of activities performed and single or combined activities can cause transition from one state of a monitoring sensor to another state and eventually reaches to the security state of a monitored system.

STATL [28] is an extensible language based on state transition technique to define attacks as a series of actions performed in order to penetrate into a computer system. STATL specification of attack includes key attributes sufficient to define that attack independent of any operating system or network environment. STATL's specification of attack is suitable to employ in IDSs in any operating system environment to screen for both host and network based attacks.

State Transition Analysis Technique (STAT) [30] [26] is a tools suite for misuse detection, which uses state transition mechanism to identify intrusive activities in computer systems. STAT tool suite includes STATL [28] language to define attack scenarios using the domain independent attributes of attacks in high level language in an abstract form. These definitions have to be included by the security system developers to meet the needs for a specific environment (such as specific host or network and operating systems). The basic idea of detecting an attack

or misuse is that before penetration by attacker, a computer system is in initial secure state and a series of activities by attacker cause system transition to various intermediate states before reaching to the final target state of successful system penetration.

UNIX State Transition Analysis Tool (USTAT) [31] [27] is the first STAT based tool to analyze audit log generated by UNIX-based systems for misuse identification in real-time. Although USTAT was initially capable of analyzing audit log of a single UNIX system host, its later capability of analyzing audit logs of multiple UNIX systems transformed it into NSTAT [32], which uses distributed computing to gather audit records of hosts and processes them in a central system for misuse and event detection in an distributed environment. WinSTAT [26] is also a STAT based misuse detection tool for host in Windows NT operating environment.

NetSTAT [29], a real-time network based misuse detection system, is also based upon STAT framework. NetSTAT uses network topology as a hypergraph model and STAT definitions of network based attacks to map specific misuse related to specific network configuration [3]. It uses network traffic log as input and a preprocessor to filter network packets for relevant network events and to generate abstract events [3].

USTAT and NetSTAT performed very well in the MIT Lincoln Laboratory's off-line intrusion detection system evaluation in 1998 and also in the Air Force Research Laboratory (ARFL) as well with a combined highest level of evaluation score [26].

Kumar and Spafford applied colored Petri nets, a mathematical representation, as a pattern matching technique for misuse detection in their intrusion detection project [33]. They represented each known attack as a sequence of events and then these sequences were transformed into graphs based on Colored Petri Nets. Pattern matching was performed using these graphs representing sequence of system state transitions that eventually ended up with an intrusion as a final state.

2.1.4 Data mining Approach

There are many applications of data mining in IDS research for misuse detection such as [34] and [35]. In this approach historical data of a monitored system usage has to be properly categorized as acceptable or unacceptable and labeled accordingly. By using a methodology, the system is trained to learn either what is acceptable or what is unacceptable for any network or system usage. If any deviation occurs, the system triggers an intrusive alert.

Lee and Stolfo [35] applied various data mining techniques such as association rule and various classification techniques to develop automated misuse detection models using audit logs and system calls. They developed an intrusion detection framework called MADAM ID (Mining Audit Data for Automated Models for Intrusion Detection). Their experimental evaluation showed very promising results [35]. The advantage of their framework is that it can automatically build models based on audit data.

2.2 Anomaly Detection based IDSs

2.2.1 Statistical Modeling Approach

Using statistical method for anomaly detection is one of the oldest techniques applied in IDS research. In this approach, the normal user behavior is first defined based on what is acceptable within the system usage policies. Using various statistical modeling techniques (such as models to identify standard deviation of normal usages patterns), user behavior is monitored and if there is any deviation from predefined normal behavior threshold, user's such anomaly activity will be considered as attack.

Qiao et al. [36] applied a statistical modeling technique, a Hidden Markov Model (HMM), on system calls to detect anomalous intrusions. To determine various state transitions that a special UNIX based process goes through from the start to the end, they collected all the system calls specific to that process and applied these system calls to a HMM. Using these state transition sequences, they built database of normal sequences and then monitored system call sequences against this database to detect anomaly. However, building a complete database of states for all the inputs are impossible since the input system calls and output states are mapped as one to one [37].

In addition to its rule based misuse detection module, IDES uses a statistical anomaly detection feature [15] [2]. In this approach, IDES creates profile of normal activities of a member user and the user group related to that member user based on audit records. Such profile contains only multivariate statistics of user normal activities such as frequency, covariance, and mean of some parameters specific to a particular user and corresponding user group [16]. Whenever a new record comes to the IDES, its anomaly detector compares it against the corresponding member user and user group profile to see if the new record deviates from the known normal profile. IDES's anomaly detection capabilities can also distinguish rare but acceptable user activities from anomalous activities by verifying whether the activity occurs during the specific user's normal working days or not.

Redesigned IDES, which resulted in NIDES described in earlier, also uses a statistical component called NIDES/STAT for anomaly detection [3] [18][38]. The system calculates a statistical anomaly score by processing the audit records based on individual characteristic measure (such as CPU usage, command

executed etc.) that defines the user profile of behaviors and calculates the sum of the squares of these abnormality scores for all the characteristics [3]. A subject's behavior on a monitored system is compared against the expected profile and NIDES/STAT calculates an anomaly value for the observed behavior. If the observed anomaly value is significantly different than a given threshold, then the observed behavior is considered as intrusive [39].

Haystack also uses an anomaly detection algorithm besides its signature based detection techniques [3]. In this algorithm a session vector is created based on the number of various characteristics count of the session and actions performed by a user during that specific session. Then the algorithm creates a Bernoulli vector consists of the components of that specific session vector that are outside of the pre-defined threshold range. Then the algorithm calculates a score using the Bernoulli vector and an attack vector for a specific type of attack. Based on the score, the algorithm can determine how anomalous the observed session is compare to other sessions for specific type of attack and can generate alarms accordingly. Haystack's anomaly detector requires additional information about attacks [3]. Haystack's anomaly detector algorithm was incorporated in Distributed Intrusion Detection System (DIDS) [40].

Emerald [2] [21], described in the misuse detection section, also includes a profiler engine to detect anomalous behavior. The profiler engine monitors user behavior from audit records and performs statistical anomaly detection based on user profile [41].

2.2.2 Machine Learning and Data Mining Approaches

Machine learning techniques have been successfully applied in IDS research, especially for anomaly detection. More recent IDS research works use machine learning techniques with combination of data mining for better detection of novel attacks. In this section we will explore some of the interesting research works employing these techniques.

Ghosh and Schwartzbaxd [42] used Artificial Neural Networks (ANNs) to detect both known and novel attacks against a computer system using supervised learning method. In anomaly detection program, they used Sun Microsystems's Solaris Basic Security Module (BSM) generated host audit data to train a multi-layered back propagation feed forward neural network to learn normal system behavior. The neural network was able to determine only a single string of events as anomaly or normal. In order to verify complete session activities, which consist of multiple BSM events data, they applied leaky bucket algorithm to capture recent activities from neural network generated outputs. The anomaly system was capable of producing 77.3% detection rate with 3.6% and 2.2% false positives by using a leak rate of .2 and .7 respectively on experiments using DARPA ' 98 IDS evaluation data.

In another work, Ghosh et al. [43] evaluated three machine learning algorithms for real-time intrusion detection based on program behavior in Sun Solaris system environment. They developed recurrent Artificial Neural Network topology based algorithm, string transducer technique using finite state automata in second algorithm and state tester algorithm that was capable of creating software behavior automatically in terms of finite automata. Their experimental results in 1999 Lincoln Laboratory/DARPA Intrusion Detection evaluation showed very promising results for all three algorithms in terms of run time for real-time detection capabilities.

Fox et al. [44] proposed one of the first neural network based IDS by modeling user behavior using unsupervised learning method. They used Kohonen's Self-Organizing Map (SOM) to learn characteristics of normal user and system behavior on multiuser computer. Some of the inputs to the SOM were total CPU consumption, RAM consumption, number of login failures, session length, total users, number of times help file was accessed, various disk accessed, paging activities of programs etc. [3]. Any statistical deviation from normal behavior would be identified as an indication of virus attack. However, their proposed system did not show interesting results.

Mohajerani *et al.* [45] used fuzzy logic and neural network combined together in Neuro-Fuzzy Intrusion Detection System (NFIDS). They used neural network to learn fuzzy rules for each type of attacks defined by the system administrator offline. After learning the fuzzy rules, neural network then performed fuzzy inference process. The authors claimed 90.6% correct attack predictions with 9.4% false positive on data collected for 15 days on an unprotected network. However, the authors did not mention the nature and volume of the network traffic tested, types of attack detected and capabilities (how fast the system ran) of handling traffic by their system. Moreover, because of its dependency on system administrators' knowledge of network attacks for defining fuzzy rules for different types of attack, such system as NFIDS might not be able to detect novel attacks accurately.

Yao *et al*. [46] developed a hybrid intrusion detection system by using both fuzzy logic and Support Vector Machine (SVM) techniques together. The authors applied SVM on network traffic data multiple times by changing values of different parameters to obtain sets of support vectors in the training phase of SVM. Then they applied fuzzy logic to develop fuzzy rules to make decision from various sets of the support vectors obtained from SVM training. The authors used KDD'99 network packets data for training and testing their system. They

reported 99% accuracy on 5 small sets of 10,000 data randomly chosen records from KDD'99 data. For higher accuracy the system had to create large number of rules, which is computationally very expensive.

A Packet Header Anomaly Detector (PHAD) uses a network intrusion detection algorithm that analyzes packet headers fields and detects anomalous field values using data mining technique for data link, network and transport layers protocols [16] [47]. The algorithm is trained with normal network traffic data where it can learn the normal range of allowable values for each field. The algorithm calculates the number of previously unseen values and frequencies of each field value for each field and assigns an estimated probability of a given field value being abnormal. An abnormal field score is calculated by using the time since the last abnormality was observed in that field. Finally a packet score is calculated by summing up all the abnormal field scores in the packet [47][48].

The authors utilized 1999 DARPA off-line intrusion detection evaluation data set [58] to train and test the system. Proposed system utilized 34 fields of packet headers available from DARPA 1999 dataset and was able to detect 76% of probes and 48% of DOS attacks at a rate of 10 false alarms per day [47]. However, merging PHAD system with other intrusion detection systems participated in the 1999 DARPA intrusion detection system evaluation improved average detection capability. PHAD's detector algorithms had very fast running time with low memory requirements.

Abouzakhar *et al.* [50] proposed a neuro-fuzzy technique for detecting distributed network attacks such denial of service (DoS). The proposed system learned the characteristics of network traffic by applying fuzzy logic function.

Chavan et al. [51] proposed a Neuro-Fuzzy based adaptive IDS for IP network where a database composed of pattern of signatures was built to complement SNORT signature database. These signatures were developed by analyzing network protocols and adaptive learning based on combination of Artificial Neural Networks and Fuzzy based inference techniques. While developing signature patterns, the authors considered some issues that were not incorporated in SNORT database. Using random records from DARPA 1998 IDS evaluation data set, the authors classified these records into five different classes of attacks and probes and extracted most relevant reduced number of variables necessary for each class by using a decision tree. Experimental results of one algorithm based on Artificial Neural Networks with 80 hidden nodes and another algorithm based on Evolving Fuzzy Neural Network of Mamdani type showed very high detection rate. Details of Mamdani type Fuzzy Inference System can be found in [52]. However, this IDS performed well on a very small dataset of pre-classified network data but its performance worsened as the number of input variables increased.

MITRE network uses IDS sensors that produce more than a million alarms everyday [54]. Many of these alarms are false alarms and it is quite impossible for human experts to review these alarms for possible attack detection. In order to detect anomalous network traffic accurately from large volume of network traffic data, there is an increasing interest in data mining for cyber security research [53] [54][55].

2.2.3 Outlier-based Data Mining Techniques

Outlier detection has been one of the single most popular data mining approaches used in IDS research. A number of research works [56, 57, 58, 59, 60 ,61] have already shown promising results by applying outlier detection techniques in network intrusion detection.

Basic idea of outlier detection is that if a data point is very different from the rest of the data, it will be defined as outlier. In a statistical distribution of data points based on a given mean and standard deviation, data points that do not fall under a specific range are considered as outlier. In network intrusion detection, anomalous network traffic data or abnormal network behavior is different from normal acceptable traffic data or normal network behavior based on some measures and may be identified as outlier.

There are various methodologies used to detect outlier in Computer Science and Statistics fields. Details of such methodologies can be found in [62]. Ramaswamy et al. [63] and Petrovskiy [64] discussed various efficient algorithms for outlier mining in large data set.

Minnesota Intrusion Detection System (MINDS), described in [6, 55, 56, 57, 58], successfully applied outlier detection techniques for mining network traffic data in order to detect cyber attacks on computer network.
MINDS uses a suite of data mining based detection algorithms in its various detection modules. MINDS has several different modules that detects various types of computer attacks and intrusive activities in a networked computing environment [55]. For example, its scan detector identifies suspicious scanning activities on a network; its anomaly detector identifies anomalous network traffic by examining the network packet headers, and its profile detector component help human network analyst characterizing abnormalities in network traffic [55].

In its anomaly detector, MINDS uses an efficient detection algorithm based on local outlier factor (LOF). Details of the LOF algorithm for outlier detection can be found in [65]. The network flows are collected for a specific time window and are fed to the anomaly detector in a batch. It assigns a LOF score (abnormality score) to each of the flow based on its nearest neighbors. The network traffic flows are sorted according to their LOF scores and human network experts further investigate the

flows and label them as attacks or normal. Experimental results showed that MINDS anomaly detector very accurately detected various types of anomaly in network traffic which were undetected by well known IDSs such as Snort [6]. Real network deployment of MINDS on University of Minnesota's computer network showed very promising results in detecting various forms of novel cyber attacks. MINDS requires batch execution of network connections in a time window and a human network analyst has to examine the network connections to determine any attack which are not suitable for real-time cyber attack detection from a large number of network connections.

Lazarevic et al. [57] compared several outlier detection techniques and algorithm based on unsupervised support vector machine for network intrusion detection. For the outlier detection techniques, the authors applied data mining algorithms based on distance to the k-th nearest neighbor (K-NN) approach, nearest neighbor approach (NN), Mahalanobis- distance based approach and density-based local outlier factor (LOF) approach. Details of these statistical techniques can be found in [62] and [65]. The authors adopted an unsupervised learning technique by transforming a standard supervised support vector machine. Experimental result on DARPA'98 data showed that NN approach and LOF approach outperformed Mahalanobis- distance based approach with a 2% false positive rate for detecting both bursty and single connection attacks, where unsupervised support vector machine approach performed best but with a higher (4%) false positive rate in both detections [57].

K-means algorithm, discussed in [66], has been widely used for clustering techniques in data mining for intrusion detection for decades. For example, Faraoun and Boukelif [67] used traditional K-means algorithm for clustring in Neural-network based network intrusion detection. However, traditional K-means algorithm *includes* number of clusters dependency and degeneracy problem that is choosing optimal number of clusters (optimal value of k) is quite problematic. To address this issue, Guan et al. [68] developed Y-means algorithm, a heuristic clustering method for intrusion detection. Experimental result of applying Y-means on KDD '99 data showed 89.89% detection rate with 1% false alarm [68].

3. Research in Intrusion Prevention System (IPS)

Traditional IDS research focus mainly on how to effectively detect attacks, not to prevent them [69]. Success of an IDS solely depends on how accurately it detects attacks with a lower false alarm. Traditional IDSs just notify administrators after detecting an attack and administrators have to manually take proper actions. IDSs are "passive" and they do very little to nothing at all to prevent an attack. Due to this limitation of IDSs and increasing number of cyber attacks threatening emerging information technology, there is a widespread attention being drawn to IPS research recently. IPS research takes its root from the IDS research. Basically IPS is a technology that holds the combine characteristics of firewall technology and IDS [69][1], that is, it has the capabilities of detecting, isolating and blocking malicious attacks in real-time. Various IPSs that are capable of preventing attacks in various network layers are discussed in [70].

Despite having considerable promise in cyber attack defense mechanisms, IPS research is still considered to be in its premature form [70]. Most of the handful IPS research works available until recently inherit from IDS research works with one more added feature of taking actions to prevent attacks being occurred. Most of the IPSs available rely on signature matching techniques for attack detection. Snort [10] is one such widely used IPS. Some of the interesting IPS research works include [69, 71, 72, 73, 74].

Zhang et al. [69] described a network-based distributed IPS. The rule-based distributed IPS combined a network management module with its intrusion detection modules where multiple IDSs were placed in various locations on the network. The IPS was based on application-specific integrated circuits *(*ASICs*)* and therefore, it had faster processing ability. However, the authors did not mention any experimental results for the system evaluation.

In a comparison of publicly available intrusion prevention tools, John Wilander and Mariam Kamkar [71] investigated various forms of buffer overflow and format string attacks using five tools such as security testing tools ITS4, Flawfinder, RATS, Splint and BOON. The authors found higher false positive rates for tools ITS4, Flawfinder and RATS, which were based on lexical analysis and lower true positive rates for tools Splint and BOON, Which were based on syntactic and semantic analysis.

Locasto et al. introduced a hybrid adaptive intrusion prevention system called FLIPS (Feedback Learning IPS) [72]. Host-based FLIPS used both signature matching technique and anomaly based classification technique to detect and prevent code injection attacks [72]. It used an intermediate emulator to detect injected malicious attack code and to generate attack signature. Experimental results showed promising capabilities of FLIPS in detecting injected code attacks. In an experiment FLIPS detected and blocked 61 of the 67 homogeneous attack instances tested and 20 of the 22 mixed attack instances tested.

Weinsberg et al. [73] proposed a high performance string matching algorithm for network-based IPS. The hardware-based algorithm had the capability of matching multiple patterns at a time which made it faster. Experimental results showed that the algorithm achieved

very faster processing speed for string matching tasks.

Battistoni et al. [74] proposed WHIPS, a host-based IPS for Windows operating systems. The system is solely based on monitoring the critical windows system calls in kernel mode. In windows environment, the proposed system identifies "privileged processes" and differentiates the harmful processes by examining the access token to identify the critical system calls. The authors mentioned that WHIPS would be more effective if it was implemented into kernel of the Windows operating system which is quite impossible to implement since Windows kernel source code are strictly protected by its authority and not available as open source product.

4. Conclusion

Current research involving cyber defense has limitations. As the prominent parts of cyber defense research, both the IDS and IPS research works have drawbacks and limitations due to detection techniques used, lack of real attack data for testing and validation, not considering detection speed etc. The main limitation of misuse detection based IDSs is that they only can detect known attacks accurately. They are unable to detect previously unseen attacks or novel attacks. Moreover, predefine attack specification has to be provided to the IDS for misuse detection, which requires human security experts to manually analyze attack related data and formulate attack specifications. Compare to huge volume of data produced for analysis, this task requires extensive human labor and time. Attack specification can be generated automatically by applying various automated techniques. Most of the misuse detection systems lack this capability. Most of the systems focus on data produced by single source. Abad *et al.* [75] showed that correlation of log information from various sources such as network traffic logs, application logs, operating system calls etc. could improve the performance of intrusion detection systems.

References

[1] Fuchsberger A., Intrusion Detection Systems and Intrusion Prevention Systems, Information Security Technical Report, Volume 10, Issue 3, 134-139 (2005)

[2] Axelsson S. Intrusion detection systems: A survey and taxonomy. Department of Computer Engineering, Chalmers University, Technical Report 99-15 (2000).

[3] Ning P.,Jajodia S. Intrusion Detection Techniques. In H. Bidgoli (Ed.), The Internet Encyclopedia. John Wiley & Sons. (2003).

[4] Debar H. An Introduction to Intrusion-Detection Systems.Proceedings of Connect'2000, Doha, Qatar (2000).

[5] Roesch M. Snort - Lightweight Intrusion Detection for Networks. Proceedings of the 13th USENIX conference on System administration, Seattle, Washington (1999),

[6] Ertoz L., Eilertson E., Lazarevic A. et al. MINDS - Minnesota Intrusion Detection System. In Data Mining - Next Generation Challenges and Future Directions. MIT Press (2004).

[7] Smaha S.E. Haystack: An intrusion detection system. In Proceedings of the IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL, USA, December 1988. IEEE, IEEE Computer Society Press, Los Alamitos, CA, USA.

[8] Network Flight Recorder. Network Flight Recorder, Inc. Available at: http://www.nfr.com. Accessed on 28 October 2008.

[9] Paxson V. Bro: a system for detecting network intruders in real-time. Computer Networks: The International Journal of Computer and Telecommunications Networking, v.31 n.23-24, p.2435-2463 (1999).

[10] Kruegel C., Toth T. Using Decision Tree to Improve Signature Based Intrusion Detection. 6Th Symposium on Recent Advances in Intrusion Detection (REID), Lecture Notes in Computer Science, Spring Verlag, USA (2003).

[11] McCanne S., Leres C., Jacobson V. libpcap, available via anonymous ftp to ftp.ee.lbl.gov (1994).

[12] Bro Intrusion Detection System. Available at : http://bro-ids.org/. Accessed 28 October 2008.

[13] Lin J., Wang X., Jajodia S. Abstraction-Based Misuse Detection: High-Level Specifications and Adaptable Strategies. csfw, p. 190, 11th IEEE Computer Security Foundations Workshop (1998).

[14] Lippmann R. The Role of Network Intrusion Detection, In Proceedings of the Workshop on Network Intrusion Detection, H.E.A.T. Center, Aberdeen, MD (2002).

[15] Lunt TF., Jagannathan R. A Prototype Real-Time Intrusion-Detection Expert System.1988 IEEE Symposium on Security and Privacy. sp, p. 59 (1988).

[16] Javitz HS., Valdes A.The SRI IDES Statistical Anomaly Detector. 1991 IEEE Symposium on Security and Privacy, sp, pp. 316 (1991).

[17] Lunt TF. Real-Time Intrusion Detection. Computer Security Journal Vol. VI, Number 1. pp. 9-14 (1989).

[18] Anderson D., Lunt T., Javitz H., Tamaru A., Valdes A. Detecting UnusualProgram Behavior Using the Statistical Component of the Next-generation IntrusionDetection Expert System (NIDES). SRI International Computer Science LaboratoryTechnical Report SRI-CSL-95-06 (1995).

[19] Lindqvist U., Porras PA. Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST).1999 IEEE Symposium on Security and Privacy, sp, p. 0146 (1999).

[20] Sebring et al. Expert Systems in Intrusion Detection: A Case Study. Proceedings of the11th National Computer Security Conference, Baltimore, MD (1988).

[21] Porras PA., Neumann PG. Emerald: Event monitoring enabling responses to anomalous live disturbances. In Proceedings of the 20th National Information Systems Security Conference, pp. 353-365 (1997).

[22] Neumann PG., Porras PA. Ex-perience with Emerald to date, In Proceedings of the 1st Workshop on Intrusion Detection and Network Monitoring. USENIX (1999).

[23] Anderson D., Frivold T., Valdes A. Next-generation intrusion detection expert system (NIDES). Technical Report SRI-CSL-95-07, SRI International, Computer Science Lab (1995).

[24] Javitz HS., Valdes A.. The NIDES statistical component: description and justication. In Technical Report, Computer Science Labratory, SRI International (1993).
[25] Dowell C., Ramstedt P. The Computer Watch data reduction tool. Proc. 13th National Computer Security Conf., Washing-ton, DC, pp. 99–108 (1990).
[26] Vigna G., Eckmann S., Kemmerer R. The STAT Tool Suite. In: Proceedings of DISCEX 2000, Hilton Head, South Carolina, IEEE Computer Society Press (2000)
[27] Ilgun K. USTAT: A Real-time Intrusion Detection System for UNIX. In: Proceedings of the IEEE Symposium on Research on Security and Privacy, Oakland, CA (1993)
[28] Eckmann S., Vigna G., Kemmerer R. STATL: An Attack Language for State-based Intrusion Detection. In: Proceedings of the ACM Workshop on IntrusionDetection Systems, Athens, Greece (2000)
[29] Vigna G. , Kemmerer RA. NetSTAT: A Network-Based Intrusion Detection Approach. Proceedings of the 14th Annual Computer Security Applications Conference, pp.25, (1998)
[30] Porras P. STAT – A State Transition Analysis Tool for Intrusion Detection. Master's thesis, Computer Science Department, University of California, Santa Barbara (1992)
[31] Ilgun K. USTAT: A Real-time Intrusion Detection System for UNIX. Master's thesis, Computer Science Department, University of California, Santa Barbara (1992)
[32] Kemmerer RA. NSTAT: A Model-based Real-time Network Intrusion Detection System. Technical Report TRCS-97-18, Department of Computer Science, UC SantaBarbara (1997)
[33] Kumar S., Spafford E. A pattern matching model for misuse intrusion detection. Proc. 17th National Computer Security Conf., pp. 11–21(1994).
[34] LEE W., STOLFO SJ. Data mining approaches for intrusion detection. In Proceedings of the 7th Symposium on USENIX Security, San Antonio, TX (1998).
[35] LEE W., STOLFO SJ. MOK KW. A data mining framework for building intrusion detection models. In Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, California (1999).
[36] Qiao Y., Xin XW., Bin Y.,Ge S..Anomaly intrusion detection method based on HMM (2002). IEEE Electronic Letters Online No: 20020467
[37] Hoang XD., Hu J., Bertok P. A Multi-layer Model for Anomaly Intrusion Detection Using Program Sequences of System Calls. 11th IEEE International Conference on Networks (ICON 2003), Sydney, Australia (2003).
[38] Ning P. , Jajodia S. , Wang XS. Abstraction-based intrusion detection in distributed environments. ACM Transactions on Information and System Security (TISSEC), v.4 n.4, p.407-452 (2001).
[39] Zhang K., Yen A., Zhao X., Massey D., Wu SF. On detection of anomalous routing dynamics in BGP. In IFIP-TC6 Networking 2004, vol. 3042 of Lecture Notes in Computer Science, Springer (2004).
[40] Snapp SR, BrentanoJ., Dias GV, Goan TL, Heberlein LT., Ho C., K. Levitt KN., Mukherjee B., Smaha SE., Grance T.,Teal DM, Mansur D. DIDS (Distributed Intrusion Detection System) -Motivation, Architecture, and an Early Prototype. In Proceedings of the 14th National Computer Security Conference, pages 167-176, (1991).
[41] Chatzigiannakis V., Androulidakis G., B. Maglaris. A Distributed Intrusion Detection Prototype Using Security Agents. HP OpenView University Association (2004).
[42] Ghosh AK., Schwartzbaxd A.. A study in using neural networks for anomaly and misuse detection. In Proceedings of the 8th USENIX Security Symposium (1999).
[43] Ghosh AK. , Michael C., Schatz M. A Real-Time Intrusion Detection System Based on Learning Program Behavior. Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection, p.93-109 (2000).
[44] Fox KL., Henning RR., Reed JH., Simonian RP. A neural network approach towards intrusion detection. Proceedings of the 13th National Computer Security Conference, Washington, D.C. Gaithersburg, MD: NIST, 125-134 (1990).
[45] Mohajerani M., Moeini A., Kianie M. NFIDS: A Neuro-fuzzy Intrusion Detection System. Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems, pp348-351 (2000).
[46] Yao JT, Zhao SL, Saxton LV. A study on fuzzy intrusion detection. In: Belur V. Dasarathy, editor. In Proceedings of SPIE Vol. 5812, Data Mining, Intrusion Detection, Information Assurance, And Data Networks Security, Orlando, Florida, USA: SPIE, Bellingham, WA, pp. 23–30 (2005).
[47] Mahoney M., Chan P. Detecting Novel Attacks by Identifying Anomalous Network Packet Headers. Florida Institute of Technology, Melbourne, FL. Technical report CS-2001-2 (2001).
[48] Mahoney MV., Chan PK. PHAD: Packet header anomaly detection for identifying hostile network traffic. Technical report, Florida Institute of Technology, Melbourne, FL. Technical report (2001).
[49] Lippmann R., et al.The 1999 DARPA Off-Line Intrusion Detection Evaluation. Computer Networks 34(4) 579-595 (2000).
[50] Abouzakhar N.S, Manson G.A. Networks security measures using neuro-fuzzyagents. Journal of Information Management and Computer Security. 11 (1), pp.33-38 (2003).
[51] Chavan S., Shah K., Dave N., Mukherjee K., Abraham A.,Sanyal S. Adaptive neuro-fuzzy intrusion detection systems. In Proceedings. ITCC 2004. International Conference on Information Technology: Coding and Computing, volume 1, pages 70–4, Las Vegas, NV, USA (2004).
[52] Mamdani EH. Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. Int.J Man-Machine Stud.,7 (1975).
[53] Julish K, Data mining for intrusion detection: A critical review. IBM Res. rep. RZ 3398, 93450, (2002).
[54] Bloedorn E., et al., Data Mining for Network Intrusion Detection: How to Get Started, MITRE Technical Report (2001).
[55] Chandola V., Eilertson E., Ert¨oz L., Simon G., Kumar V.. Data mining for cyber security. Data Warehousing and Data Mining Techniques for Computer Security (2006).
[56] Dokas P., Ertoz L.,Kumar V., Lazarevic A., Srivastava J., Tan PN. Data mining for network intrusion detection. In Proc. 2002 NSF Wrokshop on Data Mining, 21.30 (2002)

[57] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava and V. Kumar, "A comparative study of anomaly detection schemes in network intrusion detection," in Proc. of SIAM Conf. Data Mining (2003).
[58] Zanero S., Savaresi SM. Unsupervised learning techniques for an intrusion detection system. Proceedings of the 2004 ACM symposium on Applied computing, Nicosia, Cyprus (2004,).
[59] Ertoz L., Eilertson E., Lazarevic A., Tan PN., Dokas P., Kumar V., Srivastava J. Detection of novel network attacks using data mining. In Proceedings of the 2003 ICDM Workshop on Data Mining for Computer Security, Melbourne, Florida, USA (2003).
[60] Ghoting A., Otey ME., Parthasarathy S. Loaded: Link-based outlier and anomaly detection in evolving data sets. In Proceedings of the 4th International Conference on Data Mining, pp.387-390 (2004).
[61] Wang k., Stolfo SJ. Anomalous payload-based network intrusion detection. In Recent Advance in Intrusion Detection (RAID) (2004).
[62] Hodge V., Austin J. A Survey of Outlier Detection Methodologies. Artificial Intelligence Review, v.22 n.2, pp.85-126 (2004).
[63] Ramaswamy S. , Rastogi R. , Shim K. Efficient algorithms for mining outliers from large data sets. Proceedings of the 2000 ACM SIGMOD international conference on Management of data, Dallas, Texas, United States, p p.427-438 (2000).
[64] Petrovskiy MI. Outlier detection algorithms in data mining systems. Programming and Computer Software 29, 4, pp.228-237 (2003).
[65] Breunig MM., Kriegel HP., Ng RT., Sander J. LOF: identifying density-based local outliers. Proceedings of the 2000 ACM SIGMOD international conference on Management of data, Dallas, Texas, United States, pp.93-104 (2000).
[66] MacQueen J. Some methods for classification and analysis of multivariate observations. Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability, 2:28.297 (1967).
[67] Faraoun KM., Boukelif A.. Neural Networks Learning Improvement Using The K-means Clustering Algorithm to Detect Network Intrusions. International Journal of Computational Intelligence 3;2; p.28-36 (2007).
[68] Guan Y., Ghorbani AA., Belacel N. Y-means: a clustering method for intrusion detection. In Canadian Conference on Electrical and Computer Engineering, pp. 1-4, Montral, Qubec, Canada (2003). Zhang X., Li C., Zheng W. Intrusion Prevention System Design. The Fourth International Conference on Computer and Information Technology (CIT'04), (2004).
[69] Desai N. Intrusion Prevention Systems: the Next Step in the Evolution of IDS. Feburary 2003. http://www.securityfocus.com/infocus/1670. Accessed 30 November, 2008.
[70] Wilander J., Kamkar M. A comparison of publicly available tools for static intrusion detection. In Proceedings of the Nordic Workshop on Secure IT Systems, pp. 68–84 (2002).
[71] Locasto M., Wang K., Keromytis A., Stolfo S.. FLIPS: Hybrid adaptive intrusion prevention. In RAID (2005).
[72] Weinsberg Y., Tzur-David S., Anker T., Dolev D. High performance string matching algorithm for a network intrusion prevention system (nips). High Performance Switching and Routing (HPSR06) (2006).
[73] Battistoni R., Gabrielli E., Mancini LV. A host intrusion prevention system for Windows operating systems. In Proceedings of the 9th European Symposium on Research Computer Security (ESORICS 2004), pp.352–368 (2004).
[74] Abad C., Taylor J., Sengul C., Yurcik W. , Zhou Y., Rowe K. Log Correlation for Intrusion Detection: A Proof of Concept. Proceedings of the 19th Annual Computer Security Applications Conference, pp.255 (2003).

Mohammad A. Faysel received his BS and MA degrees in Computer Science and currently is a PhD candidate in Biomedical Informatics at University of Medicine and Dentistry of New Jersey. His research interests include cyber security for healthcare information systems, bio-surveillance systems, clinical informatics, and healthcare outcome research.

Syed S. Haque, PhD, is a professor and the Chair of the Department of Health Informatics at UMDNJ-School of Health Related Professions. Dr. Haque's research interests include healthcare data mining and knowledge discovery, design of intelligent training systems, surveillance techniques, healthcare outcomes research, and healthcare information systems security.