



Efficient use of System Metrics for IDS Design

Sucheta Borse

Department of Computer Engineering,
BSIOTR Wagholi, Pune University, Pune, India

Prof. Mrs. R. A. Mahajan

Department of Computer Engineering,
BSIOTR Wagholi, Pune University, Pune, India

Abstract- *Security and protection of highly sensitive and private data are the major concern in the present era of Information Technology. In the present study, various intrusion detection and prevention system are studied. The main objective of this system is to provide a crucial solution for critical evaluation of variant attacks and their classification. The work presents an adaptive approach based on anomaly detection by observing system call patterns at kernel level to increase the probability of detecting the attacks and thereby reducing the false alarm rate. DLL are used to observe the legitimate and malicious traces. Semantic rules and use of multiple decision engines is the main attraction of this work. Hidden Markov Model and Extreme Learning Machine algorithms shows promising results respective to detection rate and false alarm rate. It has been observed that if appropriate training is provided to this system hundred percent results can be achieved. Publically available data set ADFA is used for training and testing of the system.*

Index terms — *Anomaly Detection, Intrusion detection, System call, Extreme Learning Machine, Hidden Markov model.*

I. INTRODUCTION

Recently research on machine learning for intrusion detection has standard much attention in the computational intelligence community. In intrusion detection algorithm, immense strengths of audit data must be analyzed in order to conception new detection rules for increasing number of novel attacks in high speed network. Intrusion detection algorithm should consider the composite properties of attack behaviors to improve the detection speed and detection accuracy [1]. Intrusion detection is the process of identifying and responding to suspicious activities targeted at computing and communication resources. An intrusion detection system (IDS) monitors and should be protected from a target system collects data, processes and gather information correlates and begins when an influx of responses did not evidence based on your input source., in network-based IDSs system and host-based systems can be classified [2]. The World Wide Web

(WWW) plays an important role in human life. Web applications are becoming increasingly popular in all aspects of human activities; ranging from science and business to entertainments. Consequently, web servers and web application are becoming the major targets of many attacks. Due to the increasing number of computer crimes, techniques that can sure and protect web servers and web applications against malicious attacks need highlights. Unfortunately the current network and transport layers, operational security solutions, web-based attacks to provide acceptable levels of protection against insufficient capabilities. These issues rise to the ever evolving research on web intrusion detection systems (WIDSs) is given [3]. An intrusion detection system (IDS) collects information from a computer or a network, and system or network to identify potential security against breaches analyzes this information. an overview of available in the literature various IDSs attack a certain range, with improved accuracy While other classes to explore different minor display preferences shows. huge computing power available on the same network to develop and implement a variety of made it possible for IDSs. Apart from the decision arrived at integration IDSs is a technology that can strengthen the final decision as to gather information from multiple sensors fusion process. and possibly skewed as defined Can sources and be a more descriptive, Intuitive and meaningful results to achieve the combination [4]. Anomaly-based techniques to explore an approach that abuse is a supplement to follow [8]. According to IDS signature-based detection technology detection (or detection of abuse) can be classified into and anomaly detection [10][11][12].

II. LITERATURE SURVEY

Yogendra Kumar Jain and Upendra,” An Efficient Intrusion Detection Based on Decision Tree Classifier Using Feature Reduction” In this paper, they reduced the features of the data set using information gain of the attributes. This study is approached to discover the best classification algorithm for the applications of machine learning to intrusion detection. Our simulation results show that, in general, the highest classification accuracy with minimum J48 error rate. On the other hand, we also found that decreased significantly in time learning algorithm and accuracy and increase in TPR. Comparison shows that reduction of the feature using information gain technique is suitable for the feature reduction. Using Weka, we analyzed four algorithms towards their suitability for detecting intrusions from KDD99 dataset. We showed that machine learning can be effectively applied to detect novel intrusions and focused on anomaly detection. The four learning algorithms J48, BayesNet, OneR and NB were compared at the task of detecting intrusions. J48 with an

accuracy rate of approximately 99% was found to perform much better at detecting intrusions than Bayes Net, OneR and NB Based on the experiments done in the paper and their Corresponding results, we can state the following: Machine learning is an effective methodology which can be used in the field of intrusion detection.

Giovanni Vigna, *Reliable Software Group* Christopher Kruegel, *Technical University Vienna* "Host-Based Intrusion Detection" For the past decade, network-based intrusion detection systems have clearly dominated host-based systems. The ease of maintenance and the possibility to monitor several targets with a single IDS installation has tipped the scales toward the network-based solution. However, the increasing use of very fast network links and encrypted connections have change the situation. The quality of audit data that is available at the operating system and application levels, the increasing security awareness of end users, and the improved accuracy of host-based techniques have all contributed to a higher acceptance of such detection mechanisms. This chapter host based intrusion detection systems and related technologies such as file integrity checkers and discuss virus scanner. Audit data (operating system and applications) were introduced and the main sources of data analysis were presented different perspectives. In addition, we host-based solution benefits and limitations on network-based techniques for analysis and possible future developments in the field reported.

Iman Khalkhali, Reza Azmi, Mozghan Azimpour-Kivi1 and Mohammad Khansari "Host-based Web Anomaly Intrusion Detection System, an Artificial Immune System Approach"

The main goal of this research was designing a host-based WIDS. They proposed to employ the enhanced custom log file in order to eliminate the inherent problems of common log files in defining web sessions boundaries. Moreover, ECL provides us with the POST requests along with the GET requests from the HTTP protocol. Different features were extracted from the ECL file which can represent the operations of the monitored web server. In this research, a dataset of normal and attack data were produced which can be used by other researchers in the field of WIDSs. Finally, they proposed the use of a novel RNS algorithm, inspired by the natural immune system, in order to produce a set of detectors that can cover the space of non-self (attack) properly and match to the non-self data and detect them.

Ciza Thomas and Balakrishnan Narayanaswamy "Sensor Fusion for Enhancement in Intrusion Detection"

Simple theoretical model to show improvement in the performance of fusion in IDS for the purpose of this paper is illustrated in the rate of detection and false positive rates. Threshold to quantify performance gains achieved through fixing. Also, more independent and to place individual attack IDSs, better fusion of IDS.

X. D. Hoang and J. Hu "An Efficient Hidden Markov Model Training Scheme for Anomaly Intrusion Detection of Server Applications Based on System Calls" In this paper, a HMM incremental training scheme from multiple observation sequences for anomaly intrusion detection has been presented. Their experimental results show that our HMM training scheme can save up to 60% training time compared to batch training. The scheme is very promising for use in the online HMM training for real-time intrusion detection. The initial values of HMM parameters in the training are sensitive factors to the convergence rate..

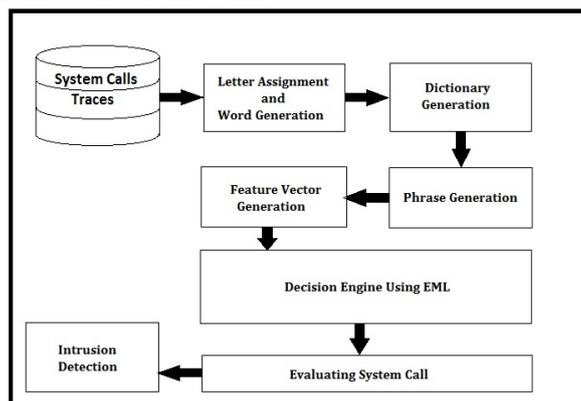
R Rangadurai Karthick, Vipul P. Hattiwale, Balaraman Ravindran "Adaptive Network Intrusion Detection System using a Hybrid Approach" In this paper, they have proposed a hybrid approach for adaptive network intrusion detection. They started off with HMM for network intrusion detection and it performed good empirically on DARPA data set. The difficulties that might arise when implementing HMM model in real time were described. They incorporated HMM model along with NB model into a hybrid model for intrusion detection. The proposed hybrid model also performed well in detecting intrusions and the experiments and results also reported. As an extension to HMM model, they would like to look at characterizing the diurnal variation characteristics of traffic to web server. It would involve learning the nature of traffic at various instances of the day.

III. IMPLEMENTATION DETAILS

Providing a Security to a single (Host) system was a challenging issue. Existing system exhibits following properties

- **A Secure host-based Intrusion Detection System:** Capable of securing a host system from all kind of threats.
- **Increase core performance of detection rate:** one of the main objectives of this project is to increase core performance of detection rate. Improve the possibility of detection of newly introduced attacks.
- **Reduce False alarm rate :** reduced false alarm rate provides better performance

Following Fig. shows system architecture of the work



This work is divided in two modules one is static module and another is dynamic module. During both modules work flow is same sequence of activity only difference is static module works on the bases of ADFA publically available data set and dynamic module captures real time system metrics and works on it.

Flow of the System:

- Behavioral traces in the form of DLL
- Application of semantic rule set
 - Letter assignment
 - dictionary generation
 - phrase generation
 - future vector generation
- Design of Decision Engine
- Evaluation of traces against current state
- Alarm Generation in case of Intrusion

Mathematical Model of System:

$S = \{ S_1, S_2, \dots, S_n \}$

S is a set of Unique System calls...

N is a no of system call.

$S_i \in \{ \text{open, read, mmap, write} \dots \}$

$T = \{ t_1, t_2, t_3, \dots, t_p \}$

T is a system call traces contain sequence of system call.

$T_j \in S$

$L = \{ A, B, \dots, Z, a, b, \dots, z, 0, 1, \dots, 9 \}$

L set of letters..

$W = \{ w_1, w_2, \dots, w_z \}$

W is a words combination of letters generated based on sequence of T.

$P = \{ p_1, p_2, \dots, p_k \}$

P phrase of combinations of word having 1 to 5 length.

K total no of phrases.

$F = \{ f_1, f_2, \dots, f_p \}$

F_i is the feature vector of given i length words.

$TD = \{ td_1, td_2, td_3, \dots, td_l \}$

Training data set given to ELM contain feature vector..

A is a set of unseen traces.

$A = \{ a_1, a_2, \dots, a_p \}$

3.1 Algorithm Used:

Existing Algorithms:

3.1.1 Semantic algorithm

Function GETWORDS(traces)

For all traces do

Counter <- 1

For system calls in trace do

Word=systemcall+nexccountercalls

If word in wordDictionary then

Increment count of word

Else

Add word to wordDictionary

End if

Counter += 1

End for

End for

Return wordDictionary

End function

Function GENPHRASES(word dictionary, length)

Create new phrase dictionary for phrase of \

Given length

For all words in words in word dictionary do

While current phrase length < length do

CurrentPhrase <- currentWord

For currentWord do

currentPhrase += next dictionary \

word

```

end for
end while
Add phrase to phraseDictionary
Word list start position++
End for
Return phraseDictionary
End function
Function GETPHRASECOUNT(trace)
featureVector = new array with\
length = number of dictionaries
for all Phrase Dictionaries do
i<- phrase length for dictionary
phraseCount <- 0
for all Phrases in Dictionary do
if phrase present in trace then
phraseCount++
end if
featureVector[i] <- phrase count
end for
end for
return featureVector
end function
function EVALUATESYSTEMCALLTRACE(newTrace)
newFeature <- getPhraseCount(newTrace)
normalize newFeature
feature -> trained decision engine
dbResult <- decision engine output
if deResult > global threshold then
classification <- anomalous
else
classification <- normal
end if
return classification
end function

```

3.1.2. Extreme Learning Machine

contains a detailed proof and exposition of the ELM methodology. This work is summarised below for convenience. The Moore-Penrose pseudo-inverse of matrix A is denoted as A^\dagger . Assume an ELM with \tilde{N} hidden neurons, training data (x_i, t_i) where $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbb{R}^n$ and $t_i = [t_{i1}, t_{i2}, \dots, t_{in}]^T \in \mathbb{R}^m$ and activation function $g(x)$. The output of this ELM, assuming that the underlying net structure is capable of approximation with zero error, is produced by:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j, j = 1, 2, \dots, N \quad (2.4)$$

In Equation 2.4, w_i is the weight vector from the i th hidden neuron and each input neuron, and β_i is the weight vector connecting the i th hidden neuron and the output layer. Recalling that an ELM only modifies the weights between the hidden layer and the output layer, β , and writing Equation 2.4 in matrix notation, we obtain the compressed representation:

$$H \beta = T$$

Where

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \quad (2.5)$$

Hence, holding the input weights constant, the solution to Equation 2.5 for β is given by:

$$\hat{\beta} = H^\dagger T \quad (2.6)$$

After solving Equation 2.6, the ELM can be updated with the new weights, β and commence assessment of subsequent data samples. System behaviour naturally changes over time as programs evolve, updates are distributed and user behaviour adjusts to reflect new tasks. As an anomaly-based IDS relies on an accurate system baseline in order to effectively highlight anomalous deviations from this defined norm, the decision engine must allow a means of updating the system baseline after initial deployment. The ELM algorithm provides for this requirement by means of the batch training method suggested, summarised below:

1. Set $M_0 = (H_0^T H_0)^{-1}$, $\beta^0 = M_0 H_0^T T_0$.
2. Use the new training data to calculate the hidden layer output vectors $h(k+1) = [g(\mathbf{w}_1 \cdot \mathbf{x}_i + b_1), \dots, g(\mathbf{w}_N \cdot \mathbf{x}_i + b_N)]^T$
3. Calculate updated $\beta^{(k+1)}$ using

$$M_{k+1} = M_k - \frac{M_k h_{(k+1)} h_{(k+1)}^T M_k}{1 + h_{(k+1)}^T M_k h_{(k+1)}}$$

$$\beta^{(k+1)} = \beta^{(k)} + M_{(k+1)} h_{(k+1)} (t_i^T - h_{(k+1)}^T \beta^{(k)}) \quad (2.7)$$

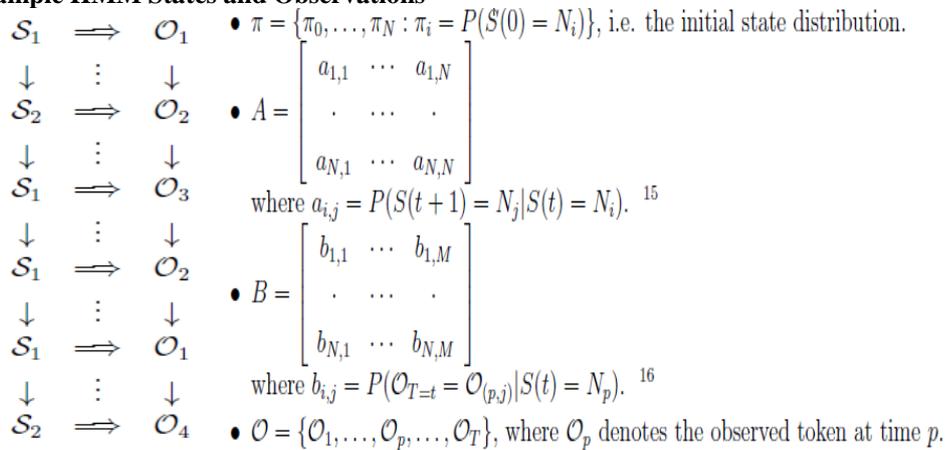
3.1.3. HMM

Consider the system shown in Figure 2.2. This diagram represents a system with two states, S1 and S2. These states are not directly observable, however, and the state transition is a stochastic process. At each state, a token is emitted. This is observable, and in this example there are 4 possible tokens, denoted O_{1-4} . The aim of an HMM is to model this process, and hence allow the user to predict the current state of the system using only the emitted tokens.

The following definitions apply to any given HMM, referred to as λ :

- N = The number of states in the system

Figure 2.2: Example HMM States and Observations



Given these definitions, a particular HMM can be denoted by the 3-tuple $\lambda = \{A, B, \pi\}$. To use an HMM, either the components of the 3-tuple must be known exactly, or a training process must be used to fit the model to a given set of training data. The Baum-Welch algorithm is commonly used for this fitting task, and performs multiple training iterations of a forward and backward pass process until a given accuracy threshold is reached. This process can be extensive, and research such as focuses on various techniques to reduce the training time without unduly sacrificing accuracy.

IV. RESULTS

Results are generated using ten different areas of traces in training and testing dataset.

Table 4.1 Results obtained

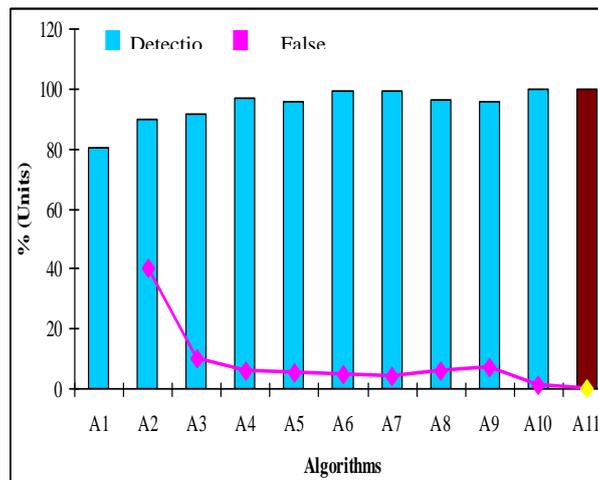
Training File Set	Testing Files	Intrusion Present/Not	Result
S1	T1	YES	DETECTED
S2	T2	YES	DETECTED
S3	T3	YES	DETECTED
S4	T4	NO	NO
S5	T5	NO	DETECTED(no training file)
S6	T6	NO	NO
S7	T7	NO	NO
S8	T8	NO	NO
S9	T9	YES	DETECTED
S10	T10	YES	DETECTED

In this table S1,S2,... are training dataset and T1.T2..... are testing data set. Row five shows intrusion as it won't find matching training Data and alarms intrusion which is a false alarm. It shows that hundred percent detection and no false alarm with proper training.

4.2 Comparative analysis and graphical representation

Comparative analysis with reference to detection rate and false alarms

Algorithm		Detection Rate	False Alarm Rate
A1	Data Mining of audit files [24]	80.2	Not cited
A2	Multivariate statistical analysis of audit data [25]	90	40
A3	HMM and entropy analysis of system calls [26]	91.7	10
A4	System call n-gram sliding window (assorted decision engines) [27]	96.9	~ 6
A5	RBF ANN analysis system calls [28]	96 mean	5.4 mean
A6	MLP ANN on subset of KDD98 [29]	99.2	4.94
A7	SVM on subset of KDD98 [29]	99.6	4.17
A8	kNN with Smooth Binary Weighted RBF [30]	96.3	6.2
A9	Rough Set Clustering [31]	95.9	7.2
A10	ELM using original semantic feature [1]	100	0.6
A11	Adaptive approach through IDS	100	0



Graph 8.3 Comparative analysis

4.3 Screen Shots of System Results

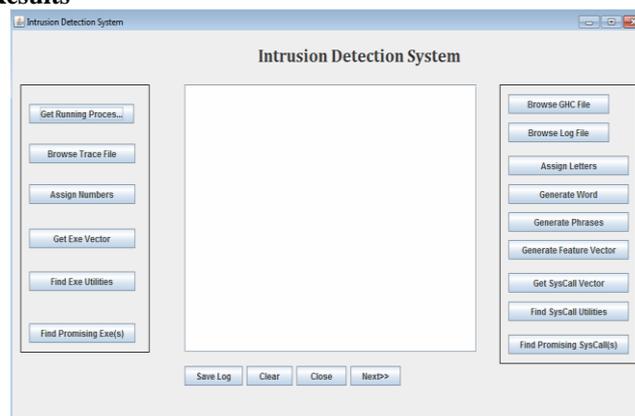


Figure 4.3.1 Intrusion Detection System snap shot

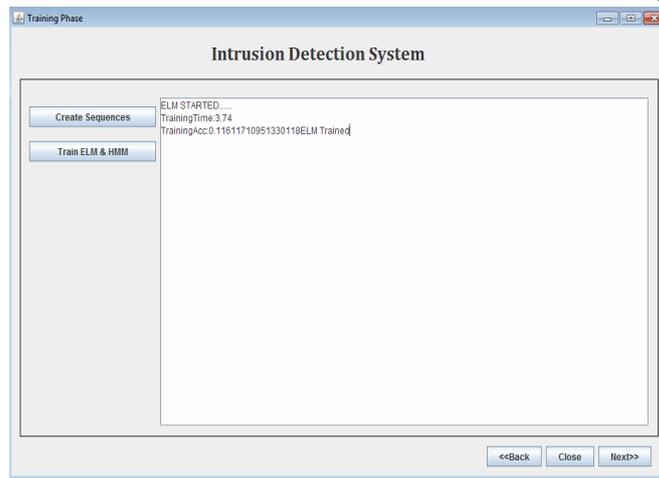


Figure 4.3.2 Training Phase

Results

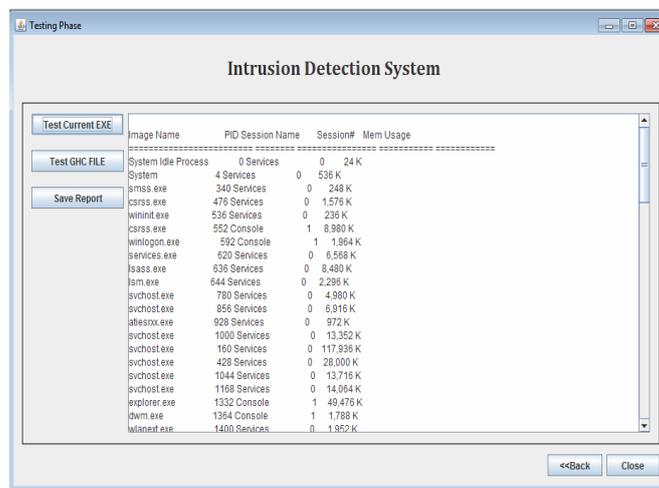


Figure 4.3.3 Testing Phase Results

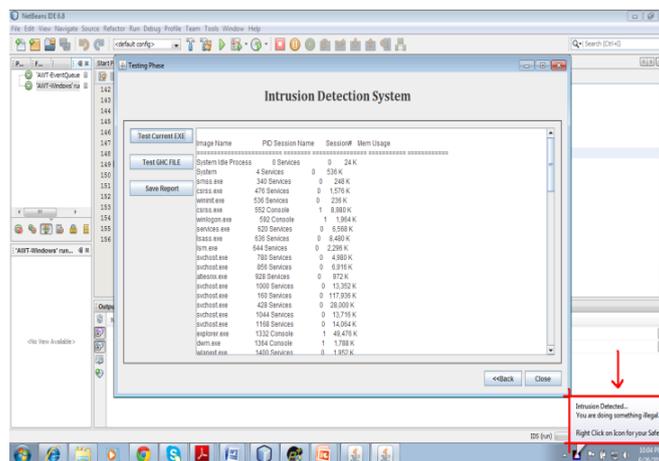


Figure 4.3.4 Alarms Intrusion Results

4.1 Hardware and Software Used

Hardware Configuration:

- PROCESSOR : PENTIUM IV 2.6 GHZ
- RAM : 512 MB DD RAM
- MONITOR : 15" COLOR
- HARD DISK : 20 GB

Software Configuration:

- Front End : JAVA
- Tools Used : NetBean 8.x
- Operating System: Windows XP/7

V. CONCLUSION

During this work study of different Intrusion detection and prevention systems are studied. The objective of this study was to improve core performance of intrusion detection and at the same time try to reduce heavy burden of false alarms present in traditional approaches. Application of semantic rules and use of multiple decision engine has helped to solve the objective. The semantic theory used defines a scalable set of rules governing the combination of terminating units. Decision engines like Hidden Markov Model and Extreme Learning Machine are the two methods used to differentiate between legitimate and malicious activities against base line of normal behaviour. Detection of these malicious activities results in system level lock for the host and provides protection against threat.

Public dataset were used for evaluation of the new algorithm in this project to allow comparison with existing approach. Results shows that if proper training is provided to IDS hundred percent results can be achieved in respect of detection rate as well as false alarm rate. To achieve this regular and rapid training ELM algorithm is used. The more rapid training and smaller on-going footprint of an ELM reduces the long term burden imposed by the IDS, without unduly affecting decision granularity. Use of Hidden Markov algorithm and ELM algorithm performs well through the experiments as expected. The results demonstrated in this paper are showing the current state of work done over practical implementation of this method. We have presented the details proposed approach for the implementation of this project. Applied proposed approach and find promising results in this area. Thus outcome of this system is hundred percent detection of intrusions. and zero percent false alarms.

REFERENCES

- [1] Yogendra Kumar Jain and Upendra, "An Efficient Intrusion Detection Based on Decision Tree Classifier Using Feature Reduction" *International Journal of Scientific and Research Publications*, Volume 2, Issue 1, January 2012
- [2] Giovanni Vigna, Reliable Software Group Christopher Kruegel, Technical University Vienna "Host-Based Intrusion Detection" JWBS001C-184.tex WL041/Bidgoli WL041-Bidgoli.cls June 15, 2005
- [3] Iman Khalkhali, Reza Azmi, Mozghan Azimpour-Kivi1 and Mohammad Khansari "Host-based Web Anomaly Intrusion Detection System, an Artificial Immune System Approach" *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 5, No 2, September 2011
- [4] Ciza Thomas and Balakrishnan Narayanaswamy "Sensor Fusion for Enhancement in Intrusion Detection"
- [5] X. D. Hoang and J. Hu "An Efficient Hidden Markov Model Training Scheme for Anomaly Intrusion Detection of Server Applications Based on System Calls"
- [6] G.-B. Huang, L. Chen, and C.-K. Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4):879–892, July 2006.
- [7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.
- [8] Darren Mutz, Fredrik Valeur, Christopher Kruegel, and Giovanni Vigna "Anomalous System Call Detection"
- [9] Bhavin Shah, Bhushan H Trivedi "Artificial Neural Network based Intrusion Detection System: A Survey" *International Journal of Computer Applications* Volume 39– No.6, February 2012
- [10] Milan Tuba, Dusan Bulatovic, 2010, Design of an Intrusion Detection System Based on Bayesian Networks, ACM.
- [11] Nabeel Younus Khan, Bilal Rauf, Kabeer Ahmed, 2010, Comparative Study of Intrusion Detection System and its Recovery mechanism, IEEE.
- [12] Ondrej Linda, Todd Vollmer, Milos Manic, 2009, Neural Network Based Intrusion Detection System for Critical Infrastructures, IEEE and also at Proceedings of International Joint Conference on Neural Networks.