CHAPTER SIX
EXPERIMENTAL
RESULTS AND
ANALYSIS OF
NIDSMAM
CHAPTER 6
EXPERIMENTAL RESULTS AND ANALYSIS OF NIDSMAM

6. Introduction
NIDSMAM, hybrid network intrusion detection system is able to provide the accurate detect to network against network attacks and malwares with prior knowledge of their characteristics signatures by using snort rules with Jess rules and multi-agents. On the other hand, we detect unknown attacks by using adaptive threshold algorithm with multi-agents to reduce time processing and response time to improve the performance of NIDSMAM.

The defense strategy, system objectives, used techniques and technologies needed to develop the system must be defined by the security experts prior to setup the network security system. These characteristics enable developing a system that is capable to accomplish its objectives with a high degree of efficiency and compatibility.

The objective of the proposed system is hybrid network intrusion detection system. It is in opposition to attacks and malwares which are depending on signatures database by using Snort rules and Jess rules. The system is employ multi-agents technology to design and implement. Because its characteristics in developing networked applications. The proposed system functions inside Microsoft windows environment. It also detects network traffic protocols especially TCP, UDP and ICMP.

The security policy of the proposed system is centering on misuse detection agents behavior rather than attacks behavior. Network attacks are increasing each and every day in number difficulty and innovation; therefore the definite behavior of intruder cannot be expected precisely to be ahead to it. Hence, concentrating on attacks behavior may add an enormous number of false positives and false negative which affecting the detection system with types of attacks. Consequently, concentrating on the agent behavior instead of attacks behavior may guide to a detect number of false positives and false negative, because it monitors the known attacks and unknown attacks behavior of the agents. On the basis, of careful monitoring system behavior by security experts an accurate detection system built. It should be identify attacks that will direct to valuable monitoring results and an attempt to detect the number of false positives and false negatives. Security system strategy makes sure that any attacks ought to bypass by one or more of these
multi-agents. Also, it enables recognition and blocks these attacks at prior phases before diffusion across the network.

6.1 REVIEW IMPLEMENTATION OF NIDSMAM

In NIDSMAM, we are explained the working of each component in the fifth chapter. Here, we are explaining the interaction of each component with other components. The above Figure 6.1 shows the implementation of the system. In NIDSMAM, packets are captured by the packet sniffer from network. At extremely first stage when the capture packets than it is to be stored on one file as for knowledge base. This file is collect and organizes the packet data in to rule format dynamically. This is known fact file named SnifferDb.txt.
After create of fact file then control of NIDSMAM comes to the detection agent. The detection agent using the knowledge base then we create the facts for misuse detection. Facts converted into Jess format same as in Jess rule engine. In detection agent we also convert the Snort rules to the Jess format. This Jess format is more convenient to read by Java. Here, both these facts and also Snort rules are providing to the misuse detection agents for detection of the attack. These snort rules and facts are preliminary requirements for detecting the attacks by NIDSMAM.

In NIDSMAM, we are using the two approaches, misuse detection and anomaly detection. The misuse detection approach contains required the snort rules and facts. The inference rule engine matches the facts with Jess rules (converted from Snort rules). The misuse detection system is for detection of known attacks. The inference rule engine we are using the Rete algorithm for pattern matching of packets. Particularly, if match found by Rete algorithm then it returns the true value and sends the packet information towards the analysis agent.

The analysis agent contains the five types of agents. The inference rule engine forwards these results of packets information to the central agent and then the central agent decides the packet for detection. Central agent takes the decision to send packets information to conversion agent. The main task of conversion agent is to transfer the received information packet to the response agent and UI agent using the methods explained in the analysis agent part in the fifth chapter. The response agent receives information packet from conversion agents and also provides notifications to UI agent and sends the informing to alarm agent to give out corresponding alarms.

UI agent interacts between the users with the system, the aim of integrating them flexibility into the multi-agents architecture. The main task of UI agent is to show proper action user can take if attack found on the system and sent packets information to the registration agent. Alarm agent sends notification to UI user when is detected attacks. The information about attack is received from response agent and conversion agent and shown to the UI user by alarm agent. Alarm agent sends this information to the registration agent. Registration agent stores the results of the misuse attack information to NIDSMAM database.
When the packet is not detected by the misuse agents then central agent sends that packet to anomaly detection agents i.e. for unknown attack. In this type of detection, we are using the Adaptive Threshold Algorithm. Anomaly detection describes the abnormal patterns of behavior, where abnormal patterns are defined. Central agent sends the packet data to the information agent. This information agent gets the information by using sendMessage() and receiveMessage() methods as explained in anomaly detection agents.

After getting the packet information, sensor agent senses the packet data from the information agent and process it according to each packet the sensor agent segregate the protocols (TCP SYN, TCP RST, TCP REST, UDP, and ICMP) and removes the unnecessary packet information and irreverent data and collect the bunch of packets to be input for adaptive threshold algorithm.

Statistic agent is responsible for making complex calculations and analysis over the packets data. According to the each type of protocols the statistic agent generates the result by using adaptive threshold algorithm. These results are forwarded to the decision agent to take the decision over detection of packets in anomaly agents.

The decision agent transfers detected type of attack to the database table. According to the type of attack and protocol type the data is sending to DSS. Results of misuse detection agents and anomaly detection agents are sent to the DSS. These results are stored in NIDSMAM database. Decision support system is an essential module is used for analyzing the information and data, so that the user can make decisions very simply.

The purpose of software based DSS to help user for compiling information from raw data, documents, and various charts. These charts will be explained further with screens. DSS includes five main components: Data Collection, Extract, transform and Load, Analysis, Reporting Tool, User Dashboard. DSS works according to the knowledge base i.e. database of NIDSMAM. We are using the Jfree charts API for displaying most of the charts on NIDSMAM user interface. These charts are shown accordance with data i.e. for selected any date. By using these charts user can easily make the decision over attack detection.

6.2 STAGES AND OPERATIONS OF THE NIDSMAM

In NIDSMAM to detect the attack we have captured the packets coming from the network. In the main screen, when click on the button start packet sniffer starts capturing
the packets from the network. The below Figure 6.2 shows the main screen of NIDSMAM. Type of protocol and start time is also shown in the main screen. The main screen we have options throw which user can interact with the system: Change interface, Add rule, Update rule, Edit rule, Misuse report, Set threshold, Anomaly report, Set password, Setting, DSS and Clear Repository.

![Figure 6.2 Illustrates Main Screen of NIDSMAM](image)

When user click on any button the respective screen is shown and the user can interact to the system. So when we want to start the system first we have to select the device interface. The available device interfaces are shown to the drop down list and the user can select the device. Following Figure 6.3 show how user can select the device interface. There are many options in the main screen of NIDSMAM:
6.2.1 Change interface

We can change interface of the system. Figure 6.4 shows change interface of NIDSMAM.

The above Figure 6.4 shows the dialog box throw which we can change the interface of NIDSMAM. Interfaces are nothing but the ways by which machine is able to connect with internet. When we have selected the device interface from the list and after click on the start button we start capturing the packets and these are shown on the table in the main screen. The table shows the packet information: source IP, source Port, destination IP, destination port, size, protocol. Figure 6.5 shows packets are received screen.
6.2.2 Add Rules
In NIDSMAM, the system having the facility that we can add user defined rules. These rules are defined by the system administrator who is having knowledge of rule signature. Following is Figure 6.6 shows add rule screen, there are several fields are to be filled by user. Like, we can select the protocol according to attack type then we can add source and destination IP addresses. The other information like, direction, content, ID message and alert message we can add to rule. These rules are already in Jess rule syntax so does not need to convert again like Snort rules. After filling this information then user can save this rule into one file. Here, we can reset button for clearing the content of all fields. Then, this file has been referred when detection of attack.

Figure 6.5 [Illustrates Packets are Received Screen]

Figure 6.6 [Illustrates Add Rules Screen]
6.2.3 Edit Rule

When the user add defined rules and after the interval the user wants to change some rules then in NIDSMAM provide the facility to edit these rules. Following Figure 6.7 shows that screen edits rules. If any rule wants to be edited the user simply clicks on that row in the table. After clicking row i.e. highlighted one the information of that the rule will be filled to text boxes according to their fields. Then user can edit these fields and click on the update button. So, the rule updated to the user defined rules file.

![Edit Rule Screen](Image)

6.2.4 Misuse Report

The information about the attack is shown in the misuse report table: target IP, protocol, type of attack, time stamp. When there is known attack then it is detected by misuse detection agents and the information is stored in the database. Following Figure 6.8 shows of misuse attacks detected by the NIDSMAM. In misuse report, we are showing the history of attacks detected by the system. We are storing in the database, and we can generate the report depending on select time interval and showing the misuse attacks information.
6.2.5 Set Threshold

In NIDSMAM, we can detect the unknown attacks using anomaly detection module. For that, NIDSMAM uses adaptive threshold algorithm. According to this algorithm system must have to maintain the value of the threshold. Because threshold value is very important to detect more valid attacks. Value of threshold must varied according to network traffic, means the number of packets for different protocols. The system provides the facility to change the value of threshold run time. System administrator can also set the value of $k$ for all protocol Figure 6.9 shows set threshold screen. If the system administrator sets these values at run time, then it is more helpful to detect the correct attacks. Administrator calculates the value of this threshold after analysis of previous values of threshold and detection rate of attack respectively. We setting default fixed of threshold the results in low accuracy of anomaly detection.
6.2.6 Anomaly Report

The information about the attack is shown in the anomaly report table: target IP, protocol, type of attack, time stamp. When there is unknown attack then it is detected by anomaly detection agents the information is stored in the database. Following Figure 6.10 shows anomaly report screen. In anomaly report we are showing the history of attack detected by the system. We are storing in the database and generating the report depending on select time interval and showing the anomaly attack information in the table.
6.2.7 Password Setting

In NIDSMAM, we are using the security and when user wants to login to the system must to provide the password. Later, can change the password and for that purpose we have provided the following screen which user can change its existing password. So with the password setting screen in NIDSMAM we can modify the passwords. Figure 6.11 shows password setting and change password.

![Password Setting](image)

*Figure 6.11/Illustrates Screen of Change Password*

6.2.8 Default Setting

In NIDSMAM, provides the facility for the default setting using that the system administrator can set the default setting for an entire application. In this module administrator can set default action after attack detected which may apply by user, default threshold for the first time running adaptive threshold algorithm and high alert on/off to set alarm sound. If administrator set the default action, then is will apply for each attack detect. If administrator set the value of threshold, than that value will use at the time of starting of application. If administrator high alert to “off” then, if the attack is detected it would not play beep sound of the system. These all values will store into the database and be applicable for NIDSMAM when administrator restarts it, Figure 6.12 shows default setting screen.
6.2.9 Clear Repository (Database)

In INDSMAM, the enormous database is maintained for analysis purpose like- detected packet information, detect time, receive time, type of attack and so on for both anomaly and misuse detection. System administrator will use that database for analysis to prevent the system from attacks. After one year or more administrators not uses the old data to analysis. The NIDSMAM provides the facility to delete all previous data from the database which is not useful according to the system administrator. Following Figure 6.13 illustrates date box.

![Figure 6.12 [Illustrates Default Setting]](image1)

![Figure 6.13 [Illustrates the date box]](image2)

In the main screen of MIDSMAM, “Clear Repository” option is there. System administrator will use that option to clear a data from database before selected date. After clicking on “Clear Repository” option will be appearing date box. Administrator can select any one date from that date box. When the administrator selects the valid date then
the below dialog box will appear for the last confirmation. When the system administrators click on “ok” then all data before the selected date will be removed. For i.e. above dialog box showing the selected value by the administrator is “2012-01-01” after clicking on “ok” all data from all tables will remove which is stored previously. Figure 6.14 illustrates the functionality of clear repository after selecting date.

6.2.10 UPDATING THE SNORT RULES

In NIDSMAM, updating the Snort rules we use the snort rules from Snort website. These snort rules are used for misuse detection agents. Snort is an open source network intrusion detection system developed by source fire. Combining the benefits of signature, protocol, and anomaly-based inspection, Snort is the most widely deployed intrusion detection system technology worldwide.
After creating the user account from Snort web site user can download Snort rules. The snort website will redirect to rules page and then user can see the updated version of Snort rules and can download the file on specified location. The above Figure 6.15 shows that example of Snort rules file. These downloaded rules are in the zipped file. By unzipping that rules, we get folder named ‘rules’. Than through NIDSMAM we can update rules library by clicking the update rules button. In Figure 6.16 shows the dialog for selecting updated rules folder. User has to select where these extracted rules are located. Than through Java code we update the new rules to our Snort rules directory.

![Updated Snort rules dialog](image)

**Fig 6.16 [Illustrates Dialog for Selecting Updated rules folder]**

Following code explains about the updating the Snort rules. We are fetching the selected folder through dialog box, and checking for each file in the selected folder that contains “.rules” file by using this condition `if(children[i].contains(".rules")).` After that we copying each file to snort directory by using this command `copyDirectory(new File(sourceLocation, children[i]));` After converting each file to Snort directory, then our system will generate the message “The Snort rules are updated successfully”.

```java
if (sourceLocation.isDirectory()) {
    if (!targetLocation.exists()) {
```

[215]
targetLocation.mkdir();

String[] children = sourceLocation.list();

for (int i=0; i<children.length; i++) {
    File f = new File(children[i]);
    if (children[i].contains(".rules")) {
        copyDirectory(new File(sourceLocation, children[i]),
                      new File(targetLocation, children[i]));
    }
}

JOptionPane.showMessageDialog(null, "The Snort rules are updated successfully");

} else {
    InputStream in = new FileInputStream(sourceLocation);
    OutputStream out = new FileOutputStream(targetLocation);
    byte[] buf = new byte[1024];
    int len;
    while ((len = in.read(buf)) > 0) {
        out.write(buf, 0, len);
    }
    in.close();
    out.close();

6.3 EVALUATION OF NIDS/MAM VIA DSS

The main problem with NIDS/MAM, it has no controls of daily/weekly in the network traffic. Therefore, thresholds had to be set large value to avoid false alarms and attempt to detection of true anomalies almost. The DSS it is a better way of handling and analysis data and shows it in charts and tables by selected date or selected last hour in the database. We show testing the network data periodically at the different interval time of one day. The date appears totally different from another date to testing the different intervals time within one day. We have different types from parameterized for each intervals time in the day. These parameters shown below are estimated on network points across the day. We have a network for each 10 sec of the day. When we receive a bunch of packets at some 10sec, we will account weighted average for each protocol using the 10sec network points to estimate mean, and we will an estimate $\alpha$ also. The network
points are updated using an exponentially weighted moving average (EWMA) from previous network points at that time. We will obtain different sets of network points, so that we get accurate results of the different dates and many remains to be achieved from advances to this system.

Decision Support System is an important module in NIDSMAM which is used for analyzing the information and data, so that the user can make decisions very simply. We used DSS database for handling a large number of flows at high speed network traffic in real time. The purpose of software based DSS to test results to the system for both misuse detection and anomaly detection. On the other hand, to help user for compiling information from raw data, documents, and various charts. The working of DSS charts is explained as per type of attacks found in misuse agents and anomaly agents. In NIDSMAM, there are two approaches are used for detection of attack. The first one is misuse detection i.e. for known attacks and anomaly detection i.e. for unknown attacks. Five levels of DSS are used for displaying the DSS screens: Data Collection, Extract, transform and Load, Analysis, Reporting Tool, User Dashboard.

6.3.1 Attack Types

There are two types of attack in DSS i.e. misuse attack and anomaly attack.

6.3.1.1 Misuse Attack

Following Figure 6.17 shows about misuse attack type for selected one day. This chart is selected on a previous date of data range. In Figure 6.18, we are trying to show time attack is detected, protocol type and how many packets are detected in that time interval. Let take an example, at time 4:0:00 we can see the attack for three protocols types (TCP, UDP, and ICMP). System administrator can analyze time attack and how many attacks are detected by misuse detection system and for which protocol types.
6.3.1.2 Anomaly Attack

Following Figure 6.19 shows how many attacks are detected by anomaly detection agents. In anomaly detection, there are five types of protocols TCP SYN, TCP RST, TCP REST, UDP and ICMP. We can see how much detection for each type of protocol with a
particular time interval. In general, UDP protocol type has more detection rather than other protocol types.

![Graph illustrating Anomaly Attack Type]

Figure 6.19 [Illustrates Anomaly Attack Type]

6.3.2 Detected Attacks

6.3.2.1 Misuse Detection

In misuse detection, there are number types of attacks. This graph is helpful for analyze the purpose if the administrator wants to get the result for each type of attack. In this graph system administrator can see, time and how many attacks are detected by misuse detection and type of attack. We can see the example 1, at 4:00:00 four various types of attacks detected. Administrator can take the decision about the actual environment of detection. Figure 6.20 shows misuse detection.
6.3.2.2 Anomaly Detection

Following Figure 6.21 shows detected attacks for anomaly detection agents. In anomaly, we have five types of attacks TCP SYN, TCP RST, TCP REST, UDP and ICMP flooding attacks. Following graph is able to show the total detection for each type of attack with time interval for detection.
6.3.3 Misuse Detect Rate

Misuse detected rate is the rate of detection of attacks in different interval of time. Following Figure 6.22 shows detection rate versus values, i.e. number of packets found in different interval of time for selected one day. By using this graph user can judge time the misuse agents detect the attack.

![Misuse Detected Rate](image)

*Figure 6.22 Illustrates Misuse Detect Rate*

6.3.4 Detect Rate with Attacks

In misuse detection agents detects the number of known attacks in different interval of time. The following Figure 6.23 shows us for selected date one day. DSS will display this graph that deals with attack type. In this way, we can make the decision on misuse detected rate with attack type. Here, user can take quick action and decision for detected attack type in misuse detection.
6.3.5 Detection Delay

In the Table 6.1, shows what delay is required by each detection in anomaly detection agents, we can show value delay time between 10-12sec. In NIDSMAM, we capture the received time for each packet in packet sniffer and also calculate the actual detected time for that packet. DSS calculating the difference between receive time and detect time and consider it as a detection delay. The detection delay is always near about 10 sec. Because in sensor agent, we collect the data for some time interval and gives that collection to adaptive threshold algorithm at once. System administrator can easily see the receive time; detect time and detection delay for each detection. Figure 6.24 illustrates detection delay anomaly.

![Figure 6.23 Illustrates Misuse Attack Type](image)
6.3.6 Probability Detection

In the Figure 6.25, screen explains table and graph also for detection probability. Detection of probability means the total detected packets for each type of attack minus successful detected attacks. Here, successful detection means detected by anomaly detection and which are actual attacks. System administrator can see the total detection
for each type of attack and also total successful detection and probability detection in percentage.

![Figure 6.25 Illustrates Probability Detection Anomaly](image)

### 6.3.7 Anomaly Analysis

For anomaly detection agents, the tuning parameters of the adaptive threshold algorithm are the amplitude factor $\alpha$, the EWMA factor $\beta$, and the $\mu$ and estimate the value of the threshold.

#### 6.3.7.1 Factor Alpha

Alpha is the value calculated using $\frac{(no\_of\_packets-threshold)}{threshold}$; following Figure 6.26 shows that variations in the value of alpha for each protocol. Parameter alpha is varied between [0.5, 2.5] for every particular time interval for every protocol on one hour, due to the interval time of delay detection. System administrator can easily see the variations in the alpha value make decisions.
6.3.7.2 Factor Mean

We can calculate Beta value by using the formula \[ \frac{2}{X_n + 1} \]; Using this value we can calculate the mean by using formula \[ \mu_n = (\mu_{n-1} \cdot \beta_n) + (1 - \beta_n) \cdot X_n \] The value of mean is calculating for every particular time interval for every protocol on one hour.

Figure 6.27 shows factor mean for anomaly analysis. This is very useful to analyze further or previous detected environment.
6.3.7.3 Threshold
In anomaly detection, we are detecting attacks using adaptive threshold algorithm, in this algorithm the value of threshold is very important. Because, detection is purely depending on the value of the threshold. Using some past time experience we calculate the value of the threshold. Following Figure 6.28 shows to selected two values of threshold (6,5) for every protocol in different interval time. Thus, by showing the variations in the value of threshold user can make the decision for attacks founded by anomaly detection at the particular time and threshold value. These are different values for threshold selected in interval time on one day shows the appropriate detection threshold for system benign behavior in real time. We can select value of threshold to improve the accuracy the system by tracking unusual behavior.

![Figure 6.28: Illustrates Threshold for Anomaly Analysis](image)

6.3.8 Anomaly Detected
In anomaly detection, we segregate the protocols TCP SYN, TCP RST, TCP REST, UDP, and ICMP according to each packet. These separated protocols are then forwarded to adaptive threshold algorithm. Here, we are showing variations in each protocol after detection in adaptive threshold algorithm.
6.3.8.1 TCP SYN Flooding Attack

Following Figure 6.29 shows that TCP SYN flooding attack. For this chart, DSS calculating how many attacks are detected of TCP SYN types. Using this charts system administrator able to analyze the future detection for TCP SYN type of attack. There is one TCP SYN flooding attack in interval time 7:24:00.

![Anomaly Detected Packets: TCP_SYN Attack](image)

*Figure 6.29 [Illustrates TCP SYN Flooding Attack by Anomaly Detected]*

6.3.8.2 TCP RST Flooding Attack

TCP RST flooding attack is shown by the following Figure 6.30. There are five TCP RST flooding attack are detected in different interval time in one day. System administrator can analyze i.e. time and at how many TCP RST attack is detected.
6.3.8.3 TCP REST Flooding Attack

Following TCP REST flooding attack is shown by Figure 6.31. We are displaying the time, and how many packets are detected of TCP REST flooding attack. Accordingly; user can figure out the exact decision on network. For example, suppose some attacks are happened on networks because of this some failure is occurred. In this case, system administrator using following graphs can able to detect the reason of failure.
6.3.8.4 UDP Flooding Attack

UDP flooding attack is the most common attack found in anomaly detection. Figure 6.32 is displaying UDP flooding attack by using DSS database; it means that we are displaying the time and how many packets are detected of UDP flooding attack. By showing this figure user can take the decision over UDP flooding attack. We can see from chart there are detection 12 of attacks for UDP flooding attack.

![UDP Flooding Attack](image)

Figure 6.32 [Illustrates UDP Flooding Attack by Anomaly Detected]

6.3.8.5 ICMP Flooding Attack

Following Figure 6.33 shows ICMP flooding attack. We can see at selected one day there are four ICMP flooding attack are detected in different interval time. System administrator can analyze i.e. the time and how many ICMP attack is detected.
6.3.9 False Detection Rate

In NIDSMAM false detection figure is generated after analysis of both anomaly and misuse detected data. In false detection rate, we are explaining false positive and false negative rate for NIDSMAM in different interval time on one day. The false detection rate increases by decreasing the speed of detection, and a trade-off between low false detection rate and minimum delay detection is necessitated. The value of threshold assures the sensitivity of the attack detection. Hence large value of threshold decreases the false positives rate. Therefore, a tradeoff between the value of threshold and false alarm is necessitated to control the sensitivity and to prevent miss detection. Several views of the predicted distribution of the network traffic. To reduce the network resources available to the flooding attacks, the breach between the normal traffic level and the threshold based on average false positive rate must be reduced. Following is a calculation for each false positives and false negatives.

- False positive: It is ignored packet by the system administrator while the attack is detected.
- False negative: is a counter for that packet which are not detected by misuse but detected by anomaly.
Figure 6.34 illustrates false detection rate, we can show total false positive rate and false negative rate for each protocol on one day. Figure 6.35 illustrates probability detection & false detection rate and Figure 6.36 illustrates detection rate & false detection rate.

Figure 6.34 [Illustrates False Detection Rate]

Figure 6.35 [Illustrates probability detection & false detection rate]
6.4 ANALYSIS NIDS/MAM SYSTEM WITH LITERATURE SURVEY

In the current state of the NIDSs, the network intrusion detection systems are prepared with different approaches. Some of the NIDS system adopts preferences for specific class of attacks which achieves improved accuracy but performs at a moderate level for other types of attacks. Another approach is followed by most of the hybrid NIDS systems is to combine different elements having preferences for a different kind of attacks and aggregating these elements to form a NIDS system having a wider range of protection from intruders. Such systems tend to become bulky and complex in operation resulting in compromising overall system performance and response time. The current network intrusion detection system are a combination of anomaly and misuse intrusion detection systems typically employ various data mining and machine learning based approaches. These approaches prove to be inefficient when compared to the signature based systems which uses pattern matching.

Though it is true that mixing multiple intrusion detection technologies into a single system can theoretically make a much stronger intrusion detection system, the resulting hybrid systems are not always better. Different intrusion detection technologies test system and/or network traffic and look for intrusive activity in different ways. Hence, the
main challenge to building an operational hybrid intrusion detection system is to obtain these dissimilar technologies to deal effectively and efficiently.

### 6.4.1 MISUSE DETECTION AND LITERATURE SURVEY

There are several works previously that are done for producing a complete system which can be used on any platform and affords the advantages of performing a detailed analysis and evaluation of the system attacks.

Chen et al. [CS09] using Two-stage decomposition of Snort rules towards effective hardware implementation. The normal and direct implementation of Snort rules in the hardware is an ineffective style. An efficient method of implementation is to break down the Snort rules into simpler signature patterns and then to implement them in the hardware and then the smaller patterns are stored. A feature of using this style in the project is easier implementation in the hardware along with dynamism. Smaller signature patterns enable high scalability of the matching operation and modify the online updating of the rules. This project requires more production cost of the hardware elements and their design and is very humble when done by using software tools. By doing use of software tools improve algorithms can be designed which help to gain the needed speed but implementation of all several rules may not be potential in such a system. The proposed project its execution in software makes it cost efficient and as well as it can be used on any system that needs security. Aside from this, the select of the Jess software which uses a highly efficient algorithm (Rete) enables to accomplish and maintain the speed inside the software itself.

While Attig and Lockwood [AL05] using A Framework for Rule Processing in Reconfigurable Network Systems. Several elements of the intrusion detection system are changed in this framework to integrate for a reconfigurable system implemented in hardware. It has also been tested on a system that was developed on Xilinx FPGA hardware. This project gives only the framework that can be used for the development of a security system in hardware. Simply, it is not a complete system which can be used immediately on any machine. It requires an overhead of designing, developing and testing of a system that uses this framework and as well as makes use of more time and cost, therefore it is not cost efficient. Implementation of these intrusion detection systems
in hardware has another drawback of being system particular i.e. the overhead needed or needed to move intrusion detection system from one platform to another is very high. In some cases, intact system might be required to alter for doing the system to work with this hardware. These restrictions on the system and its hardware do not exist if the system is developed in software and as well as the Jess based intrusion detection system makes it platform independent.

Sachidananda et al. [SSG06] using Priority-based high-speed intelligent rule-checking, it is another approach of implementing the rule checking algorithm for intrusion detection system. In priority based algorithm, priorities are specified to the packets in order to increase the probability of malicious packets detection during high speed data transfers. It is done by Intelligent String Arrangement (ISA) and using this ISA method gives us a higher probability of the malicious packets being detected even at higher data rates or throughputs. At higher data rates the system inclined to drop packets which it is not being capable to process and so there is a chance for the intrusion packet to be dropped. This sort of matter with the systems is removed by specifying priorities to the packets. [SSG06] Specifying priorities to the packets will enable the processing of high priority packets at the cost of low priority packets. This creates the system susceptible to dangers or intrusions which are given lower priorities. Therefore, the system is not completely secure but still is apt to some sort of risks forever. Packets are dropped because of slow processing time and the need to display and process the results directly at high input data speeds. The processing of the results can be made independent. Achieve such kind of independency enables the system to be capable to process the data at higher data rates. As a result to this currently analysis, the independency of the results processing fetched by writing the fired or matched rules onto a file. Storage of the results serves us in processing the results at point of time also.

While Chinthireddy [Chi11] developed a system that supplies of security against the intrusions in a network by indication messages and also storing these indications in a database is the main idea of this project. This is done by implementing several signatures developed by Snort also known as Snort Rules in Java based software Jess. Implementation in Jess gives the advantage of platform independency because of which this can be used on any system making it platform independent. Storage of the results or
indications is enabled to do a detailed analysis and notices on the sort of attacks passing in the network currently also.

6.4.2 ANOMALY DETECTION AND LITERATURE SURVEY

There are various works previously that are done for making a complete system which can be used on any platform and gives the benefits of executing a detailed analysis and estimate of the system attacks.

Statistical analysis has been used to detect both anomalies corresponding to network failures, also network intrusions [WZS02]. It is interesting to observe, both of these cases make use of the standard sequential change point detection approach. The Flooding Detection System, which was proposed by Wang et al. [WZS02] uses measured network data that describe TCP operations to detect SYN flooding attacks. The sequential change point detection employed here makes use of the nonparametric cumulative sum (CUSUM) method. Using this approach on trace-driven simulations, it has been shown that SYN flooding attacks can be detected with high accuracy and rationally short detection times.

Siris et al. [SPF04] used two adaptive threshold algorithm and (CUSUM) algorithm for change point detection to actively the parameters of the detection algorithm to achieve specific performance requirements.

Lazarevic and Ertoz [LE03] suggest different anomaly detection algorithms like supervised SVMs which were very promising in detecting new intrusions since they had very high detection rate but very high false alarm rate too. Additionally, in the Mahalanobis based approach, they were presently investigating the idea of defining various types of normal behavior and measuring the range to each of them in order to describe the anomalies.


Shaikh et al. [SAK] compared two statistical based anomaly detection algorithms one adaptive threshold algorithm and second cumulative SUM algorithm particularly with respect to the anomaly that was occurred due to SYN Flooding attacks. In the case of high intensity SYN Flooding attacks both were good and have less false alarm ratio
where as in the case of low intensity attacks adaptive threshold algorithm performance degrade and depicts high false alarm where as in the case of cumulative sum algorithm its performance remain same with lesser false alarm ratio. [ATF] prove that SIP flooding attacks on IMS can result in denial of service to the legitimate users. Afterwards, we our project using adaptive threshold algorithm with multi-agents for detection of flooding attacks in NIDSMAM. We evaluate the accuracy of the algorithm using traffic dataset that received from central agent consists of unknown attacks to identify normal and abnormal attacks for improved performance of the system and lesser processing time on the other hand, to reduce of response time and detect of false alarms.

6.4.3 HYBRID NIDS SYSTEM WITH LITERATURE SURVEY

We introduce a brief overview of works previously, identifying, where it differs from the formula presented in this work, to detect anomalies, different types of methods are currently used like rule based approaches, pattern matching, finite state machine, statistical analysis.

Tombini et al. [TDM+04] utilized an approach wherein the anomaly detection technique is used to list of unwanted provisions. The classifier module which uses a signature detection technique then classified the unwanted provisions into false alarms, attacks, and unknown attacks. This approach works on the hypothesis that the anomaly detection element would have a high detection rate, since neglected intrusions cannot be detected by the follow-up signature detection element. Additionally, it also presumed that the signature detection element will be capable to identify false alarms. While the hybrid system can remains miss specific types of attacks, its reduced false alarm rate increases the potential of testing most of the alerts.

Proposed in the literature Lunt et al. [LTG+92] and Porras and Neumann [PN97] and Tombini et al. [TDM+04] and Zhang and Zulkernine [ZZ06] that the monitoring potentiality of current intrusion detection systems can be improved by carrying a hybrid approach that includes of both anomaly also signature detection schemes. In such a hybrid system, the anomaly detection technique helps in the detection of novel or unknown attacks while the signature detection technique detects known attacks. The signature detection technique will also be capable to detect attacks established by a
malicious attacker who tries to change the behavior patterns with the target of retraining the anomaly detection module so that it will accept attack behavior as normal.

EMERALD [PN97] was developed in the late 1990’s at SRI. It is an extension of the seminal work done in [Den87] [DN85]. EMERALD is a hierarchical intrusion detection system that monitors systems at a variety of levels viz. individual host machines, domains and enterprises to form an analysis hierarchy. EMERALD uses a subscription-based communication scheme both within and between monitors. Yet, inter monitor subscription methodology is hierarchical and therefore limits access to events and/or results from the layer directly below. The system has a built-in feedback system that enables the higher layers to request more information about specific anomalies from the lower layers. To achieve a high rate of detection, the designers of EMERALD utilized an ensemble of techniques like statistical analysis engines and expert systems. The single most defining feature of EMERALD is its ability to analyze systemwide, domain-wide and enterprisewide attacks like Internet worms; DDoS attacks etc. At the top level.

On the other side, Zhang and Zulkernine [ZZ06] and Breiman [Bre01] applied the random forests algorithm in the signature detection module to detect known intrusions. Subsequently, the outlier detection supplied by the random forests algorithm is used to detect unknown intrusions. Approaches that use signature detection and anomaly detection in parallel have also been proposed. In such systems, two sets of reports of potential intrusive activity are made , and a correlation component analyzes both sets to detect intrusions. An example of such a system is NIDES [AFV+95].

Aydın et al. [AZC09] recommend a hybrid IDS by combining the two approaches in a single system. The hybrid IDS is obtained by combining packet header anomaly detection (PHAD) and network traffic anomaly detection (NETAD) which are anomaly-based IDSs with the misuse-based IDS Snort which is an open-source project. The hybrid IDS obtained is assessed using the MIT Lincoln Laboratories network traffic data (IDEVAL) as a test bed. Assessment compares a lot of attacks identified by misuse based IDS on its own, with the hybrid IDS achieved by combining anomaly-based and misuse based IDSs and proves that the hybrid IDS is a more powerful system.

Lee et al.[LNY+00]and Stolfo et al. [SCE+01]and Lee et al.[LSM99] they suggested a hybrid detection scheme that applied the Common Intrusion Detection Framework
(CIDF) to automatically get audit data, build models, and distribute signatures for novel attacks to check that the time needed to detect them is reduced. The aims of (NIDSMAM) system it have enabled different intrusion detection and response components to deal and share information and resources in a distributed environment.

6.5 EVALUATION OF NIDSMAM

NIDSMAM with various multi-agents and tools and algorithms, and how can interact between them to access the satisfactory results through explanations comprehensive about network security. Overview of network intrusion detection system and describes technologies selected for development system and overview of specifications and design NIDSMAM. Finally, the stage of experimental results and analysis of NIDSMAM. All the detections results of the system can be tested using DSS. Researches undertaken in NIDSMAM show how important resources in this system.

Matching pattern is an essential component of intrusion detection technologies. It is essential to develop efficient data structures and algorithms that permit the detection mechanism to the procedure all the tests. We use snort rules and Java expert system shell to help for matching pattern with facts after convert into Jess rules in NIDSMAM because these provides flexibility and using Rete algorithm for completed fast of the matching.

The main problem with statistical based approaches is that not all dramatic changes in the network are anomalies as they announce abnormality to any sudden changes. This announces the sudden changes in the network it is an abnormality. Therefore, requires an increase in detection accuracy. On the other hand, in the case of rule based approach it work only on the basis of signatures exist in its database, and it will capable to detect known attack, but it will unable to detect unknown attack whose signature is not existed in the database. Therefore, it is much specified and time wasting hence, common solution is necessitated. To avoid this problem we suggested hybrid network intrusion detection system by using two approaches (misuse detection system and anomaly detection system) with multi-agents.

In NIDSMAM, used adaptive threshold algorithm with multi-agents to display high accuracy for detected unknown attacks. On the other hand, adaptive threshold algorithm
based on the theoretical basis can display strong performance over several attack types, and not requires being complex or costly to carry out. In anomaly detection, agents concentrated on detecting TCP, UDP, and ICMP flooding attacks.

Investigated in the results the performance of NIDSMAM, each of the misuse detection agents and anomaly detection agents through all charts extracted by DSS included: (detection probability, attack types, anomaly analysis, detect rate, detection delay, false detection rate, anomaly detected, detected attacks, detect rate with attacks) and how the performance is influenced by the parameters of the algorithm.

Hence, there was the need to develop a light weight, less complex NIDS system which can achieve broadest possible detection from various types of network attacks and to reduce response time and fair system performance. In NIDSMAM attempts to address this problem with achieving high performance and less processing time without sacrificing on accuracy. There are many results of the analysis of the proposed system can be summarized below:

1) Use Java is a platform independent language because system specific libraries that provide the required programming language support for any software to build up a complete system do not exist in Java. Instead of it reduces the overhead of the addition of required libraries for the advanced system to execute, or to recompile the software every time there is a modification in the system that uses the software. Entire this distinctiveness that approach with Java as well as existing even in Jess, since of which the system becomes a platform independent system.

2) The basic step in NIDSMAM to detect misuse attacks we have to compare received packets with a database of snort rules. They use simple pattern matching concept it takes too long time for several rules. The use of Jess rule engine to reduce the time leads to increased productivity. Additionally, we can embed Jess into Java code for that Jess provides API.

3) When the Jess code is modulation in Java all the several advantages of the Java software will be present for creating the system tight, platform independent.

4) Rete algorithm makes Jess one of the most excellent ever rule engines existing now by eliminated the passive in terms of the memory usage as of the time taken by them to relate the rules and these flows.
5) The facts or data from the system being tested are removed from the working memory and then sent across the network for processing.

6) Central agent reduce the time for procedures the system which decided on whether the incoming packets should go to analysis agent or to be sent to the anomaly agents in accordance with the results from detection agent. It is play the important role to maintain flow of received packets information.

7) The purpose of the use pattern matching between facts and the stored rules to provide a high-level performance to a given other agents and furthers coupling between the other agents and the system.

8) Use multi-agents to communicate with each other to manage their activities to solve sophisticated problems that cannot be solved by a single agent.

9) We make use of JADE on agent technology because it has all the agent features that we need and also cause of communication between agents is easy to balance, and it is capable and tolerant of faulty programming.

10) NIDSMAM monitors traffic on a real time. NIDSMAM can detect malicious activity as they occur. Based on how the system is configured, such attack can be stopped even before they can access to the system.

11) Anomaly detection agent can detect attacks which misuse detection agent failure to detect.

12) NIDSMAM is easier to deploy as it does not affect existing systems or infrastructure because NIDSMAM is operating system independent.

13) It has detected false alarm rates for misuse and anomaly based on the predetermined rules and accounts.

14) Can detect unusual behavior and thus have the capable to detect evidences of attacks.

15) This system is the ability to supply the accurate detect to network against network attacks and malwares.

16) To reduce the processing time and reduce of response time which leads to improve the performance of NIDSMAM.

17) Multi-agents technology is used to design and implement the proposed system, due to its features in developing networked applications.

18) The proposed system operates inside Microsoft Windows environment.
19) NIDSMAM system it defends network traffic protocols especially TCP, UDP and ICMP.

20) NIDSMAM monitor packets in the network and compares with pre-configured and pre-determined attack patterns known as signatures.

21) Traditionally, one of the keys challenges most of NIDS employ deep packet inspection which limits the performance. We select high performance algorithms with multi-agents which enhance the performance. Current system can easily scale to throughputs, and performance in the future is likely to become very highly.

22) The proposed system its execution in software makes it cost efficient as well as it can be used on any system that needs security. On the other hand, the select of the Jess software which uses a highly efficient algorithm (Rete) enables to accomplish and maintain the speed inside the software itself.

23) To avoid dropping packets by reduced processing time via displaying the process and results directly at high input data speed and all results for misuse detection and anomaly detection are processed independently in the DSS.

24) The independency of NIDSMAM enables the system to be capable to process the data at higher data rates and the independency of the results processing got in DSS.

25) Storage of the results misuse agents and anomaly agents are enabled to do a detailed analysis and notices on the sort of attacks passing in the network currently.

6.6 CHAPTER SUMMARY

The defense strategy, system objectives, used techniques and technologies needed to develop the system must be defined by the security experts prior to setup the network security system. These characteristics enable developing a system that is capable to accomplish its objectives with a high degree of efficiency and compatibility. The objective of the proposed system is hybrid network intrusion detection system. It is in opposition to attacks and malwares which are depending on signatures database by using snort rules and Jess. To design and implement the proposed system multi-agents technology is employed. Because its characteristics in developing networked applications. The proposed system functions inside Microsoft Windows environment. It also detects network traffic protocols especially TCP, ICMP and UDP. In this chapter,
review implementation of NIDSMAM. And explain stages and operations of NIDSMAM includes main screen we have options throw which user can interact with the system: Change interface, Add rule, Update rule, Edit rule, Misuse report, Set threshold, Anomaly report, Set password, Setting, DSS, Clear repository. We explain evaluation of NIDSMAM via DSS. In NIDSMAM there are two types of approaches are used for detection of attack. The first one is misuse detection i.e. for known attacks and anomaly detection i.e. for unknown attacks. As explained in DSS, many levels of DSS are used for displaying the DSS screens includes (detection probability, attack types, anomaly analysis, detect rate, detection delay, false detection rate, anomaly detected, detected attacks, detect rate with attacks). We explain analysis NIDSMAM with literature survey includes misuse detection with literature survey, anomaly detection with literature survey, and hybrid NIDS system with literature survey. Finally, we explain evaluation of NIDSMAM, and described many results of the analysis of the proposed system.