INTRUSION DETECTION IN CLOUD COMPUTING

by

Adeel Rasheed Minhas

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of MS in Information Security

Dec 2016

SUPERVISOR'S CERTIFICATE

It is certified that the final copy of thesis has been evaluated by me, found as per specified format and error free.

Dated: ________ Dec 2016

(**Col Imran Rashid, PhD**)

DECLARATION

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

____

(Adeel Rasheed Minhas)

ABSTRACT

Cloud Computing is a diversified field which enables commodious and convenient sharing of computing resources including storage, processing, networks, servers, applications and services over the internet. The hassle-free utilization of cloud computing from around the globe at any time without the involvement of own resources makes it very attractive to the users and the main reason for its widespread application. However, with all the benefits, cloud environment is prone to many security concerns. Network intrusion is one such obstruction which raises concerns as it could affect confidentiality and integrity of data besides disrupting availability of services. Virtual Machines are a key component of cloud computing model. The security of these VMs is of paramount importance as their appearance is invisible but utility is ineluctable. In this research, I have made an endeavor to use anomaly based intrusion detection methodology to identify events and observe incidents which do not comply with expected legitimate patterns of network usage in a cloud environment utilizing VMs. The proposed solution observes the VM traffic, detect anomalous behavior and restrain the network system through filtered influx.

ACKNOWLEDGMENTS

I would like to thank Allah Almighty for His countless blessings. I express my appreciation to my family, colleagues and the faculty for providing their enormous support to help me to do this research. Without their relentless support, assistance and prayers, I would not have reached to culmination point in a peaceful state of mind.

I extend my deepest gratitude to my supervisor, Colonel Dr. Imran Rashid, who provided me a platform and gave me the liberty to work in the area of my interest and extended his tremendous support prior to as well as during the course of this research. His technical guidance, encouragement, ideas and perspective were vital for completion of this tedious task. His support gave me confidence and helped me to understand about the subject matters deeply and inspired me towards my goals.

I would also like to express my sincerest appreciation to Assistant Professor Dr. Hammad Afzal and Lecturer Waleed Bin Shahid for being an important part of my Research Supervisory Committee. Their scholarly guidance, assistance and knowledge have been meaningful for successful completion of my research.

Finally, I am grateful and thankful to Military College of Signals and National University of Sciences and Technology for providing me a chance to help achieve excellence by being associated with the prestigious institutions.

Adeel Rasheed Minhas,
December, 2016.

LIST OF FIGURES

LIST OF TABLES

LIST OF ANNEXURES

CHAPTER 1

INTRODUCTION

Cloud Computing is a useful and progressing technology which allows its users to conveniently access a pool of full featured applications, software development tools and computing infrastructures on rental basis. It is a cost-effective solution which provides the convenience of ubiquitous use of hassle free computing resources. Due to its appealing features, Cloud Computing attracts a lot of users to use the offered services. All these services are provided to the intended users swiftly, with negligible management efforts and almost no collaboration with the service provider. Cloud services are provided to its users, are mainly of three types being Software as a service, Platform as a service and

Infrastructure as a service in four different deployment models being Private cloud, Community cloud, Public cloud and Hybrid cloud. These convenient arrangements of services and deployment scenarios facilitate users to acquire services which suit the best to their demands [1]. The widespread popularity of Cloud computing has not only attracted the users but it has equally drawn the attention of hackers and attackers. The security of cloud at the network level is generally affected by Denial of service (DoS), Distributed denial of service (DDoS), Address resolution protocol (ARP) poisoning or spoofing, Man-in-the-middle (MITM), Internet Protocol (IP) spoofing, Back door channel attacks, Routing information protocol attack and other such types. The attempt to secure the Cloud environment at the network level gave rise to Network based Intrusion Detection System (NIDS) [2].

Intrusion Detection System (IDS) is an effective instrument to safeguard the computer network system from invasions or attacks. The rapid advancements in this growing field has provided an opportunity to intruders, attackers and hackers to find different illegitimate ways to comprise a system or network [3]. Hence, with the evolution of new technologies the threat vector is also rising and the biggest problem encountered today is the protection of 'Big Data' i.e a great volume of network traffic data gathered in network communication [4]. The NIDS analyzes the attacks by detecting the network traffic packets, impacting several hosts which are connected to the network. NIDS have proved to be effective against the network based attacks and a preferred choice in these times where networks have dominated many other technologies. The application of a smart and intelligent NIDS can only be effective because of Big Data. Its positioning in the network is very critical for its effectiveness. A network security analyzer must place the sensor at the appropriate location because it will have serious effects on its efficacy [3]. Anomaly based IDS is yet another valuable variant in NIDS which helps to detect both new and already known attacks. Anomaly based technique uses a different methodology which deviates from the signature based approach which allows it to detect the zero day, insider or any other novel attack approach. However, as a side effect, it generates greater false alarms on various anomalous behaviors which actually are not attacks. A careful training of the system is, therefore, vital to get the appropriate or desired results [5].

Machine Learning techniques is quite useful in Anomaly based IDS for the purpose of identifying the true attacks and disseminating the non-attacks, enabling higher true positive / negative rates. The parallel processing approach is also useful in dealing with the 'Big Data'. Machine Learning focuses on developing the computer programs and train them to grow and change when they are exposed to new data which can either be supervised: applying changes to data what has been learned in the past or unsupervised: where inferences can be drawn from the datasets [6].

In my approach, I have tried to build an anomaly based IDS for network traffic. The main idea behind was to deviate from the traditional signature based approach to detect new and insider attacks in a Cloud environment. I have made use of MIT DARPA datasets for preparation, preprocessing and exposed it to Machine Learning processes through training and classification to achieve better true positive and negative results.

1.1 Problem Statement

Anomaly based IDS is an effective option of detecting attack patterns in networks but there are certain obstructions attached to it. First, the availability of Big Data on network traffic has made intrusion detection a complicated task [5]. Secondly, deployment of anomaly based IDSs is a cumbersome activity and hence one of the reasons for its under deployment [7]. Thirdly, a mainstream anomaly based IDSs use either machine learning or data mining techniques which take certain feature vectors comprising complicated attributes as their inputs. The extraction of these feature vectors remains a tedious and complicated task [8].

Therefore, deployment of anomaly based IDS is virtually impractical for a normal network administrator relying on such complicated feature vectors.

The idea of using Machine Learning in the field of Information Security is a fine choice. Machine learning techniques are powerful tools and have evolved to be important in detection and prevention of malicious activities in a network based environment. There are some problems affiliated with these techniques where a system requires constant human interactions for attack labelling with respect to the normal traffic behavior to support the process [9]. On the other hand, many IDS perform a satisfying job but their performance deteriorates and a real-time performance is not essentially achieved.

Thus, the main objective of this research is to offer an efficient machine learning based technique for intrusion detection to extract and sift out a suitable subset of data and use the required patterns for intrusion detection in real time.

1.2 **Motivation**

Security has always been one of the basic needs of human beings and one cannot image a life without security. Computer security has the same level of importance in the world of technology and Intrusion detection is a core area of network security and requires to be highly efficient. Any unauthorized access to these networks can play havoc and jeopardize the working of the resource owner. Therefore, the security of a network begins with intrusion detection. Another concern that has emerged over a period of time is dealing with Big Data. Many approaches of IDS utilizing Machine Learning techniques are suggested with their merits and demerits, thus causing a void as well as room for further research in this field.

My research work started with the idea of developing my own dataset but I realized that working on a globally accepted dataset, which has been used in many other research works will be a better idea. Therefore, I made a choice of MIT DARPA Intrusion Detection Datasets. The main attraction was to have a depiction of real network logs which allows the system to be implemented in a practical environment. By doing so, the research work became more complicated as it was anticipated earlier but provided a motivation to proceed with the development work.

1.3 Relevance to Industrial and Military needs

1.3.1 **Industrial Needs.** The governmental, private and commercial use of Cloud Computing is on the rise. Being a cost-effective solution a large percentage of the organizations are being migrated to the Cloud environments. Network is the most vital as well as vulnerable part of Cloud Computing and network attack detection will play a significant role in prevention of malicious attacks.

1.3.2 **Military Needs.** Data centers of military establishments are providing services based on the methodology of Cloud Computing. The users are being facilitated through a dedicated network. The security of this network is the key element which is always prone to malicious attacks. Anomaly based intrusion detection technique will help mitigate these attacks.

1.4 Research Objectives

The whole research work has been divided into different tasks which have been perceived from the problem statement and motivation.

Task 1: Analyze the security concerns of VMs at network level in Cloud Computing environment.

Task 2: Select and analyze a globally recognized dataset for further experimentation.

Task 3: Propose a framework / model to provide anomaly based intrusion detection in network traffic using machine learning technique

Task 4: Develop an algorithm to extract benign and malicious traffic patterns in the dataset

Task 5: Test the developed system on a Cloud testbed

1.5 Thesis Organization

This study is organized in seven chapters, detail as under: -

- Chapter 2 offers the overview of Cloud Computing, Intrusion Detection System, Anomaly Based IDS and Machine Learning.
- Chapter 3 analyses MIT Lincoln DARPA Intrusion Detection Database in detail.
- Chapter 4 present the proposed model / architecture of IDS.
- In Chapter 5, the deployment methodology of Cloud Testbed is described.
- Results and related graphs are shown in Chapter 6.
- Finally, Chapter 7 concludes the thesis with future work.

CHAPTER 2

BACKGROUND

This chapter will present an overview of Cloud Computing, Intrusion Detection System and Machine Learning techniques and processes.

2.1 Cloud Computing

The ‘National Institute of Standards and Technology (NIST)’ defines cloud computing as: cloud computing is a demand based approach of computing resources built for universal access to include networks, storage areas, servers, applications and services offered without directly interacting with the service providers [1]. Figure 2.1 shows how a cloud system is connected with its consumers.

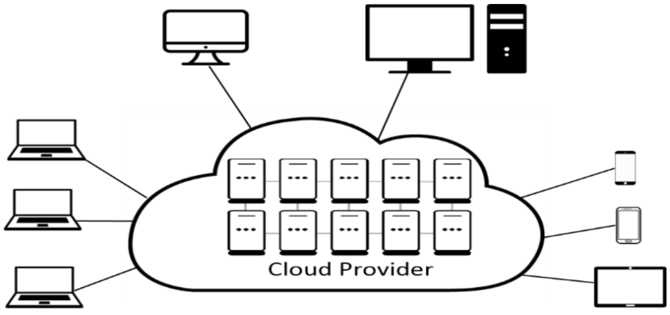


Figure 2.1: Cloud System connected with consumers

2.1.1 **Cloud Computing Deployment Models**

Cloud computing model can be deployed in four basic environments: Private, Public, Community and Hybrid Cloud Environment [10]: -

a. **Private Cloud Environment**

This cloud environment is employed at customer premises to access computational resources. It may be centralized for a single consumer or dispersed for numerous consumers. This arrangement of cloud can also be termed as internal cloud. Of all the cloud environments, private cloud offers enhanced security with fault tolerance [11]. Figure 2.2 shows a private cloud scenario.

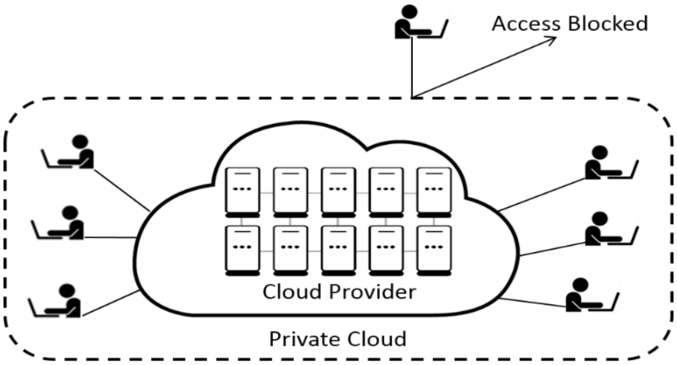


Figure 2.2: Private Cloud Environment

b. **Community Cloud Environment**

Community can be called a set of organization where each participant performing the prescribed job to meet the certain requirement. The community cloud environment can either be on site or out sourced. On Site Community Cloud Computing is the solution where some member of the community implements the cloud environment. This scenario merits solutions to sharing as clients from different participating organization access common group of computing assets. It is necessary that at least one community member must implement the cloud services for a community cloud to be operative. As far as security is concerned, each participant organization will have its own security perimeter and connected through a secure communication link [10]. Figure 2.3 shows how a community cloud scenario.

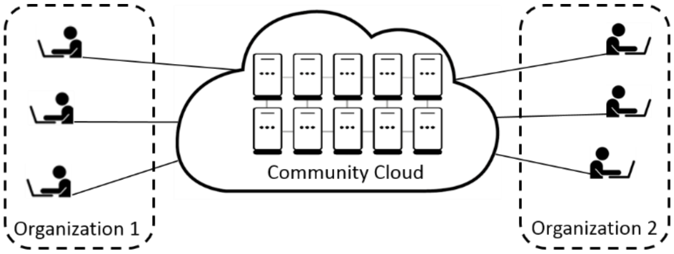


Figure 2.3: Community Cloud Environment

c. **Public Cloud Environment**

In this scenario, provider implements cloud environment on a larger scale where client from anywhere access the cloud computing resources over the network. Like other cloud environments, public cloud scenario is not client specific such as group of people, community or some organization. Public cloud is a shared cloud where end users do not possess requisite control over the structural resources. It is comparatively an inexpensive solution of cloud environments [11]. Figure 2.4 shows how a public cloud scenario.

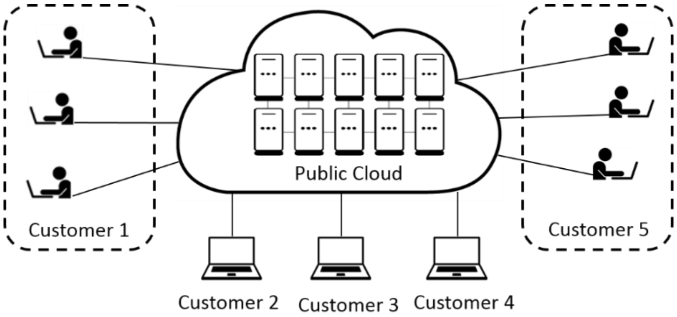


Figure 2.4: Public Cloud Environment

d. **Hybrid Cloud Environment**

This cloud environment is the blend of two or more private, community or public clouds. It is the consumer's demands which derive the vendor towards specific combination of cloud scenarios in order achieve desired results. Secondly, cloud provider is also giving a workable and efficient environment keeping in view all possible scenarios. There may be a situation where client is using a private cloud for routine workloads but as and when required accesses one or more external clouds for high demand job. The security space and infrastructure of the system is controlled by role and personal policies [11]. Figure 2.5 depicts a Hybrid Cloud Environment.

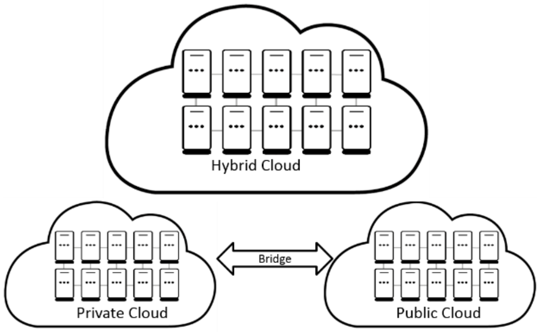


Figure 2.5: Hybrid Cloud Environment

Table 2.1 shows the security issues pertaining to different cloud computing deployment models [12].

Cloud Models	Setup Cost	Security Issues	Control Issues
Private Cloud	High	Low	Low
Community Cloud	Low	Moderate	Moderate
Public Cloud	Very High	Very High	Very High
Hybrid Cloud	Moderate	High	High

Table 2.1: Issues in Cloud Models

2.1.2 Cloud Computing Services

Cloud computing model is a stack of three services, each one being a particular resource in a cloud computing environment. These services describe how cloud services are delivered to the clients. The fundamental model has software, platform and infrastructure as the cloud services. Figure 2.6 shows a usual cloud computing stack.

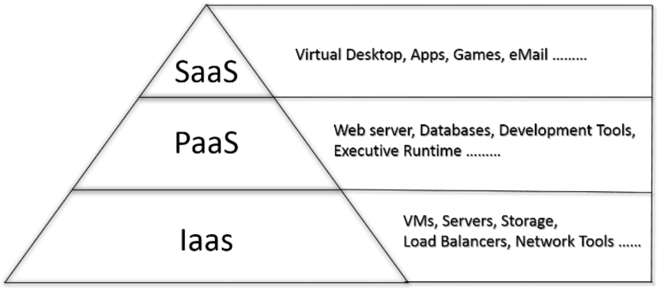

SaaS
Virtual Desktop, Apps, Games, eMail ....
PaaS
Web server, Databases, Development Tools, Executive Runtime ....
Iaas
VMs, Servers, Storage, Load Balancers, Network Tools ......

Figure 2.6: Cloud Computing Model Stack

a. Software as a Service

In its simplest term, Software as a Service (SaaS) can be defined as softwares deployed over the cloud. These softwares are made available to the clients through a subscription from the provider on demand basis. It is a very convenient and cost effective way of accessing desired applications or softwares without the hassle of buying licenses. The user is provided access to demanded softwares which is not needed to by bought or even installed on own hardware. Instead, the user rents it for a particular time as per requirement [12]. Some of the advantages of SaaS are listed below: -

- Convenient access to commercial softwares.
- Software management to a central location.
- Software can be delivered to multiple clients simultaneously.
- Worldwide access to softwares without the need of own hardware resources.
- Upgrades and patch installation is provider's responsibility.
- APIs allow for integration of various software pieces.

b. **Platform as a Service**

Platform as a Service (PaaS) is a computing platform which allows a user to create applications or softwares conveniently without the complexity of buying and maintaining the base requirements. PaaS is a more generalized form of SaaS. Contrary to SaaS, PaaS provides developers the platform for creation of software. The development environment is provided by the cloud provider that can be utilized by the user for development purposes. It offers prospects for developers to develop and deploy web based application without specialized expert opinion [12]. Some of the advantages of PaaS are listed below: -

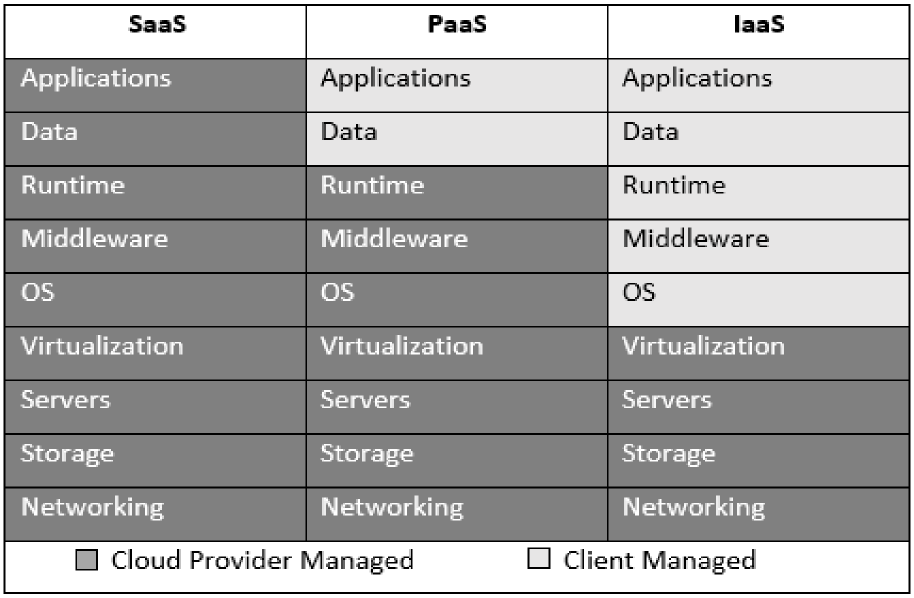
- User can instantly develop, deploy and run the application.
- Eliminate overheads to deploy and manage applications.
- Best and up to date technology for cheaper prices.
- User Interface (UI) based on Web technologies to create, modify, test and deploy different applications.
- Numerous users use the development applications concurrently.
- Integration of web services and databases through common standards.

c. Infrastructure as a Service

Infrastructure as a Service (IaaS) is a mean of providing on demand OS, storage, servers and networks as cloud computing services. Instead of purchasing infrastructures like network equipment, servers, relevant softwares, datacenter space and hardwares, clients subscribe to fully outsourced service on demand basis. The whole infrastructure is provided by the cloud provider to the customer to run his application including all the dedicated softwares and hardwares in a way where customer can handle varying load conditions. In IaaS, the sole responsibility of managing all infrastructure is that of the cloud provider [12]. Some of the advantages of IaaS are listed below: -

- Resources are provided as a service.
- Allows dynamic scaling of resources.
- Multiple users can be handled on same hardware using virtualized environments.
- A very cost effective solution of building environments.
- Finished products can be marketed much faster.

The separation of responsibilities between the cloud provider and cloud client is depicted in Table 2.2.



SaaS	PaaS	IaaS
▨ Applications	□ Applications	□ Applications
▨ Data	□ Data	□ Data
▨ Runtime	▨ Runtime	□ Runtime
▨ Middleware	▨ Middleware	□ Middleware
▨ OS	▨ OS	□ OS
▨ Virtualization	▨ Virtualization	▨ Virtualization
▨ Servers	▨ Servers	▨ Servers
▨ Storage	▨ Storage	▨ Storage
▨ Networking	▨ Networking	▨ Networking

▨ Cloud Provider Managed □ Client Managed

Table 2.2: Separation of Cloud Responsibilities

2.1.3 Virtualization related major challenges in Cloud Computing

a. VM Image Sharing

A VM image is utilized for representing various instances of VMs. These can be created or downloaded from a shared image repository. However, the common practice is downloading from a repository which can be a serious threat if used in a malicious way. These codes contained in these images can aid an attacker to identify possible attack points and hence can become an attractive way of injecting malwares in cloud computing environment. Furthermore, an infected VM can result in a confidentiality breach if used to observe data and events of other users or the least can reveal sensitive information of the concerned user.

b. VM Escape

VM escape is a condition in which an attacker escapes from the control of the hypervisor or the virtual machine manager (VMM). It could also be the breaking out of the VM and interacting with the host OS. In this situation, the hacker can access other VMs and in extreme cases bring down the VMM. A successful attack has the capability to extend access to the computing and storage hardware. The most effected cloud service in case of VM escape attack is IaaS service model which in turn can affect other service models.

c. VM Migration

It is the process of moving a VM from one host to another for the purpose of recovery from a host failure, load balancing or maintenance. VM migration is a very critical stage and requires to be conducted in a protected way. During this relocation phase, almost all the contents of a VM are vulnerable and exposed to the network along with the VM code. This situation might lead to security concerns like data confidentiality and integrity. The VM server may be relocated by a hacker to a compromised server or under the control of malicious or compromised VMM or hypervisor [13].

2.2 Intrusion Detection System

2.3.1 Intrusion

A network intrusion can be any undesired or unauthorized activity occurring in a computer network. This is also termed as a network attack. Broadly speaking these attacks can either be active or passive in nature.

Passive attacks affect confidentiality of a network but it does not violate the state of the system. In passive attacks a computer network is eavesdropped, monitored or scanned for open ports and vulnerabilities. These attacks have a very low probability of detection. Active and Passive reconnaissance, both, are housed in Passive attacks. In this case, a hacker does not actively interact with the system to gain desired information about the target system whereas in active reconnaissance an attacker engages with the target system typically conducting a port scan. Passive attacks can be eavesdropping, tapping, war driving, traffic analysis, dumpster diving, social engineering, port scanning etc.

Active attacks violate the confidentiality, integrity and availability of a network thus modifying the target system. In case of active attacks, an attacker or hackers tries to make changes to target data or the data which is enroute to the target system. These attacks can be masquerading attacks, DoS, DDoS, session replay attacks etc [14].

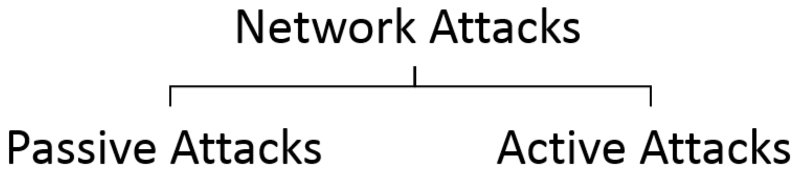


Figure 2.7: Network Attacks

2.3.2 Intrusion Detection System (IDS)

An IDS is a software or hardware device that examines a computing system or network for any malicious activity or other policy breaches. IDSs were originally built for detection of vulnerability exploits against a target application or computer. The intrusions caused to a system or network are possible activities which may compromise confidentiality, integrity or availability as well as to circumvent the security mechanism of a system or a network. These intrusions can be caused both by authorized and unauthorized users of the system or network. These attempts are caused as the attackers who try to gain access for malicious purposes, the authorized users who misuse the assigned privileges or to gain more privileges than authorized. With the advancements in technology the threat spectrum is also on the rise, hence, the demand of IDS has become more than it ever was. Thus, IDS help organizations to safeguard against threats that are allied with the increasing reliance on information systems and network connectivity with big data. The choice, now, is not whether to make use of such systems or not but which type of the system fulfills the demand of a particular organization [3].

IDSs are now considered as a fundamental applications or devices for combatting against cyber-attacks. A few of the compelling reasons to deploy such systems are listed below: -

- A prevention against attacks or abuse of the system
- Detection of attacks and other security violations
- Detection of preambles to attacks
- Easy deployment
- Documentation of existing threat to an organization
- Activation of incident response mechanism
- An aid to quality control for administration and security scheme
- To provide useful information pertaining to intrusions for improved diagnosis, recovery and correction of contributing factors

2.3.3 Classification of Intrusion Detection System

IDS are categorized into two types: -

a. Host-Based Intrusion Detection System

A Host based Intrusion Detection system (HIDS) is a classification of IDS that is concerned with monitoring the security of a single computer from internal and external attacks. It inspects the features of a single host and the events that occur in that particular host for any suspicious activity. Internal attacks may be the situations where detection is specific to check which program access which resource and whether is there any security break whereas, in external attacks HIDS analyses packets to and from that system on its interfaces.

Basically, HIDS performs most probably in the similar manner as many virus scanners. The response mechanism of HIDS is based on logging the activity and generating an alert. These systems are designed to monitor and evaluate network traffic along with different system-specific settings including software calls, local security policy, local log audits, and more. HIDS needs to be installed on each machine and requires configuration to a particular operating system and software [15].

b. **Network-Based Intrusion Detection System**

A NIDS analyzes and evaluates network traffic at every layer of the Open Systems Interconnection (OSI) stack. It also makes decisions about the intended aim of the traffic and analyzes it for untrustworthy activity. An alert is generated on identification of an attack or when an abnormal behavior is sensed. As opposed to HIDS, most NIDSs are deployed on a network and can have the ability to monitor traffic from many systems simultaneously [15].

The NIDS are deployed at a strategic point or multiple points within the network where firewall is located to perform its job of prescribed operation from all devices on the network. It monitors the passing traffic on the entire network against the threats of any attacks. In case any abnormal behavior is sensed or an attack is identified alert is generated. Though NIDS monitor all inbound and outbound traffic which at times might create a bottleneck that would affect the network speed. NIDS can be classified as on-line and off-line. Online NIDS are concerned with dealing the network in real time by analyzing the packets to decide for any malicious behaviors whereas offline NIDS deal with stored data thereby passing it through some processes to decide if it is an attack or not.

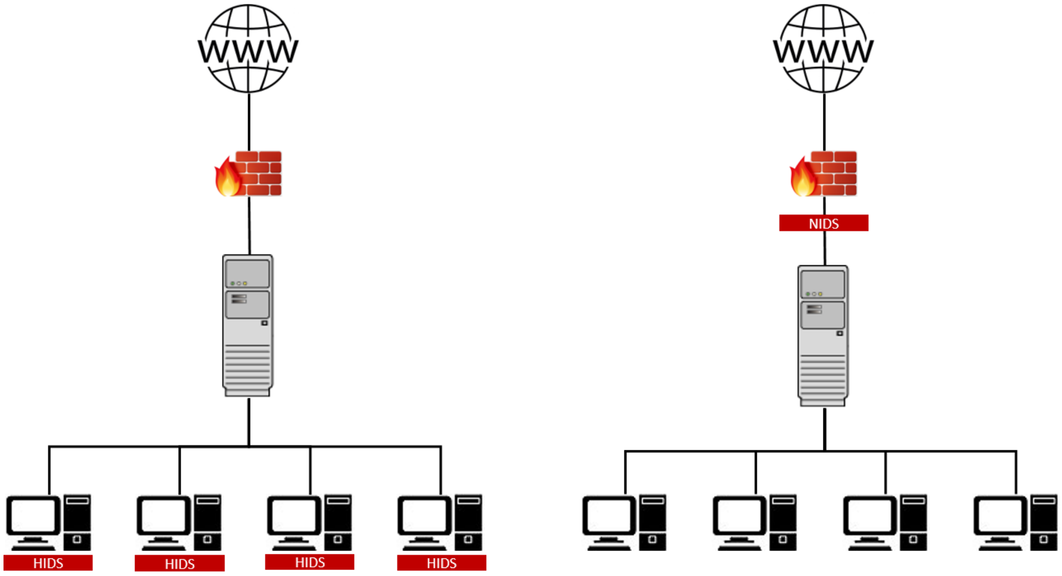


Figure 2.8: Host and Network IDSs

2.3.4 Categories of Intrusion Detection System based on Detection Method

a. Signature Based IDS

Signature based IDS rely upon the database of already known attack signatures, it generates an alarm in case of observing any malicious network activities after cross matching with the held signatures in the database. This type of detection methodology possesses a high rate of detection against known attacks but are not effective against detection of new attacks. Hence, updating the signature database is vital for detecting new attacks [16].

Most IDS are based on the signature based methodology and conduct their operations in the similar manner as that of a virus scanning tool. It searches for a known signature for every event of detection. Though signature based IDSs are very efficient at sniffing known attacks but the whole process depends on receiving regular signature updates, just like an antivirus software, to its optimum performance. In other words, the performance of signature based IDS is good basing on its database of stored signatures.

b. **Anomaly Based IDS**

Anomaly based IDS make use of system and network characteristics to model normal network behaviors. Any deviation from the normal traffic patterns are considered as attacks. Anomaly based IDS uncovers on going intrusive attacks and anomalous activities in information systems. These types of systems work in a dynamically changing environment. The aim remains the same to correctly identify all true attacks while adversely identifying the non-attacks. It therefore, remains an effective approach to determine and reactive to malicious activities at computing and networking assets [16].

Unlike the signature based approach, anomaly based IDS do not rely on predefined signatures for detecting attacks and thus are capable to identify novel attacks. As a drawback, they are characterized by high false positive (FP) rate, in which even, sometimes, legitimate traffic can also be reported as attacks. Moreover, a comprehensive examination is required to find out whether traffic represents a real attack and what it achieves. Therefore, extreme care be exercised in deploying an anomaly based IDS to detect attacks.

2.3.5 Types of Anomaly Based IDS

Anomaly based IDS are categorized into four main type, while each type has its further variants. Figure 2.8 depicts the categories of Anomaly-based IDS. The categorization of Anomaly based NIDS is discussed below: -

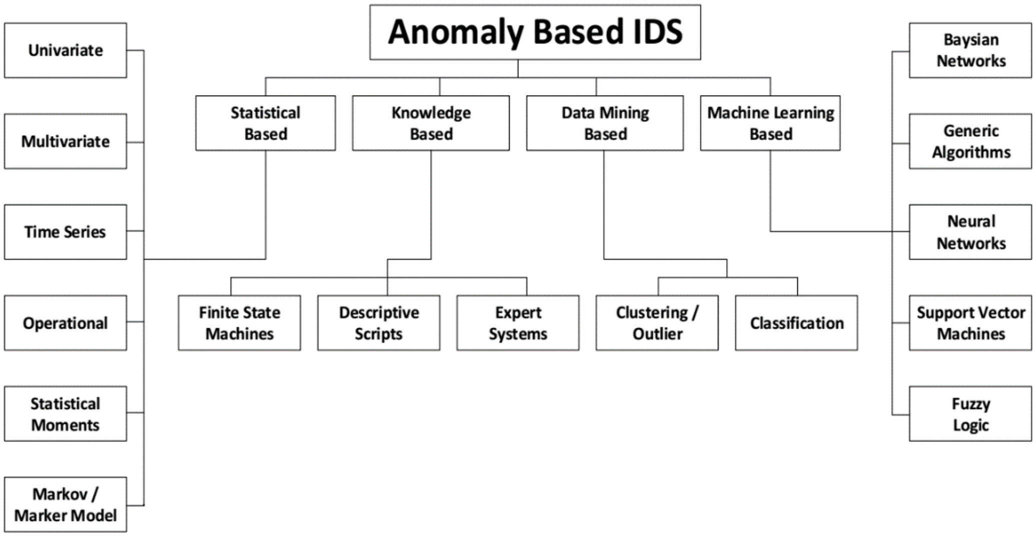


Figure 2.9: Categories of Anomaly-Based IDS

a. **Statistical Based Anomaly IDS Models**

In statistical-based techniques, the profile of the captured network traffic activity is created representing its stochastic behavior. This is a metrics based profile having features such as traffic rate, connections rate, amount of different IP addresses and total number of packets for each protocol, etc. Two datasets of network traffic are considered during the process of detection of anomalies: the first corresponds to the currently observed profile over the passage of time whereas the second is for the previously trained statistical profile. As soon as the network event occurs, the current profile is determined and an anomaly score, depicting the degree of irregularity, is estimated by comparing the two behaviors. The IDS then flags the occurrence of an anomaly when the score transcends a certain threshold.

This technique does not necessitate the previous knowledge of the security weaknesses and thus are quite useful to detect new or zero day attacks but a majority of the statistical anomaly detection techniques need accurate statistical distributions and hence needs the process assumption, that can't be assumed for most of the data processed by anomaly detection systems. The further sub types of statistical based Anomaly IDS are as below: -

- **Univariate Model**

  In univariate statistical analysis, each variable in a data set is separately explored and individually described. It observes the range of values and the central tendency of the values. Further, it describes the pattern of response to the variable on its own. The descriptive statistics describe and summarize the data.

- **Multivariate Model**

  In multivariate statistical approach, analysis is done using statistical process of simultaneously analyzing multiple independent variables with multiple dependent variables using matrix algebra. Multivariate statistical analysis make use of multiple advanced techniques for observing relationship of multiple variables at the same time.

- **Time Series Model**

  The major components of this model are interval timers, event counter and resource measure. The order and interarrival times of the observations along with their values are stored and if the probability of happening of a new observation is too low then it is treated as an anomaly.

- **Operational or Threshold Metric Model**

  Operational or Threshold Metric model is based on the assumption of identification of anomaly by comparing observations with a predefined limit. On the basis of cardinality of observation that is observed over a period of time an alarm is generated. This model is well applicable to metrics where certain values are often associated with intrusions.

- **Statistical Moments Model**

  The moments in this model are statistical mean, standard deviation and any other correlations. If the event falls outside the set interval of the moment, it is referred to be anomalous. In this model, no prior knowledge is required for determining the normal traffic activity and the confidence intervals depends upon the data observed by the users which vary from one user to the other.

- **Markov / Marker Model**

  The Markov or Marker Model is employed with the event counter metric to establish regularity of specific events by comparing with the event that preceded it. Each observation of the network traffic is characterized as a particular state which employs a state transition matrix in order to establish whether the probability of the event is high or normal, based on the preceding events. It is further categorized in two approaches: Markov chain and Hidden Markov models (HMM).

b. **Cognitive or Knowledge Based Anomaly IDS Models**

- **Finite State Machines or Finite Automation**

  A finite state machine (FSM) is a representation of capturing behaviors in states, actions and transitions. A state contains information of the past happening, i.e. whether any change in the input is observed. Based on the state, transition occurs i.e. the state is passed to the next state. An action is a representation of an activity that is to take place at a particular moment. In this type of technique, the intrusion is regarded as a series of action or multiple actions those are carried out by an intruder from an preliminary state of a computing system to a target state.

- **Descriptive Scripts**

  The intrusion detection community provides various proposals for scripting languages describing signatures of attacks on computers as well as networks which are capable of identifying the sequences of specific events that indicate attack patterns. This is an efficient technique which permits a very appropriate implementation and can be applied in various commercial IDS products. But there remains a need for frequent updates the system to detect new vulnerabilities.

- **Expert Systems**

  Expert systems employ a rule based classification i.e. it includes a ruleset describing attacks. The incidents are audited, interpreted into facts and their semantic implication is carried to the expert system and this rule help inference engine for drawing conclusions. With the attachment of semantic it enhances the level of abstraction of audit data.

c. **Data Mining Based Anomaly IDS Models**

- **Clustering / Outlier Detection**

  Clustering is the method of discovering patterns in an unlabeled data. Generally, k-mean clustering is employed for finding natural grouping of similar events. New attacks are represented by the records that are at a long distance from any one of this cluster indicating an unusual activity.

- **Classification**

  In classification, each instance is arranged or ordered by classes or categories as a normal process or an attack. The main goal is to learn from the class-labeled training events to calculate classes of fresh data. This new dataset is classified basing on the training set. The testing phase of classification-based methods are usually fast and thereafter every test case is compared against the pre-computed model.

d. **Machine Learning Based Anomaly IDS Models**

- **Baysian Networks**

  A Baysian network establishes a probabilistic relationship among variables. It is a graphical structure which allows to represent and reason about an uncertain domain. The nodes in this graphical network represent a set of random variables from a particular domain, whereas, the set of directed arcs represent direct dependences between different variables and connect pairs of nodes. Assuming the variables as discrete, the relationship strength among variables is quantified by conditional probability distributions which are associated with each node. It should be noted as a constraint on the arcs that there must not be any directed cycles i.e. by simply following the directed arcs once cannot return to a node. The networks are very well suited to anomaly detention due to the handling of high dimensional data which is difficult for humans to interpret.

- **Genetic Algorithms**

  Genetic algorithm (GA) is a branch of evolutionary biology and natural selection inspired computational models which convert the problem in a specific domain into a chromosome like data structure model and evolve the chromosomes using selection, regrouping and mutation operators. The process of a genetic algorithm normally starts with a population of chromosomes which is selected randomly. These chromosomes represent the problem which needs to be solved. The different positions, referred to as genes, of every chromosome is encoded as bits, characters or numbers as per the attributes of the problem, various positions of each chromosome. These genes are randomly changed within a range during the process of evolution. These GAs can be employed to evolve simple rules for network traffic in order to differentiate normal network events from anomalous connections.

- **Neural Networks**

  Artificial neural networks (ANN) are multilayered perception based technology where a biased weighted sum of the inputs are performed by each unit in order to pass this activation level through a transfer function to produce their output. These are inspired by the observation from human nervous system with the built in complex webs of interconnected neurons. The connection that exist among any two units having some weight and observation required to gauge that how much a unit will affect the other one. ANNs possess the capability to generalize from past behavior to detect novel attacks and hence an effective tool to determine anomalous traffic patterns in a network traffic.

- **Support Vector Machines**

  Support Vector Machines (SVM) are used to map the input vector into a relatively higher dimensional feature space and consequently acquire the optimal separating hyper-plane in a higher dimensional feature space. The support vectors determine the separating hyperplane instead of the whole training samples. Tainting samples depicts the support vectors which are closer to a decision boundary. SVMs enlists itself in category of supervised learning models of machine learning, it has its related learning algorithms which analyzes the used data to classify and performs regression analysis.

- **Fuzzy Logic**

  Fuzzy logic is a methodology in computing which is based on the level of truth instead of the usual true or false Boolean logic. Fuzzy logic includes 0 and 1 as utmost cases of truth but it also involves various stages during the process to depict states of truth producing various attributes of a particular investigation. The system designed on this approach is responsible to handle large quantity of input parameters and deal with the vagueness of the input data. Once this logic is fused with machine learning technique, it minimizes the size of input data set and selects features focusing as an anomaly.

2.3 Machine Learning

Machine Learning can be defined as a process in which computer systems are built in such a way which implement a learning process and automatically improve with experience. It is the subfield of computer science which provides computers the ability to learn without being explicitly programmed. The evolution of machine learning can be traced back from the studies of Pattern Recognition and Artificial Intelligence. It focuses on the development of computer programs which have the ability to teach themselves in order to develop and change when exposed to new data. It does so by exploring the study and construction of algorithms by learning and making predictions on the data [20].

Machine learning is very closely related to computational statistics as well as data mining while having close ties with mathematical optimization. It is a very strong approach to predictive analysis which helps the analysts to produce reliable and repeatable decisions and results while uncovering the hidden insights in the data [21].

2.3.1 Categories of Machine Learning Algorithms

Machine learning algorithms are often categorized in three categories being supervised, unsupervised or semi-supervised learning.

a. Supervised Learning

It is the job of deducing a function from the labeled training data. In supervised learning, the output datasets are provided in such a way that they are able to train the machine so that requisite output is attained. We have input variable (x) which is mapped by a function by using a learning algorithm to yield the output variable (y), such that: -

$$y = f(x)$$

The aim is to estimate the mapping function so well that the output for the next input data can be predicted using the algorithm. The algorithm make predictions on the training data by using a series of iterations which is then corrected by the function. The learning process stops after achieving an acceptable performance level.

Supervised learning is further grouped into two types being Classification or Regression. If the output variable is a category like color being red or blue, or a state like healthy or unhealthy then it is referred to as a classification problem whereas in case the output variable is a real value such as amount of money or weight of a person, it will be referred to regression problem.

b. **Unsupervised Learning**

It the task of deducing a function to describe hidden structure from unlabeled data. In unsupervised learning, the data is clustered into different classes without the requirement of using datasets. In this scenario, the data is available as input data (x) however, corresponding output variables are not available. Here the goal is to model a underlying structure in the data to acquire knowledge or learn more about the data. In this approach, there are no correct answers rather the algorithms are left on the discretion to discover and present structures in the data.

Unsupervised learning is further grouped into two types being Clustering and Association. Clustering problem is used where discovery of inherent grouping in the data is required whereas association problems are concerned to find rules which explain parts of the data.

c. **Semi-Supervised Learning**

It the task of deducing a function to describe hidden structure from a small amount of labeled and a larger portion of unlabeled data. This category falls between the supervised and unsupervised learning. The unlabeled data can be used as training data by utilizing supervised learning problems to make predictions on new data while the unsupervised learning techniques can be used to determine and learn the structure in the input variables.

Summarizing it all, in supervised learning the whole data is labeled and the algorithms learn to predict the output from the input; in unsupervised learning, whole data is unlabeled and the algorithms learn to inherent structure from the input data, finally in semi-supervised learning most of the data is unlabeled while some is labeled therefore, a mixture of supervised and unsupervised techniques are utilized.

CHAPTER 3

ANALYSIS OF MIT DARPA INTRUSION DETECTION DATASET

This chapter will present the analysis carried out on MIT DARPA Dataset. It will describe the tools used for analysis and characteristics of the dataset.

3.1 Introduction

Development of a dataset specific to a particular research may be a favorable choice but using a globally accepted dataset having all the facets of desired behaviors which has been tested by other researchers, remains a better option. This is the same motivation which changed my earlier stance of developing my own dataset for this research. Furthermore, during the course of literature review it revealed that MIT DARPA Intrusion Dataset has its importance and advantages to be used in research related to intrusion detection. The dataset is available free and large enough with variety of network attacks for thorough analysis.

3.2 Development of DARPA Dataset

In 1958 Defense Advanced Research Projects Agency was established as part of US Department of Defense agency which is responsible for research and development of emerging technologies to be used by US military. MIT Lincoln Lab is a part of Department of Defense Research and Development, which conducts research and development to provide solutions to the problems which are critical to national security. The MIT Lincoln Lab along with the US Air Force Research Lab built a computer network to generate a realistic network traffic. The logs of about fifty nodes running different operating systems were captured on daily basis over a prescribed period of time.

3.3 Tools Used for Analysis

Following tools were used for analyzing the dataset: -

3.3.1 Wireshark

Wireshark [22], is a widely used, open-source and free network protocol analyzing tool. It is effectively employs an application which sniffs packets and uses its internal library for capturing of packets. It helps to analyze various activities on the network at a microscopic level. The main features of Wireshark are listed below: -

- Deep inspection of various networking protocols
- Live capture of network packets
- Offline analysis of captured or provided packets
- Options for analyses on a variety of OSs including Windows, Linux, Mac, Solaris etc.
- An easy to understand GUI format.
- Availability of various display filters.
- Option to read and save different file formats.

- Availability of colored rules for easy understanding.

Wireshark provided an excellent platform for analyzing DARPA's raw dataset network data in the form of tcpdump files. It was also helpful to understand the characteristics, protocol and pattern of traffic in the tcpdump with time stamps, IP address and other related features.

3.3.2 TCPTrace

TCPTrace [23] is an open source analysis tool developed by Shawn Ostermann at the Ohio University. It is used for the analysis of tcpdump files, which serves as the input and produces several types of output containing information of the connection. The output parameters may range from elapsed time, number of bytes, sent and received segments, round trip times, retransmissions, window advertisements, throughput and much more. It can also produce a number of graphs for further analysis.

3.4 Layout of DARPA 98 and DARPA 99 Datasets

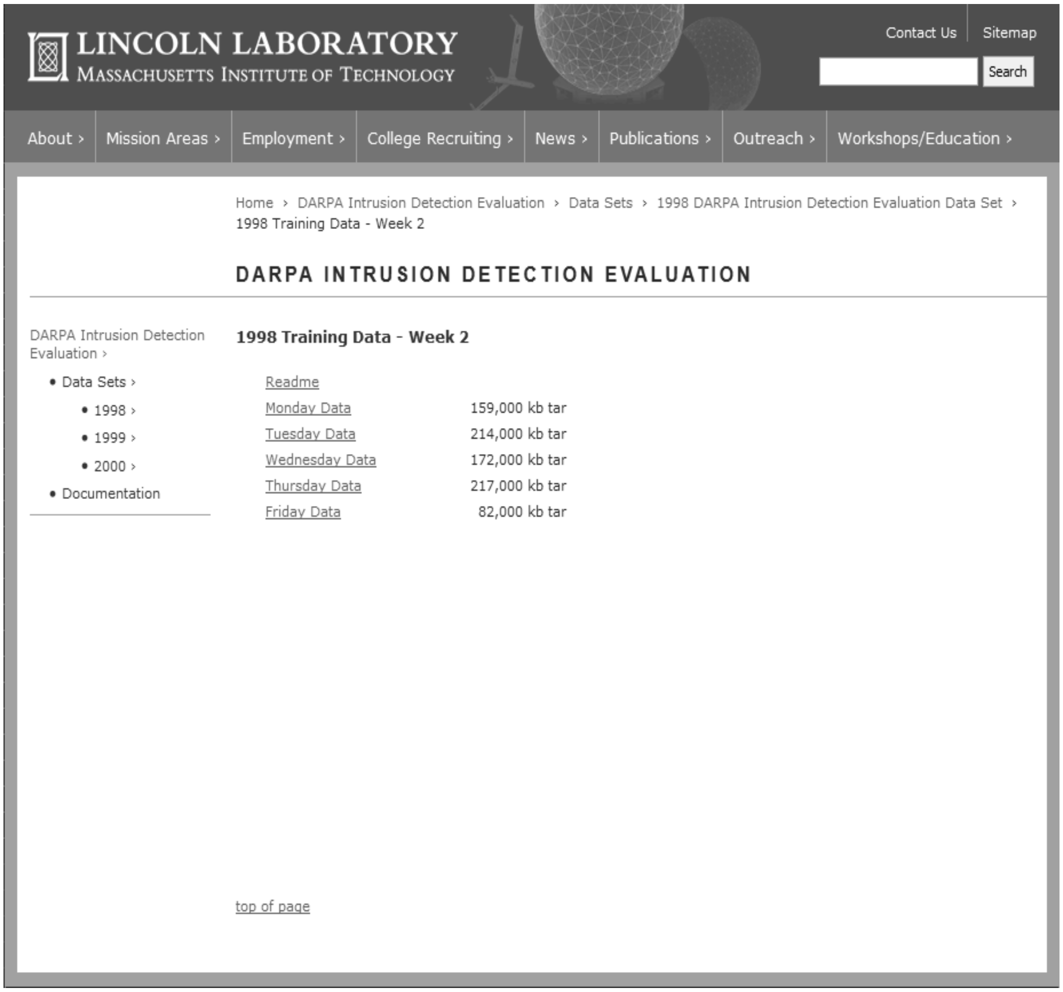
DARPA Intrusion Detection Data Sets are available at [24]. The overview of each dataset is given below: -

3.4.1 1998 DARPA Intrusion Dataset

The 1998 DARPA Intrusion dataset is divided into two parts being offline and real time evaluation. The dataset was collected over a span of nine weeks and distributed in two forms being training data and test data. Training data was simulated and captured for the first seven weeks and distributed over five days of the week i.e from Monday to Friday. Similarly, the test data was captured for the next four weeks for the same days of the week. For the purpose of training network data with various attacks was collected, while two weeks of test data contained some new attacks which were not present in the training data. During the course of data capturing two types of data was collected:

- Tcpdump data
- System audit data to include Sun Basic Security Module (BSM) audit data from one UNIX Solaris host and file system dump

The training data contains about file million records with a total file size of 4 GB whereas the test data contained two million records. Figure 3.1 shows the layout of 1998 DARPA Intrusion Dataset.



Home › DARPA Intrusion Detection Evaluation › Data Sets › 1998 DARPA Intrusion Detection Evaluation Data Set › 1998 Training Data - Week 2

DARPA INTRUSION DETECTION EVALUATION

1998 Training Data - Week 2

Readme	
Monday Data	159,000 kb tar
Tuesday Data	214,000 kb tar
Wednesday Data	172,000 kb tar
Thursday Data	217,000 kb tar
Friday Data	82,000 kb tar

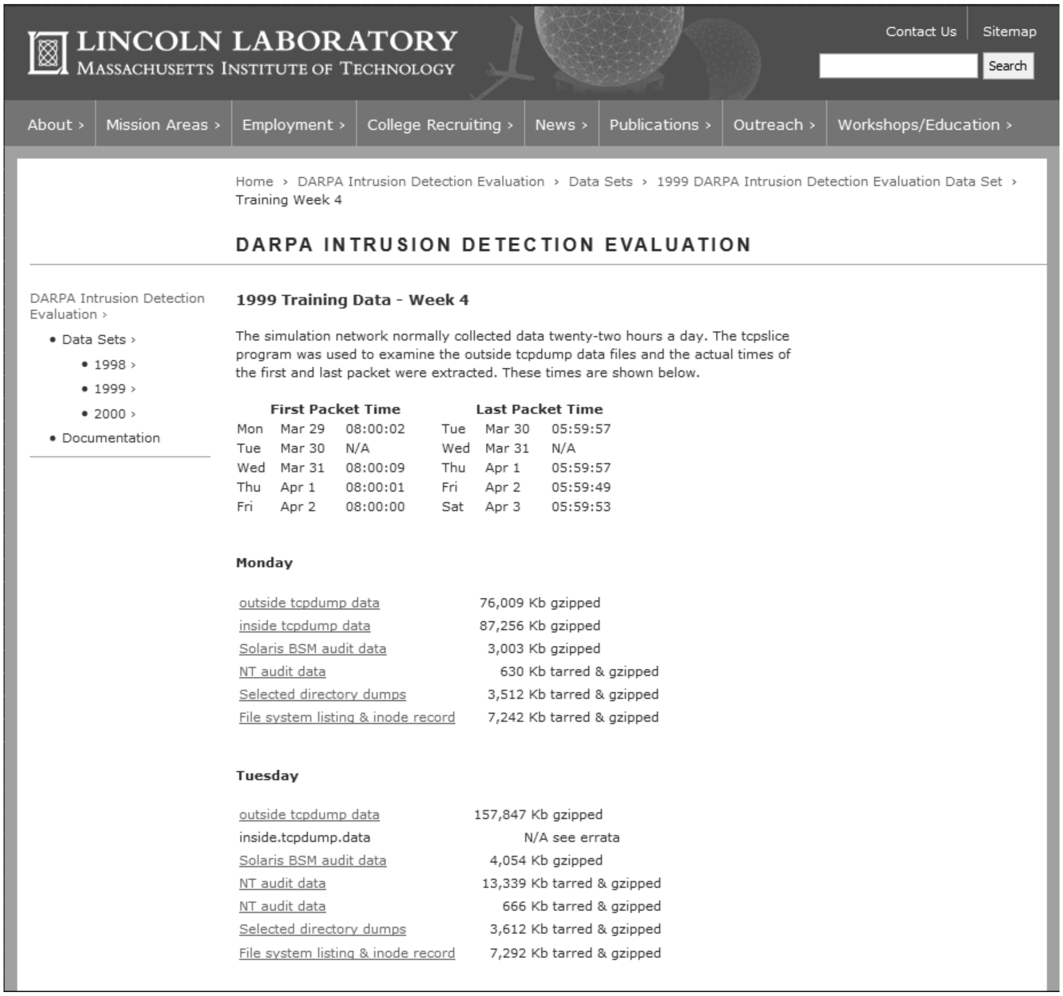
Figure 3.1: Layout of DARPA 1998 Dataset

3.4.2 **1999 DAPRA Intrusion Dataset**

Just like 1998 DARPA Intrusion dataset, 1999 DARPA Intrusion dataset is also divided into two parts being offline and real time evaluation. The dataset was collected over a span of five weeks and distributed in two forms being training data and test data. Training data was simulated and captured for the first three weeks and distributed over five days of the week i.e from Monday to Friday. Similarly, the test data was captured for the next two weeks for the same days of the week. Seven weeks of network based data with various attacks was collected for the purpose of training, while two weeks of test data contained some new attacks which were not present in the training data.

The DARPA 1999 was further improved and it presents a comprehensive view of attack detection and identification. The main goal was to measure the ability of an IDS to detect new attacks which was a major problem found in 1998 evaluations. Another important improvement is the addition of inside attacks.

Each daily tcpdump file contains about 1 million TCP records. The third week dataset contain sixteen tcpdump files comprises of 6.46 GB, whereas the fourth week contain nine tcpdump files of 2.80 GB whereas, the fifth week dataset has ten tcpdump files of 4.85 GB. Figure 3.2 shows the layout of forth week of 1999 DARPA Intrusion Dataset.



Home › DARPA Intrusion Detection Evaluation › Data Sets › 1999 DARPA Intrusion Detection Evaluation Data Set › Training Week 4

DARPA INTRUSION DETECTION EVALUATION

1999 Training Data - Week 4

The simulation network normally collected data twenty-two hours a day. The tcpslice program was used to examine the outside tcpdump data files and the actual times of the first and last packet were extracted. These times are shown below.

First Packet Time			Last Packet Time		
Mon	Mar 29	08:00:02	Tue	Mar 30	05:59:57
Tue	Mar 30	N/A	Wed	Mar 31	N/A
Wed	Mar 31	08:00:09	Thu	Apr 1	05:59:57
Thu	Apr 1	08:00:01	Fri	Apr 2	05:59:49
Fri	Apr 2	08:00:00	Sat	Apr 3	05:59:53

Monday

outside tcpdump data	76,009 Kb gzipped
inside tcpdump data	87,256 Kb gzipped
Solaris BSM audit data	3,003 Kb gzipped
NT audit data	630 Kb tarred & gzipped
Selected directory dumps	3,512 Kb tarred & gzipped
File system listing & inode record	7,242 Kb tarred & gzipped

Tuesday

outside tcpdump data	157,847 Kb gzipped
inside.tcpdump.data	N/A see errata
Solaris BSM audit data	4,054 Kb gzipped
NT audit data	13,339 Kb tarred & gzipped
NT audit data	666 Kb tarred & gzipped
Selected directory dumps	3,612 Kb tarred & gzipped
File system listing & inode record	7,292 Kb tarred & gzipped

Figure 3.2: Layout of DARPA 1999 Dataset

3.5 Characteristics of DARPA Dataset

The main characteristics of the simulated DARPA dataset are: -

- It is a simulation of a real network traffic.
- Attacks were injected using software tools.
- The connection and file system information was captured.
- Traffic from certain points in the network was aggregated.

3.6 Main Services used in DARPA Dataset

The main services used in the simulated network are listed below: -

- Hyper Text Transfer Protocol (HTTP)
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- Simple Network Management Protocol (SNMP)
- Post Office Protocol (POP)
- Domain Name System (DNS)
- X Windows System Protocol
- Internet Relay Chat Protocol (IRC)
- Telnet
- Certain other frequently used services in a network

3.7 List and Description of Attacks in DARPA Dataset

Table 3.1 shows the list and brief description of the attacks that were injected in the simulated network using the software tools: -

Ser	Attack Type	Purpose
1.	Denial of Service (DoS) attack	It is a form of attack on a network where the attacker denies the resources to the legitimate user by flooding the network with undesired traffic.

2.	User to Root (U2R) attack	Type of exploit in which an attacker can exploit vulnerabilities of the system to get the administrative or root privileges of the system by logging as a normal user.
3.	Remote to Local (R2L) attack	Type of attack in which a remote attacker exploits certain vulnerabilities to get local access to a system or systems within a network.
4.	Probing attack	Type of attack in which an attacker tries to gain information about a system or network to exploit security flaws therein.

Table 3.1: Attack types and their purpose in DARPA Dataset

The complete list of list of attacks in DARPA dataset in shown at Annexure A.

3.8 Features of DARPA Intrusion Dataset

The detailed features of DARPA Intrusion dataset in shown at Annexure B.

CHAPTER 4

PROPOSED MODEL / FRAMEWORK

This chapter will describe the proposed model / framework for Intrusion Detection based on hybrid Machine Learning techniques.

4.1 General Description

The proposed architecture uses machine learning technique i.e Fuzzy Logic hybrid with Genetic Algorithm (GA) approach analyzing network database to detect anomalous behaviors. The proposed fuzzy logic function contributes towards extraction of new rules which will enhance accuracy to achieve better true positives. The GA rule in machine learning has the ability to handle discrete and continuous attributes in the dataset which was the limitation in earlier researches.

The DARPA Intrusion Detection database contains a variety of features. I have prepared this dataset to make use of only the relevant attributes which contain all the appropriate features which are helpful to identify the network traffic along with their peculiar characteristics.

In this approach, each individual value of the continuous attributes in the dataset is translated into linguistic terms i.e low, middle, and high. By doing so each attribute is further divided into three sub attributes in terms of linguistic values, which are then analyzed by the GA at the judgement nodes to calculate the measurements of association rule at the processing end. These rules are then saved in a pool, whenever GA extracts an important rule it is stored with following attributes: -

- Support
- Confidence
- Chi-Squared (χ^2) value
- Fuzzy rule parameter

There may be occasions where the already extracted GA rule may be extracted again which will change χ^2 value and fuzzy rule parameter will be changed. Now, if the new value is higher than the previous value it will be replaced with the associated fuzzy parameter. By doing so the pool will always be updated with new generation of values. Finally, these results are fed to the classifier to get the output.

4.2 **Proposed Architecture**

The proposed architecture is translated in Figure 4.1 below: -

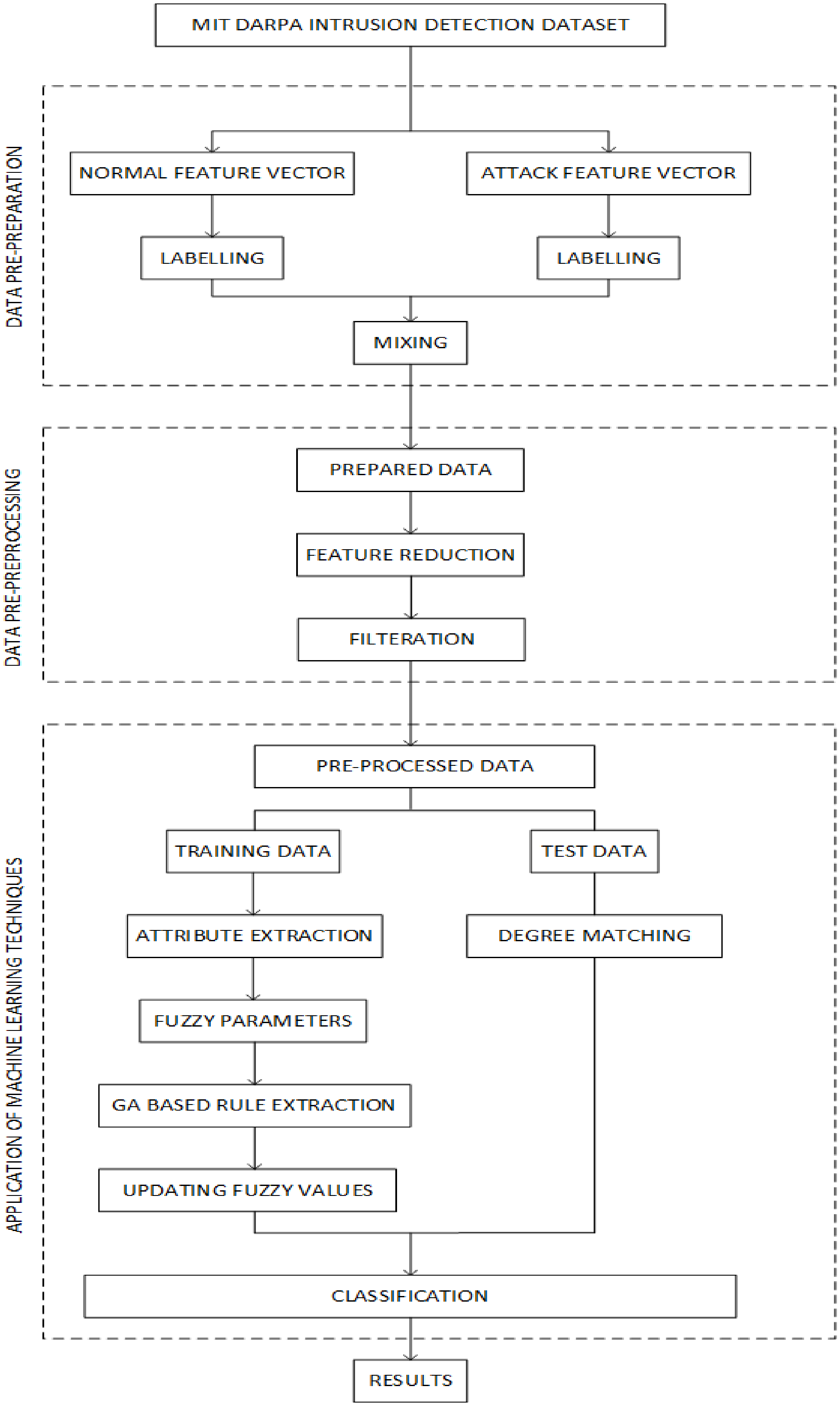


Figure 4.1: Proposed Architecture

4.3 Data Preparation

Table 4.1 and Table 4.2 shows the type of compressed files contained in a set of DARPA 1998 / 1999 Intrusion Detection dataset for a particular day of the week: -

Ser	Type of File
a.	BSM list
b.	Pascal BSM
c.	Pascal praudit
d.	Pascal psmonitor
e.	Tcpdump
f.	Tcpdump List

Table 4.1: Files in DARPA 1998

Ser	Type of File
a.	Outside tcpdump data
b.	Inside tcpdump data
c.	Solaris BSM audit data
d.	NT audit data
e.	Selected directory dumps
f.	File and inode listing

Table 4.2: Files in DARPA 1999

The tcpdumps contained in both DARPA 1998 / 1999 datasets were utilized for the purpose of my research. These dumps were analyzed using analysis tools such as Wireshark etc. The dataset was then loaded to the database to see all available features. The relevant features were earmarked and I have prepared the experimentation dataset to extract 10 attributes containing all the relevant features which are helpful to identify the network traffic along with their peculiar characteristics. These attributes are shown as under: -

- Record ID
- Date of occurrence
- Time of occurrence
- Duration
- Type of service
- Source Port
- No of bytes
- Source IP Address
- Destination IP Address
- Flag
- Type of Attack

4.4 Dataset Input

The program is designed such that it asks the user to input dataset from either DARPA 1998 or DARPA 1999 prepared data. The command is executed in the form of a drop down menu option to choose from. The selected dataset is then displayed in the form of a table where each column represents a particular attribute while a row represents a particular record.

4.5 Data Preprocessing

Two main actions are performed in the Data Preprocessing phase being Feature Reduction and Data Filtration.

4.11.1 Feature Reduction

At this stage the dataset will be read to database and the attributes will be reduced and only those attributes are selected which will be required for further processing. These extracted attributes are those which identify each entry in the network connection. The list is as under: -

- Record ID
- Duration
- Type of Service
- Source Port
- Source IP Address
- Destination IP Address
- Type of Attack

The preprocessor will input the data and eliminate the missing values from the input and make it suitable to be processed by the next module. Furthermore, all non-attacks will be labeled as normal records.

4.11.2 **Data Filtration**

At this stage the reduced features are examined for any incomplete records. These records are filtered out and eliminated from the data to be further processed. Data filtration is a very important stage because if the incomplete records are not separated then they may cause further analysis incomplete and may jeopardize the whole effort.

4.6 **Fuzzy Logic Function**

Fuzzy logic is a degree of truth based form computing in which the truth values of variables may be any real number between 0 and 1, and not just merely being true or false. This logic is contrast to the Boolean logic where the truth values of variables may only be 0 or 1. These logics are employed in this research to deal with the concept of partial truth which varies for being completely true and completely false.

There are certain characteristics in a network connection which contain important information. These are the vital information that signifies a connection over a particular network and cannot be lost. This information can be in the form of discrete or continuous values of attributes. In this research a sub attribute mechanism has been utilized which takes the symbolic, binary and continuous attributes of a connection to preserve it in order to maintain the completeness of the data.

The binary attributes are extracted basing on the judgement function. The symbolic attributes are divided in to different sub attributes whereas, continuous attributes are further divided into three sub-attributes, which are signified by the linguistic values of Fuzzy logic and are pre-defined for every continuous attribute: -

- Low
- Middle
- High

Now the parameter in a Fuzzy rule are assigned as under: -

- $\alpha = 2\beta - \Upsilon$
- β = Average value of continuous attribute
- Υ = Largest value of continuous attribute

4.7 **GA Based Rule Extraction**

I have utilized a class association rule mining algorithm which is based on Genetic Network Programming (GNP). GNP is a further advancement to GA which employs focused graph structures instead of strings and trees. There are three types of nodes in the structure of GNP: -

- Start Node
- Judgement Node
- Processing Node

Figure 4.2 shows basic structure of a GNP.

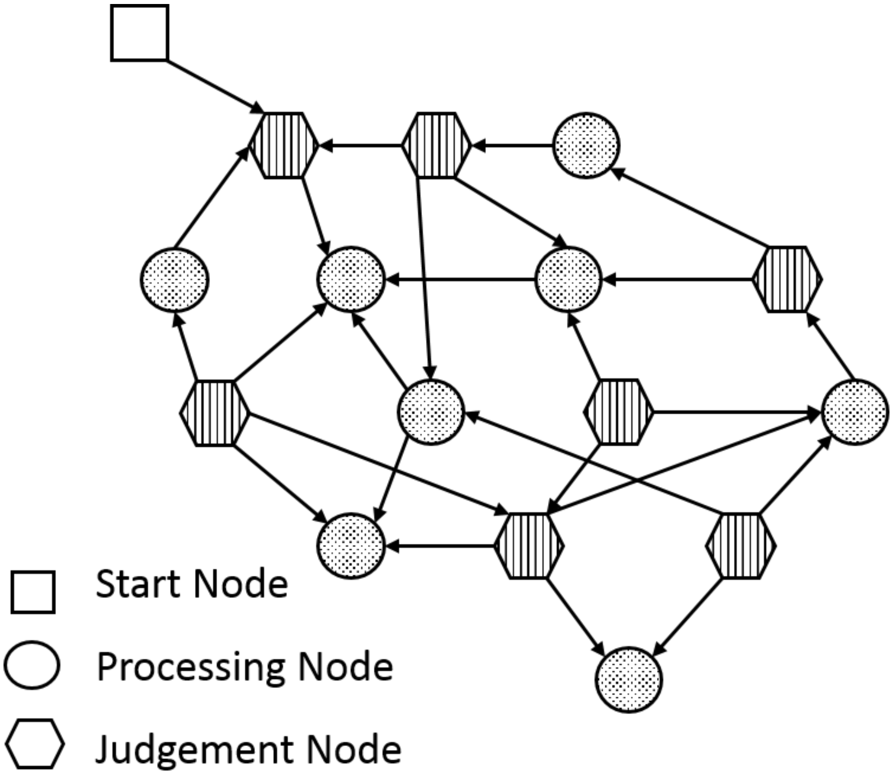


Figure 4.2: Basic Structure of GNP

The start node represents the number of node that is executed at the beginning, a judgment node has branch decision functions whereas Processing nodes decides an action to be executed.

Once GNP architecture is initialed it starts the executing the start node, determines the next node to be executed in accordance with the connection between several nodes and currently activated node’s judgment result. In other words, GNP searches for the attributes of the tuples at the judgement node and calculates the association rule at the processing nodes. The judgement nodes are the prime nodes the calculate the values that are assigned to any sub-attribute such as Land=1 and Protocol=ftp etc. In this approach the class association rule mining technique the sub-attributes are successfully combined in a single rule. There are two possible sides of a judgement node, Yes or No. The Yes side is further connected to another judgement node while the No side connects to the next processing node. Figure 5.3 illustrates this phenomenon.

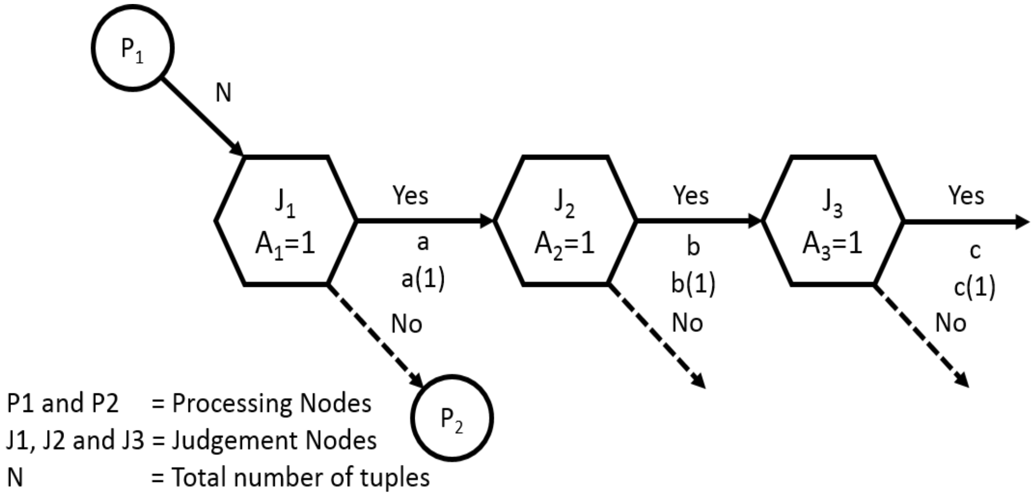

P1
N
J1
A1=1
Yes
a
a(1)
No
J2
A2=1
Yes
b
b(1)
No
J3
A3=1
Yes
c
c(1)
No
P2
P1 and P2 = Processing Nodes
J1, J2 and J3 = Judgement Nodes
N = Total number of tuples

Figure 4.3: Functioning of Nodes

The Binary attributes, here, are further divided into two sub-attributes which corresponds to the judgment functions. As an example, the binary attribute A_1 (Land) is divided into A_{11} (Land= 1) and A_{12} (Land= 0). The symbolic attribute A_2, depicting the type of protocol is further divided into several sub-attributes based on the type of protocols used for the records in the preprocessed data and represented as A_{21} (tcp), A_{22} (http), A_{23} (ftp-data) ……… and so on. Finally, the continuous attributes are distributed into three sub-attributes representing the values of the linguistic terms A_{31} (Low), A_{32} (Medium), A_{33} (High) of the fuzzy rule function which is predefined for every continuous attribute. Table 5.3 shows an example of sub-attribute by the rule extraction function.

Ser	Attribute Type	Sub-Attribute	Detail
1.	Binary Attributes	A_{11}	Land = 1
		A_{12}	Land = 0
2.	Symbolic Attributes	A_{21}	Protocol = telent
		A_{22}	Protocol = ftp-data
		A_{23}	Protocol = smtp
		A_{24}	Protocol = http
3.	Continuous Attributes	A_{31}	Count = Low
		A_{32}	Count = Medium
		A_{33}	Count = High

Table 4.3: Example of sub-attribute utilization in rule extraction

4.8 Calculation of Class Association Rules

The GA based rule extraction using fuzzy logic and class association rule mining effectively chains discrete and continuous values in a single rule by using sub attributes. Whenever a continuous attribute A_i is represented by a judgment node with a linguistic term $L_i \in$ {Low, Middle, High}, the fuzzy value is applied to determine the transition from the current judgment node to the preceding node. Here the fuzzy value is the probability of moving a judgement node to the Yes side of the network. Figure 4.4 represents an example of node transition.

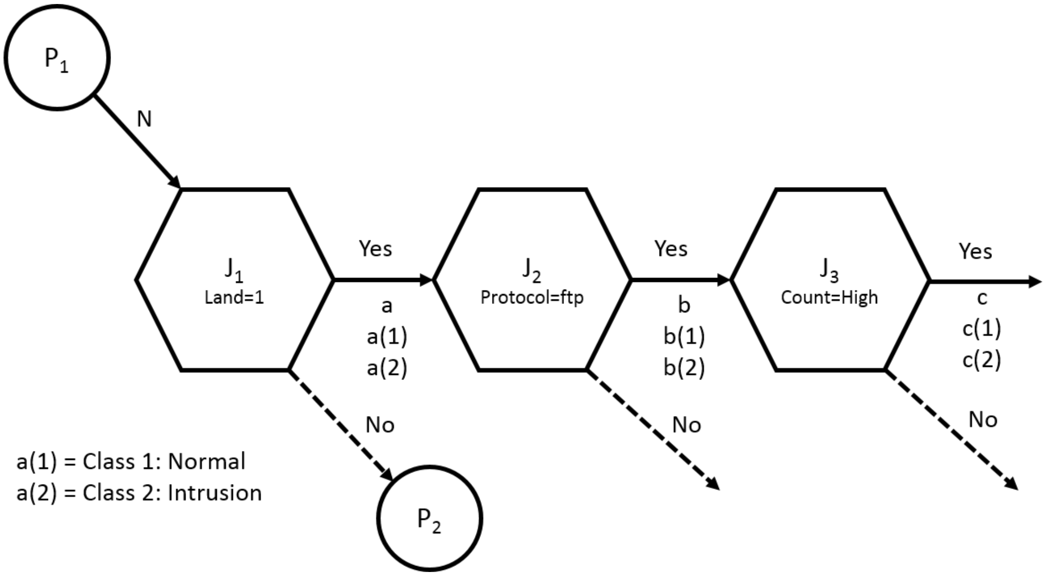


Figure 4.4: Node transition in GA based Rule Extraction

In Figure 4.4, P_1 is a processing node that represents the initializing point of class association rules. P_1 connects with a judgement node J_1 whose Yes side of is connected to another judgment node J_2, while the No side is connected to the next processing node P_2. The Judgment nodes J_1, J_2 and J_3 shown in the above figure performs the function of examining discrete and continuous attributes. The function is shown in Table 4.4.

Ser	Node	Function
1.	Judgement Node J_1	Binary Sub-Attribute are inspected Land = 1
2.	Judgement Node J_2	Symbolic Sub-Attribute are examined Protocol = FTP
3.	Judgement Node J_3	Examines the value of Continuous Sub Attribute Count = High

Table 4.4: Working of Judgement Nodes in Figure 4.4

The total number of tuples in the database that are moving to True side of the judgment nodes (N) is stored in the buffer of the processing node from where rule extraction starts on the basis of node transition. Here in Figure 4.4 the variable a, b and c represents the number of corresponding tuples moving to the True side of each respective judgment nodes. The No side of the judgement node is connected to the next processing node P_2. Hence, the division from the judgment node shows the predecessor part of the class association rules, whereas, the fixed subsequent part can be predefined. This procedure is explained in Table 4.5.

CLASS 1	**CLASS 0**
$A_1 = 1$	$A_1 = 0$
$A_1 = 1$ and $A_2 = 1$	$A_1 = 1$ and $A_2 = 1$
$A_1 = 1$, $A_2 = 1$ and $A_3 = 1$	$A_1 = 1$, $A_2 = 1$ and $A_3 = 1$
Representation of own described Binary and Discrete Sub Attributes	
CLASS 1	**CLASS 0**
Phf=1	Phf=0
Phf=1 Λ Protocol=http	Phf=0 Λ Protocol=http
Phf=1 Λ Protocol=http Λ Count=low	Phf=0 Λ Protocol=http Λ Count=low

Table 4.5: Extraction of Class Association Rules

The number of the judgment functions (J_1, J_2, J_3 …… J_n) are equal to the number of attributes (A_1, A_2, A_3 …… A_n) in the training data. The node processing starts from the processing node P_1 by reading the first tuple, the judgement nodes keep on transferring to the next judgement node till Yes side of the previous judgement node is selected, in case of No side is reached, the node is assigned to the next processing node P_2 and similarly the process is iterated till the node transition started from the last processing node P_n is finished. After exploring first tuple of the training data, the second tuple is read with node transition starting again from processing node P_1. The same process is repeated for all the tuples in the training data. Since the training data used in this research contains both types of connection being normal and with intrusions. Hence, all the tuples are tried out in rule extraction segment within the database. In figure 4.4 a, b, and c are the numbers of tuples moving to Yes side at judgment nodes J_1, J_2, J_3 …… J_n. The GA rule observes all the tuples of the training data and calculates the numbers a, b, c, a(1), b(1), c(1), a(2), b(2), and c(2), where:

- a, b and c are number of tuples progressing to Yes side of judgment nodes.
- a(1), b(1), and c(1) are for Normal Class (C = 1)
- a(2), b(2), and c(2) are for Intrusion Class (C = 2)

Furthermore, Support, Confidence and Chi-Squared value (χ^2). The calculation procedure of support and confidence is described in Table 4.6.

Ser	**Rule**	**Class**	**Support**	**Confidence**
1.	$A_1 = 1$	1	a (1) / N	a (1) / a
2.	$A_1 = 1 \land A_2 = 1$	1	b (1) / N	b (1) / b
2.	$A_1 = 1 \land A_2 = 1 \land A_3 = 1$	1	c (1) / N	c (1) / c

Table 4.6: Calculation of Support and Confidence

Let

- I = A_1, A_2, A_3 …… A_i
- N = Set of Tuples
- T = Set of attributes such that $T \subseteq I$
- X = Set of attributes in I contained in a Tuple T such that $X \subseteq T$
- Support (X) = x,
- support (Y) = y and
- support ($X \land Y$) = z
- An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \emptyset$.

The χ^2 value is calculated as under:

$$\chi^2 = \frac{N\,(z\text{-}xy)^2}{xy\,(1\text{-}x)\,(1\text{-}y)}$$

The definition of important rules is described below: -

- $\chi^2 > \chi^2_{min}$
- support ≥ $support_{min}$
- confidence ≥ $confidence_{min}$

The corresponding values of α, β and ϒ will be updated based on the definition of important rules. Figure 4.5 represents the complete procedure for association rule mining.

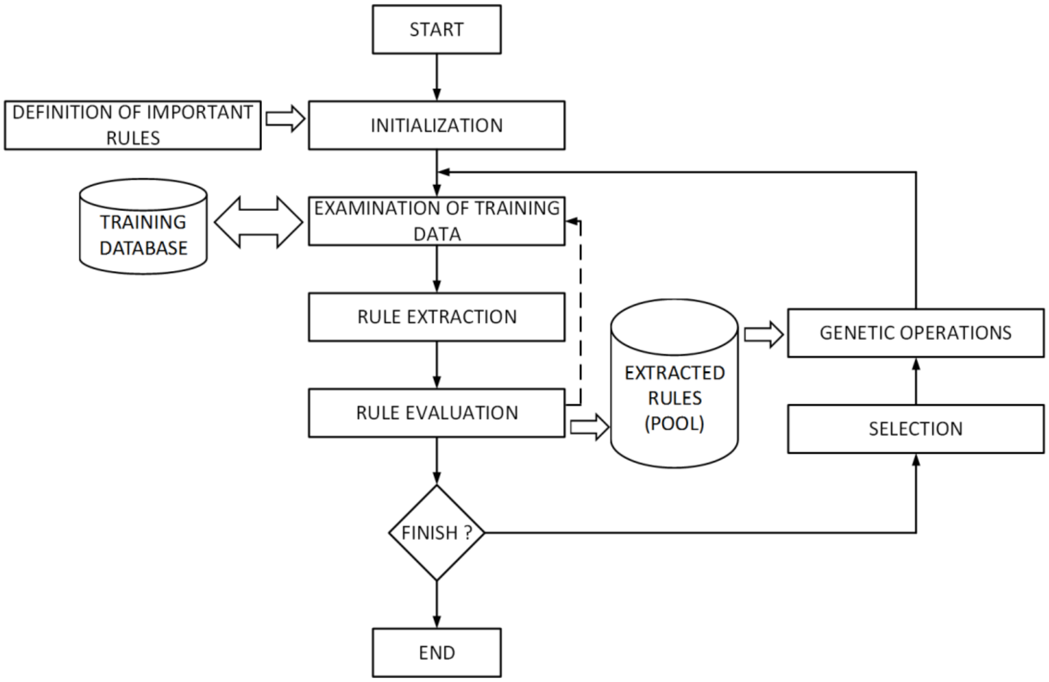


Figure 4.5: Association Rule Mining Procedure

4.9 **Updating Extracted Fuzzy Rules**

Figure 4.5 represents that the extracted fuzzy class association rules are stored in a rule pool with its corresponding support, confidence and χ2 value. Whenever an important rule is obtained and the fuzzy rule has a higher χ2 value, it will replace the respective old fuzzy rule along with its fuzzy parameters. Resultantly, the pool is updated and stored with important fuzzy rules with higher χ^2 values.

4.10 Exposure of Trained Data to Test Data

Test data refers to the data which has never been exposed to the training data during the complete process of machine learning. The test data contains both normal and intrusion records. This data is analyzed with the trained data for the purpose of detecting intrusions from the dataset.

Before the process of classification, the degree of the test data is matched between the continuous attribute A_i and its corresponding linguistic term L_i in a rule r such that $r \in R$ in a class k and the value $a_i \in A_i$ of the test data where $ʃL_i$ represents the membership function for linguistic term L_i: -

$$\text{Match\_Degree}_k (L_i , a_i) = ʃL_i (a_i)$$

Then, the matching between rule r in class k (including C continuous attributes and D discrete attributes) and new unlabeled connection d is defined as: -

$$\text{Match\_Degree}_k (d , r) = \frac{1}{C + D} \left(\sum \text{Match\_Degree}_k (L_i , a_i) + t \right)$$

Where C is the number of Continuous Attributes in rule r, D is the number of Discrete Attributes in rule r, i is index of continuous attributes in rule r and t is the number of matched discrete attributes with new unlabeled connection d in rule r.

Once the degree of test data is matched, in case a new unlabeled connection data arrives, the matching between the new data (d_{new}) and the rules in the normal rule pool is calculated by the above equation. Then, mean (μ) and standard deviation (σ) of $\text{Match}_n(d)$ of all the training data is calculated.

4.11 **Classification**

Finally, the classifier calculates the average matching between the test data and rules stored in the rule pool with all normal and intrusion rules and gives the output result.

- **Classifier for Normal Connection**

  If $\text{Match}_n (d_{new}) \geq (\mu - k\sigma)$, it is labeled as normal.

- **Classifier for Intrusion Connection**

  If $\text{Match}_n (d_{new}) < (\mu - k\sigma)$, the connection is labeled as intrusion.

4.12 **Softwares Requirement**

The softwares used / required for the proposed model are shown at Annexure C.

Ser	Software	Version
1.	Java Development Toolkit	8.0.1110.14
2.	Microsoft Visual C++ 2015 Redistributable	14.0.23026
3.	Wampserver	3.0.6
4.	Apache HTTP Server Project	2.4.23
5.	MySQL Community Server	5.7.14
6.	PHP	7.0.10
7.	Netbeans IDE	8.2
8.	Notepad ++	7.1
9.	Mozilla Firefox	50.1.0

Table 4.7: Software Requirements

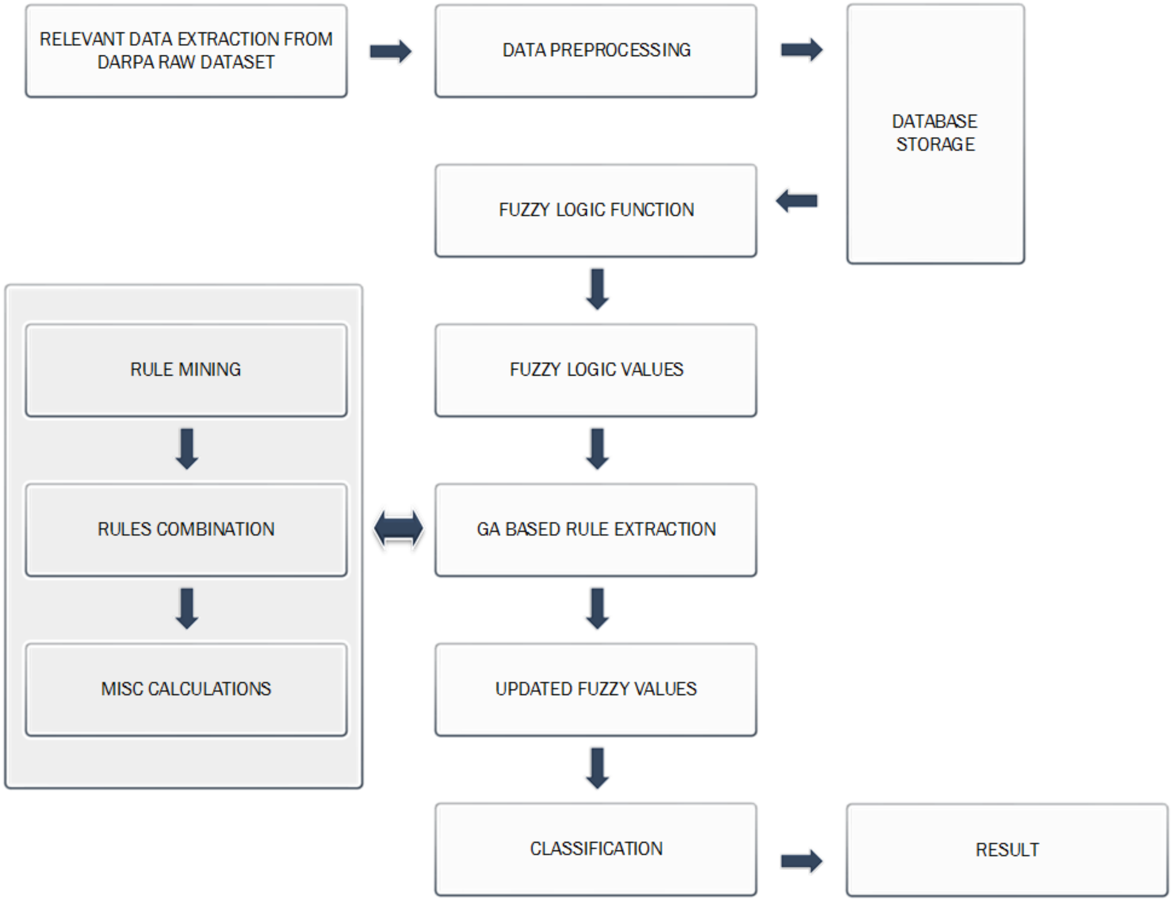


Figure 4.6: Flow Diagram

CHAPTER 5

DEPLOYMENT OF CLOUD TESTBED

This chapter will describe the preparation and development of Cloud Testbed including hardware and software requirements.

5.1 Introduction

During the course of the research it was considered appropriate to test the prototype on a testbed so that the efficacy of the model be analyzed. The cloud environment was built to test the response of different versions of Windows based OS. Furthermore, the effect of network data in a VM environment was required to be tested for intrusion detection. The details of development are covered in succeeding paragraphs.

5.2 Server Specifications

The specifications of the server used for developing of Cloud Testbed is shown in Table 5.1 below: -

Ser	Type		Detail
1.	Server name		Sun Fire X4450
2.	Processor		Quad-Core Intel Xeon processor 7300 series
3.	Memory		42 GB DDR 2 (extendible up to 128 GB)
4.	Ethernet Ports		4, 10 / 100 / 1000 Mbps
5.	Internal Storage		4 x 146 GB SAS disk drives
6.	Removeable Media		1 x DVD slimline drive
7.	USB Ports		5 x 2.0 USB Ports
8.	Service Ports		
	a.	Serial Management	1 x RJ 45
	b.	Network Management	1 x 10 MB
	c.	VGA Video	1 x HD 15 VGA
9.	Power Requirement		100 – 240 V AC, 50 – 60 Hz
10.	Remote Management		Onboard embedded Integrated Lights Out Manager (ILOM)
11.	OS		Sun Solaris

Table 5.1: Technical Specifications of Cloud Testbed

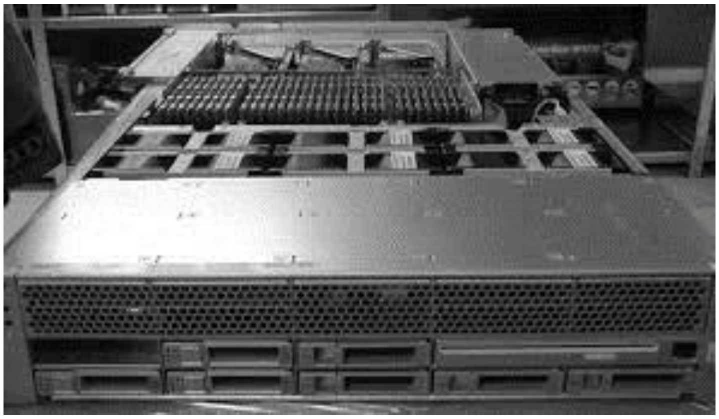


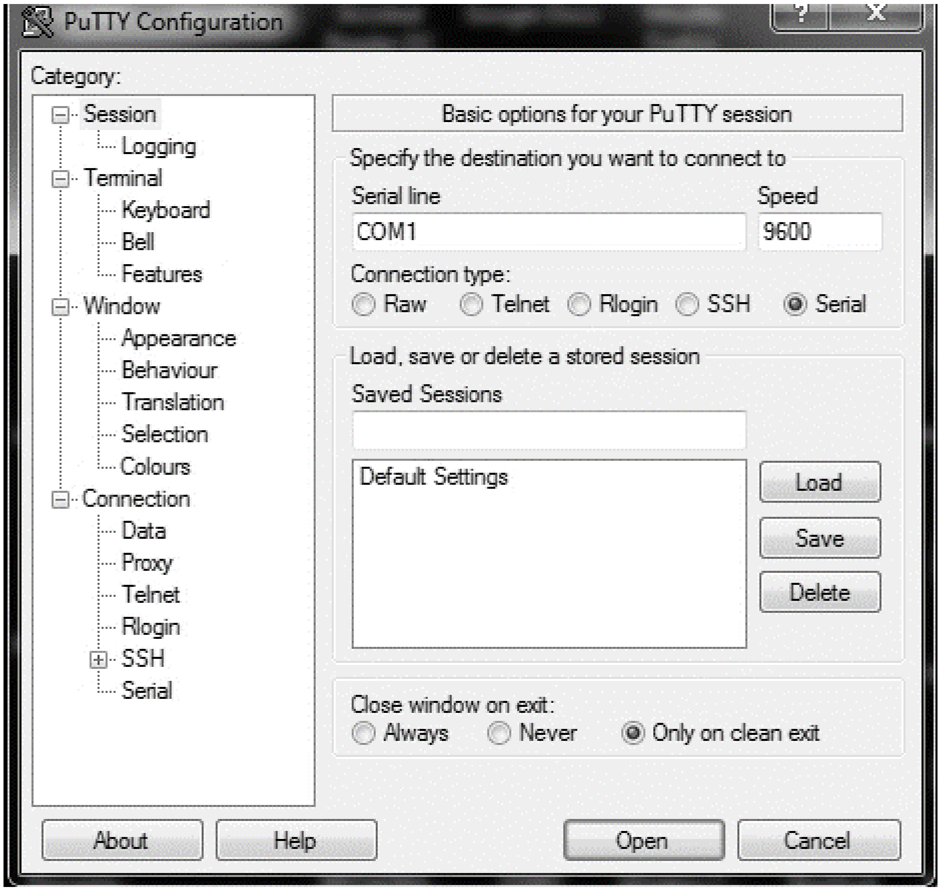
Figure 5.1: Sun Fire X4450 Server

5.3 Management Terminal

A separate laptop, with serial connection port, was utilized for terminal management of Sun Fire X4450. This terminal provided support for assigning a static IP to the cloud server.

5.4 Terminal Emulation

The terminal emulation, serial console and network file transfer mechanisms were achieved on Sun Fire X4450 using 'Putty'. The ability of connecting to the serial port was utilized to make connection with Management Terminal.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM1

Speed: 9600

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 5.2: PuTTY Configuration Console

5.5 Connection to ILOMS

The terminal PC was connected to the Cloud server using a serial connection, initiating from the serial port of terminal PC to serial management port at the server. Then the server was booted as access obtained on the terminal PC through Mozilla Firefox browser using access credentials.

5.6 Configuring Static IP Address

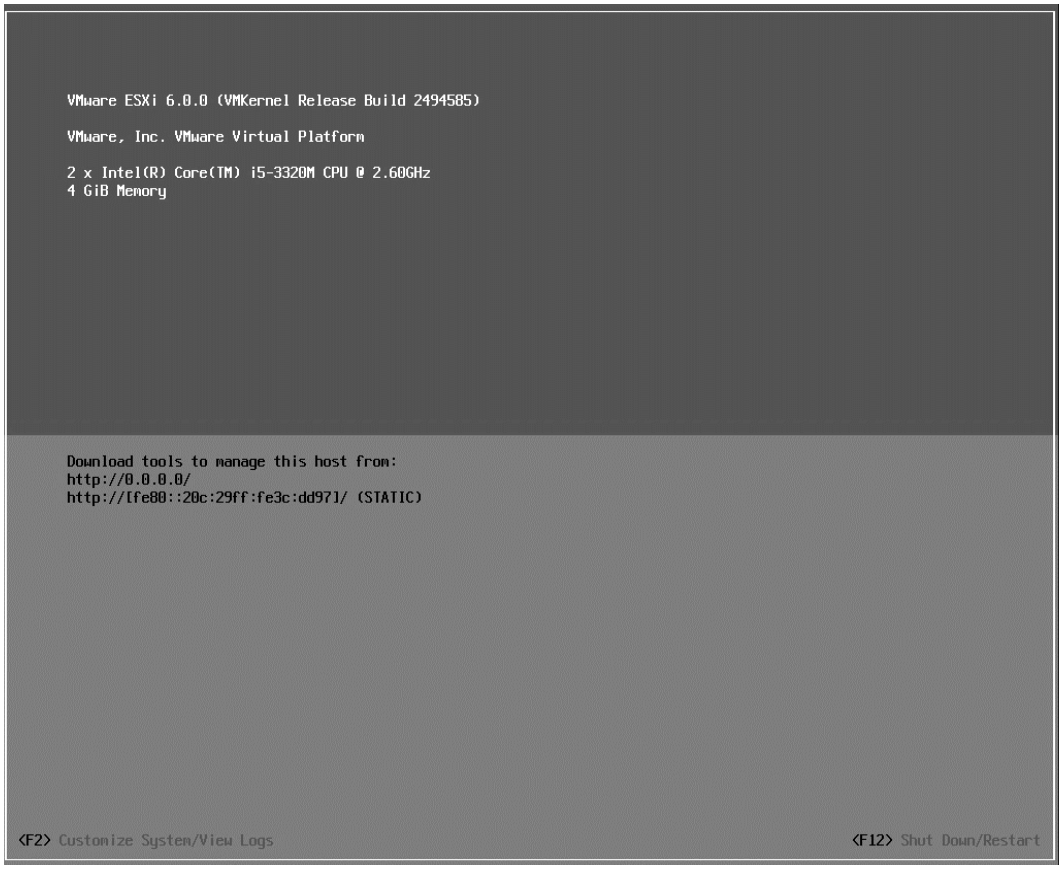
After successful serial connection, the working directory was set and the server was assigned a static IP address.

5.7 Creation of User Account

A user was created in the ILOM and provided another static IP for the hypervisor.

5.8 Configuring Hypervisor on Cloud Server

‘VMWare VSphere ESXI’ was selected as hypervisor for the cloud server and was installed by establishing a serial connection, initiating from serial port of client machine and terminating on Management Terminal port at the host side.



```
VMware ESXi 6.0.0 (VMKernel Release Build 2494585)

VMware, Inc. VMware Virtual Platform

2 x Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz
4 GiB Memory


Download tools to manage this host from:
http://0.0.0.0/
http://[fe80::20c:29ff:fe3c:dd97]/ (STATIC)
```

Figure 5.3: Installation Activity of VMWare ESXi 6.0

5.9 ClientSide Access to the Hypervisor

‘VMWare VSphere Client’ was used to access the hypervisor on the host machine through Mozilla Firefox browser and using static IP for the hypervisor.

5.10 Creation of VMs

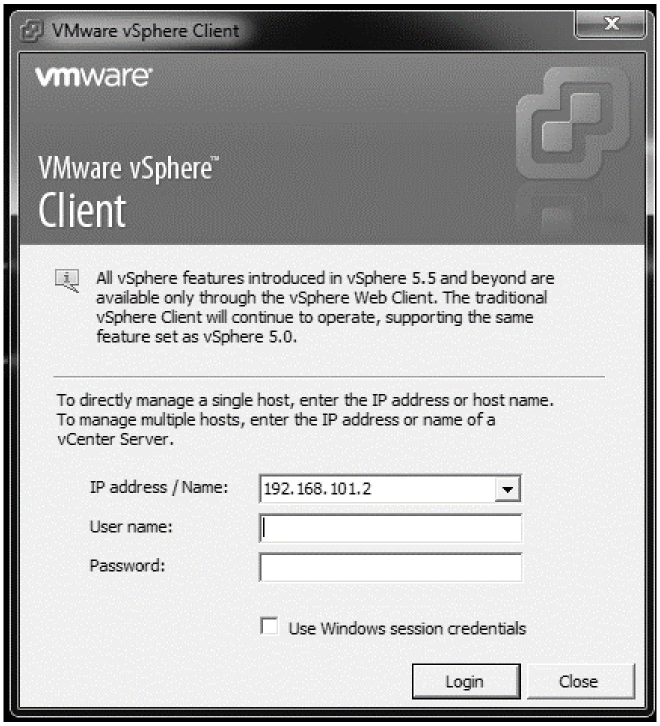
Different VMs were created on the cloud server to depict services offered by the Cloud server to the users using the hypervisor. Table 5.2 shows the created VMs representing various roles / Cloud services: -

Ser	VM	OS	Role / Cloud Service
1.	VM 1	Windows Server 2012	Active Directory, DB and Mail Server
2.	VM 2	Windows 7 Ultimate	SaaS
3.	VM 3	Windows 7	PaaS
4.	VM 4	Windows 8.1	SaaS
5.	VM 5	Windows 10	SaaS
6.	VM 6	Windows 10	PaaS

Table 5.2: VMs on Cloud Testbed

5.11 **Client Access to Cloud Server**

The cloud environment was provided basing on the model of Private Cloud and was not assigned public access. The cloud users were provided access depending upon the level of access wirelessly to the cloud using ‘VMWare VSphere Client’ from where they accessed the cloud by logging on to the respective VM using specified IP address.



All vSphere features introduced in vSphere 5.5 and beyond are available only through the vSphere Web Client. The traditional vSphere Client will continue to operate, supporting the same feature set as vSphere 5.0.

To directly manage a single host, enter the IP address or host name. To manage multiple hosts, enter the IP address or name of a vCenter Server.

IP address / Name: 192.168.101.2

User name:

Password:

□ Use Windows session credentials

Figure 5.4: VMWare VSphere Client Login Terminal

CHAPTER 6

TEST RESULTS

This chapter will show the results of the experimentation using test data and respective graphs.

6.1 Test Data

The term ‘Test Data’ refers to that subset of MIT DARPA Intrusion Dataset which was used to test the proposed model. This dataset contained both normal and attack vectors. The training dataset has not been exposed to this dataset during the course of entire process. The main objective behind is to input the test data and get results.

6.2 Performance Matrices of Classification

The performance metrics is an indication to gauge the performance of the IDS. The chosen criteria will evaluate the results after exposing the system to test data. And the metrics will represent the precision and accuracy of the classifier. Following parameters will be used for evaluation criteria: -

- **True Positives**: Number of intrusions detected correctly by the classifier.
- **False Positives**: Number of normal connections detected as intrusion by the classifier.
- **True Negatives**: Number of normal connections detected correctly by the classifier.
- **False Negatives**: Number of intrusions detected as normal by the classifier.

6.2.1 Confusion Metrics

Prediction Results			
	Intrusion	**Normal**	**Total**
Intrusions	TP	FN	TP + FN
Normal	FP	TN	FP + TN
Total	TP + FP	FN + TN	TP + FN + FP + TN

Table 6.1: Confusion Metrics

6.2.2 Judgement Parameters

- **True Positive Rate (TPR)**: This will define how precise the system is in detection of intrusions. TPR is derived from the following formula:

$$TPR = \frac{TP}{TP + FN}$$

- **False Positive Rate (FPR)**: FPR is the rate of detecting normal connections as intrusions from the total normal connections. FPR is derived from the following formula:

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

- **Accuracy**: It is the percentage of correctly identified intrusions from the total number of connections. Accuracy is derived from the following formula:

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}$$

6.3 System Evaluation with DARPA 1998 as Training Dataset

6.3.1 Confusion Metrics

Prediction Results			
	Intrusion	**Normal**	**Total**
Intrusions	80	32	112
Normal	36	128	164
Total	116	160	276

Table 6.2: Confusion Metrics for DARPA 1998

6.3.2 Evaluation Parameters

- **TPR**

  TPR = 80 / (80+32) = 0.71

- **FPR**

  FPR = 36 / (128+36) = 0.21

- **Accuracy**

  Accuracy = (80+128) / (80+36+128+32) = 0.75

6.4 System Evaluation with DARPA 1999 as Training Dataset

6.4.1 Confusion Metrics

Prediction Results			
	Intrusion	**Normal**	**Total**
Intrusions	84	28	112
Normal	48	116	164
Total	132	144	276

Table 6.3: Confusion Metrics for DARPA 1999

6.4.2 Evaluation Parameters

- **TPR**

  TPR = 84 / (84+28) = 0.75

- **FPR**

  FPR = 48 / (116+48) = 0.29

- **Accuracy**

  Accuracy = (116+48) / (84+28+116+48) = 0.75

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

Intrusion detection is improving with time and this enhancement is a continuous process because of the advancements in the technology which, at one end facilitated the users but on the other, opened a vast door of opportunities for the intruders to improvise their malicious intents.

It was a challenging task for me to provide a tool which practically demonstrate the concept of theoretical logics which exist around us but many a times are difficult to exhibit their efficacy. The most important stage of the complete research remained data preparation and choosing relevant attributes from the raw dataset. This feature selection phase looked simple in the beginning but later proved to be a key factor for conduct of whole research. The use of

Machine Learning techniques in the field of Information security and dealing with Big Data scenarios has really paved the path to reduce human efforts in interaction with the complex nature of computing and for the acquisition of improved results. The development stages of the code provided an opportunity to look deep down in the mechanism of Machine Learning along with its efficacy of dealing multifaceted nature of data attributes.

MIT DARPA Intrusion Detection Dataset has its effectiveness and viability to this extent of time that it is being used by researchers across the world. However, with the advancements in networking protocols and attack vectors there is a dire need for a comprehensive dataset be made on the lines of DARPA Dataset to include the latest normal and attack characteristics of modern networking. The developed framework is flexible enough to incorporate the characteristics of any other dataset prepared on the lines of input matrices encompassing latest changes.

Annexure A

Complete List of Attacks in DARPA Intrusion Dataset

Ser	Attack Name	Type	DARPA 98	DARPA 99
1	Apache 2	DoS Attack	Yes	Yes
2	ARP poison	DoS Attack	No	Yes
3	Back	DoS Attack	Yes	Yes
4	Crashils	DoS Attack	Yes	Yes
5	DOS Nuke	DoS Attack	No	Yes
6	Land	DoS Attack	Yes	Yes
7	Mail Bomb	DoS Attack	Yes	Yes
8	Neptune	DoS Attack	Yes	Yes
9	Ping of Death	DoS Attack	Yes	Yes
10	Process Table	DoS Attack	Yes	Yes
11	Self Ping	DoS Attack	No	Yes
12	Smurf	DoS Attack	Yes	Yes
13	SSH Process Table	DoS Attack	No	Yes
14	Syslogd	DoS Attack	Yes	Yes
15	TCP Reset	DoS Attack	No	Yes
16	Teardrop	DoS Attack	Yes	Yes
17	UDP Storm	DoS Attack	Yes	Yes
18	Anypw	U2R Attack	No	Yes
19	Casesen	U2R Attack	No	Yes
20	Eject	U2R Attack	Yes	Yes
21	Ffb Config	U2R Attack	Yes	Yes
22	Fd Format	U2R Attack	Yes	Yes
23	Load Module	U2R Attack	Yes	Yes

24	Ntfs DoS	U2R Attack	No	Yes
25	Perl	U2R Attack	Yes	Yes
26	Ps	U2R Attack	Yes	Yes
27	Sechole	U2R Attack	No	Yes
28	X Term	U2R Attack	Yes	Yes
29	Yaga	U2R Attack	No	Yes
30	Dictionary	R2L Attack	Yes	Yes
31	FTP Write	R2L Attack	Yes	Yes
32	Guest	R2L Attack	Yes	Yes
33	Http Tunnel	R2L Attack	Yes	Yes
34	Imap	R2L Attack	Yes	Yes
35	Named	R2L Attack	Yes	Yes
36	Nc FTP	R2L Attack	No	Yes
37	Netbus	R2L Attack	No	Yes
38	Netcat	R2L Attack	No	Yes
39	Phf	R2L Attack	Yes	Yes
40	Ppmacro	R2L Attack	No	Yes
41	Send Mail	R2L Attack	Yes	Yes
42	SSH Trojan	R2L Attack	No	Yes
43	X Lock	R2L Attack	Yes	Yes
44	X Snoop	R2L Attack	Yes	Yes
45	Inside Sniffer	Probing	No	Yes
46	IP Sweep	Probing	Yes	Yes
47	Ls Domain	Probing	No	Yes
48	M Scan	Probing	Yes	Yes
49	NT Info Scan	Probing	Yes	Yes
50	Nmap	Probing	Yes	Yes
51	Queso	Probing	No	Yes

52	Reset Scan	Probing	No	Yes
53	Saint	Probing	Yes	Yes
54	Satan	Probing	Yes	Yes

Annexure B

Features of 1998 & 1999 DARPA Intrusion Dataset

Ser	Name	Category	Detail
1	Duration	Individual Connection	Time of connection
2	Protocol Type	Individual Connection	Protocol type
3	Service	Individual Connection	Destination network service
4	Flag	Individual Connection	Normal / Error status of connection
5	Source Bytes	Individual Connection	No of data bytes, source - destination
6	Destination Bytes	Individual Connection	No of data bytes, destination - source
7	Land	Individual Connection	1=Connection from/to same host/port, 0=Otherwise
8	Wrong Fragment	Individual Connection	No of wrong fragments
9	Urgent	Individual Connection	No of urgent packets
10	Count	Traffic features with 2s windows	No of connections to same host as current

11	SYN Error Rate	Traffic features with 2s windows	Percentage of connections with SYN errors
12	REJ Error Rate	Traffic features with 2s windows	Percentage of connections with REJ errors
13	Same Service Rate	Traffic features with 2s windows	Percentage of connections to same services
14	Different Service Rate	Traffic features with 2s windows	Percentage of connections to different services
15	Service Count	Traffic features with 2s windows	No of connections to same service as current
16	Service SYN Error Rate	Traffic features with 2s windows	Percentage of connections with SYN error
17	Service REJ Error Rate	Traffic features with 2s windows	Percentage of connections with REJ error
18	Service Different Host Rate	Traffic features with 2s windows	Percentage of connections to different hosts
19	Destination Host Count	Traffic connection with 100 connection window	No of connections to same host as current
20	Destination Host SYN Error Rate	Traffic connection with 100 connection window	Percentage of connections with SYN errors
21	Destination Host REJ Error Rate	Traffic connection with 100 connection window	Percentage of connections with REJ errors
22	Destination Host Same Service Rate	Traffic connection with 100 connection window	Percentage of connections to different services
23	Destination Host Different Service Rate	Traffic connection with 100 connection window	Percentage of connections to different services

24	Destination Host Service Count	Traffic connection with 100 connection window	No of connections to same service as current
25	Destination Host Service SYN Error Rate	Traffic connection with 100 connection window	No of connections with SYN errors
26	Destination Host Service REJ Error Rate	Traffic connection with 100 connection window	No of connections with REJ errors
27	Destination Host Service Different Host Rate	Traffic connection with 100 connection window	Percentage of connections to different hosts
28	Destination Host Same Service Port Rate	Traffic connection with 100 connection window	Percentage of connections to same service port
29	Hot	Content Features	No of hot indicators
30	Number Failed Logins	Content Features	No of failed login attempts
31	Logged In	Content Features	1=Logged In, 0=Otherwise
32	Number Compromised	Content Features	No of compromised attempts
33	Root Shell	Content Features	1=Root Shell obtained, 0=Otherwise
34	SU Attempted	Content Features	1=SU Root, 0=Otherwise
35	Number Root	Content Features	No of Root access
36	Number File Creations	Content Features	No of file creation operations
37	Number Shells	Content Features	No of shell prompts
38	Number Access Files	Content Features	No of operations on access control files

39	Number Outbound Commands	Content Features	
40	Is Hot Login	Content Features	1=Login in Hot List, 0=Others
41	Is Guest Login	Content Features	1=Guest, 0=Others
42	Class	Attack Type / Normal	Normal or Attack traffic

[1] Peter Mell and Timothy Grance. “The NIST Definition of Cloud Computing”. NIST Special Publication 800-145, 2011.

[2] Md Tarique Prwez and Kakali Chatterjee. “A framework for Network Intrusion Detection in Cloud”. In 6th International Conference on Advanced Computing, 2016.

[3] Rebecca Bace and Peter Mell. “Intrusion Detection Systems”. NIST Special Publication 800-31, 2001.

[4] Ming Zhao, Arun Kumar, G.G. Md. Nawaz Ali and Peter Han Joo Chong. “A Cloud-based Network Architecture for Big Data Services”. In IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, 2016.

[5] Nguyen Thi Thanh Van and Tran Ngoc Thinh. “Accelerating anomaly-based IDS using Neural Network on GPU”. In International Conference on Advanced Computing and Applications, 2015.

[6] Yagang Zhang. “New Advances in Machine Learning”. ISBN: 978-953-307-034-6, InTech, 2010.

[7] S. Suthaharan and T. Panchagnula. “Relevance feature selection with data cleaning for intrusion detection system”. In Proceedings of IEEE Southeastcon, pp. 1-6, 2012.

[8] Minh Tuan Pham, Phuc Hao Do and Kanta Tachibana. “Feature Extraction for Classification Method using Princial Component based on Conformal Geometric Algebra”. In International Joint Conference on Neural Networks (IJCNN), 2016.

[9] Xiaohua Li and Jian Zheng. “Joint Machine Learning and Human Learning Design with Sequential Active Learning and Outlier Detection for Linear Regression Problems”. In Annual Conference on Information Science and Systems (CISS), pp 407-411, 2016.

[10] Lee Badger, Tim Grance, Robert Patt-Corner and Jeff Vaos. “Cloud Computing Synopsis and Recommendations”. NIST Special Publication 800-146, 2012.

[11] S. Mahdi Shariati, Abouzarjomehri and M. Hossein Ahmadzadegan. “Challenges and Security Issues in the Cloud Computing from two perspectives: Data security

and privacy protection". In 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), 2015.

[12] Geetanjali Nenvani and Huma Gupta. "A Survey on Attack Detection on Cloud using Supervised Learning Techniques". In Symposium on Colossal Data Analysis and Networking (CDAN), 2016.

[13] Candid Wueest, Mario Ballano Barcena and Laura O'Brien. "Mistakes in the IaaS cloud could put your data at risk". Symantec security response, 2015.

[14] Mohan V. Pawar and Anuradha J. "Network Security and Types of Attacks in Network". In International Conference on Intelligent Computing, Communication & Convergence, 2015.

[15] B.Santos Kumar, T.Chandra Sekhara Phani Raju, M.Ratnakar, Sk.Dawood Baba and N.Sudhakar. "Intrusion Detection System-Types and Prevention". In International Journal of Computer Science and Information Technologies (IJCSIT), Volume 4 (1), pp 77 – 82, 2013.

[16] Kayvan Atefi, Saadiah Yahya, Amirali Rezaei and Siti Hazyanti Binti Mohd Hashim. "Anomaly Detection Based on Profile Signature in Network Using Machine Learning Technique". In IEEE Region 10 Symposium (TENSYMP), 2016.

[17] Abhinav S. Raut and Kavita R. Singh. "Anomaly Based Intrusion Detection-A Review". In International Journal on Network Security, Volume 5, 2014.

[18] V. Jyothsna and V. V. Rama Prasad. "A Review of Anomaly based Intrusion Detection Systems". In International Journal of Computer Applications Volume 28–No.7, 2011.

[19] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández and E. Vázquez. "Anomaly-based network intrusion detection: Techniques, systems and challenges". In International Journal of Computer and Security, Volume 28, 2009.

[20] Phil Simon. "Too Big to Ignore: The Business Case for Big Data". ISBN 978-1-118-63817-0, Wiley, 2013.

[21] Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. "Foundations of Machine Learning". ISBN 978-0-262-01825-8, MIT Press, 2012.

[22] Website: https://www.wireshark.org/

[23] Website: https://www.tcptrace.org/

[24] Website: https://www.ll.mit.edu/ideval/data/index.html

ABBREVIATIONS

ANN	Artificial Neural Networks
API	Application Programming Interface
ARP	Address Resolution Protocol
DARPA	Defense Advanced Research Projects Agency
DB	Database
DDoS	Distributed Denial of Service
DoS	Denial of Service
FP	False Positives
FPR	False Positive Rate
FN	False Negatives
FSM	Finite State Machines
GA	Genetic Algorithms
GNP	Genetic Network Programming
GP	Genetic Programming
GB	Giga Bytes
GUI	Graphical User Interface
HIDS	Host-based Intrusion Detection System
ILOM	Integrated Lights Out Manager
HMM	Hidden Markov Model
IaaS	Infrastructure as a Service
IDS	Intrusion Detection System
IP	Internet Protocol
KDD	Knowledge Discovery and Data Mining
MIT	Massachusetts Institute of Technology
MITM	Man in the Middle

NIDS	Network-based Intrusion Detection System, Network Intrusion Detection System
OS	Operating System
OSI	Open Systems Interconnection
PaaS	Platform as a Service
SaaS	Software as a Service
SVM	Support Vector Machines
TCP	Transmission Control Protocol
TP	True Positives
TPR	True Positive Rate
TN	True Negatives
UI	User Interface
VM	Virtual Machine, Virtualized Machine
VMM	Virtual Machine Manager