A Comprehensive Mechanism for Intrusion Detection and Prevention

1Aastha goswami, 2Deepak Singh Rajput
1Research Scholar, 2Professor
1Computer Technology and Applications, 2Department of Computer Science and Engineering
Gyan ganga College of technology, Jabalpur, MP,
Aasthagoswami01@gmail.com
Deepakrajput16@gmail.com

***Abstract*— Intrusion Detection System is classified on the basis of the source of Data and Model of Intrusion. Anomaly in the Anomaly based Intrusion Detection System can be detected using various Anomaly detection techniques. Dimension Reduction can be done using Principle Component Analysis. Support Vector Machine can be used to specify the classifier construction problem. Intrusion Detection systems offer techniques for modelling and recognising normal and abusive system behaviour. Such methodologies include: statistical models, immune system approaches, protocol verification, file and taint checking, neural networks, white listing, expression matching, state transition analysis, dedicated languages, genetic algorithms and burglar alarms. This research describes these techniques including an IDS architectural outline and an analysis of IDS probe techniques finishing with a summary of associated technologies by using OSSEC. In this, OSSEC is used for detection and indicating the malicious activities trying to perform on server by different nodes in the network. After identifying these malicious nodes, prevention system is used for prevent server from these malicious nodes. The activity monitoring is done by log based analysis. Thus, security mechanisms to ensure its secure adoption are in demand. One security mechanism is intrusion detection and prevention systems (IDPS).**

***Index Terms*—Intrusion Detection and Prevention, Intrusion Alert, OSSEC, JMS, Log Analysis**

I. INTRODUCTION

An Intrusion Detection System (IDS) is a hardware/software combination or a combination of both hardware and software that detects the intrusions into a system or network. IDS complements a firewall by providing a thorough inspection of both the packets' header and its contents thus protecting against attacks, which are otherwise perceived by a firewall as seemingly benign network traffic. Firewalls look at the control rules; a packet is either allowed or denied. A rule specifies whether a host or a network, or an application should be allowed into the trusted network. To check the rules, a firewall has to just inspect the header of the TCP/IP protocol such as FTP, HTTP, or Telnet. However, it does not inspect the data contents of the network packet. Even if the data contains a malicious code, the firewall will allow this packet to pass through as the packet header has conformed to the rules configured in the firewall. Hence, you can still have a firewall but your trusted network can be compromised.

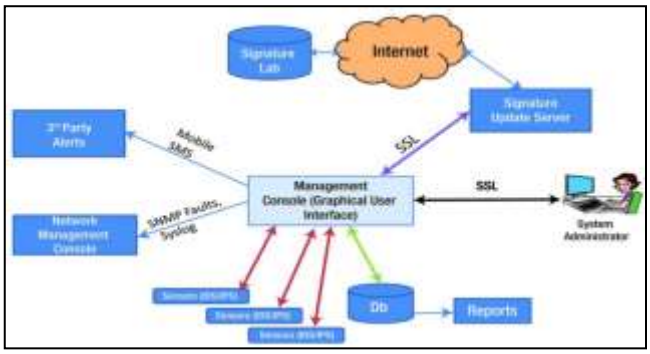

Figure 1.1 IDS/IPS Architecture

1.2 TYPES OF IDS

There are two types of IDS:

i. Host-based IDS: Protects the end system or the network resources.

ii. Network-based IDS: Monitors network traffic for attacks. A Network IDS is deployed on the network near a firewall, on the DMZ or even inside the trusted internal network.

1.2.1 Host-Based IDS (HIDS)

Host-based Intrusion Detection System refers to the detection of intrusion on a single system. This is normally a software-based deployment where an agent, as shown in Figure, is installed on the local host that monitors and reports the application activity. HIDS monitors the access to the system and its application and sends alerts for any unusual activities. It constantly monitors event logs, system logs, application logs, user policy enforcement, root kit detection, file integrity, and other intrusions to the system. It constantly monitors these logs and creates a baseline. If any new log entries appear, HIDS checks the data against the baseline and if any entries are found outside of this baseline, HIDS triggers an alert. If any unauthorized activity is detected, HIDS can alert the user or block the activity or perform any other decision based on the policy that is configured on the system.

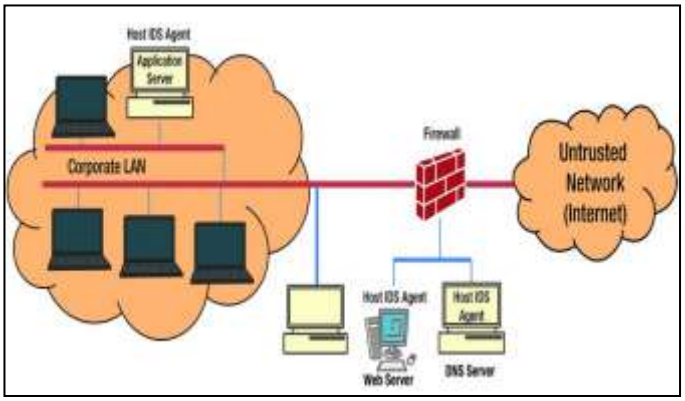

Fig. 1.1 Host-Based Intrusion Detection Systems

1.2.2 Network-Based Intrusion Detection System (NIDS)

A Network-Based Intrusion Detection System (NIDS) [1] monitors (and detects) any suspicious activity on a network. It checks each and every packet that is entering the network to make sure it does not contain any malicious content which would harm the network or the end system. Network Intrusion Detection System sniffs the network traffic continuously. The traffic is matched against known signature profiles and if there are any abnormalities found in the traffic, then a NIDS triggers an alarm to the management console. A single sensor, as shown in Figure, deployed in promiscuous mode or inline mode can monitor/protect several hosts in the network. Network IDS protects the network and its resources from the network perspective. For example, network IDS can detect reconnaissance attacks, Denial of Service attacks right at the network level. NIDS generates alerts as soon as it discovers these attacks. NIDS is a hardware/software solution placed near the firewall as an independent device (sensor) and has network operating system (TCP/IP stack). Sensors have interfaces to monitor the network (monitoring interface) and a management interface which is used for controlling and receiving alerts and for sending these alerts to the central management controller.

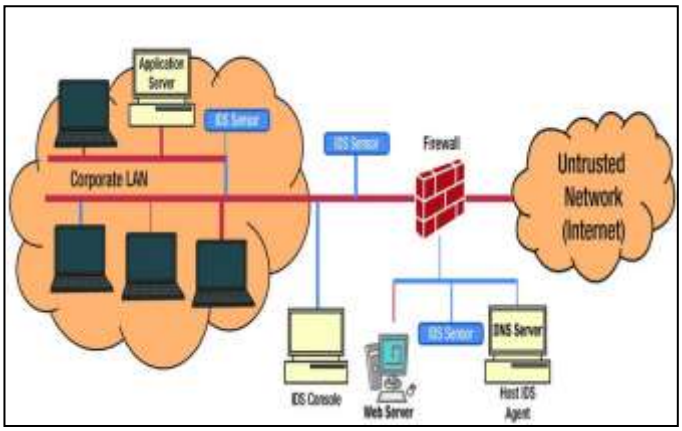

Fig. 1.2 Network-Based Intrusion Detection and Prevention System

II. WORKING METHODOLOGY OF LOG BASED HIDS

There are six phases, which had to go through in order to develop this system. The phases are System Analysis, System Design, System Development, System Implementation, System Testing and System Evaluation. For this project, only four phases had used to develop the system because of limitation of time and effort. First phase is System Analysis phase, which is information according to the study, will gather as much as needed. Second phase is System Design phase, which is the Host-based IDS, had designed according to the study and requirement of the project. The user and system requirement for this project are specify and project had reanalyzed according to the needs of requirement. The interface and architecture of the system had designed. The system analysis and design had modelled by using structured method. The third phase is the System Implementation phase, which is system, had implemented according to the design of system. Programming had done in this phase to transform the system from logical concept to a usable system by using Microsoft Visual Basic programming. The final phase is System Testing phase, which is system, had tested according to test case and the overall functionality of the system.

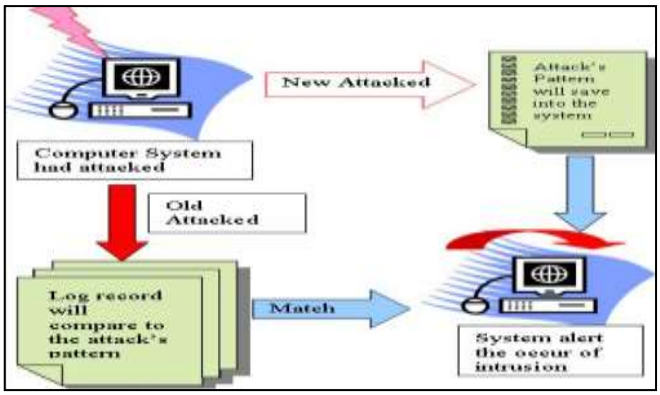

Fig. 1.3 Log Based IDS System Architecture

The system will recognize two types of attack and its pattern. If an attack is unknown pattern, the system needs to keep that pattern in the database for the future assessment. Then, if an attack knows pattern, the systems will match that pattern in their database and alert the host user about the attack or intrusion. Therefore, within that alert, user can take any possible action to react with the intrusion [6]. A context diagram in following Figure 2 is show how this system interacts with the end users to analysis log file and gets possible intrusion.

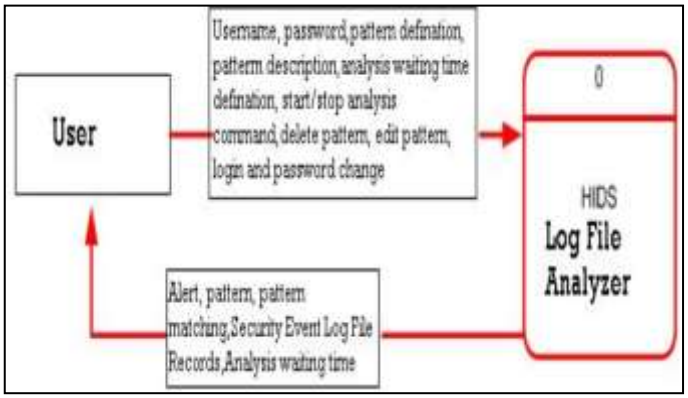

Fig. 1.4 Context Diagram of Log Based IDS System Architecture

III. PROPOSED METHODOLOGY

In computer network, both service providers and clients should secure the resources from malicious attacks by unauthorized elements. As it is a requirement for networks environment to have Intrusion Detection and Prevention System to detect attacks on their services, so, we are proposing this IDPS using OSSEC intrusion detection system.

The proposed intrusion detection and prevention system is host based Intrusion detection system (HIDS). This system is based on client's events monitoring process on server system. To implement the system, we are using virtualization software for deploying server over the Linux operating system. That complete virtualization is implemented on the windows operating system, which serves as the host operating system for Linux and OSSEC server system. In the proposed system, server's logs files are used to analyze the events perform in service provider system or server system. These log files are the activities or events of clients in the network. These log files are analyzed by using rule based analysis in the OSSEC server in real time. When IDS find some malicious activities during log analysis, it generates alerts messages to admin and sends these alerts to admin by using JMS (Java Message Services) in real time. When admin get these alerts, then admin take necessary actions to block the client's system's activities by using IDPS's API. The proposed framework is designed to show how the modules are integrated into the components and how they interact with each other to efficiently ensure the resilience of the enterprise network against intruders. Detailed below are the proposed design and the components

3.1 PROPOSED ARCHITECTURE

The diagram showing here is the general architecture of our proposed system. It contains basic working modules, modules interaction and control flow of our proposed system.

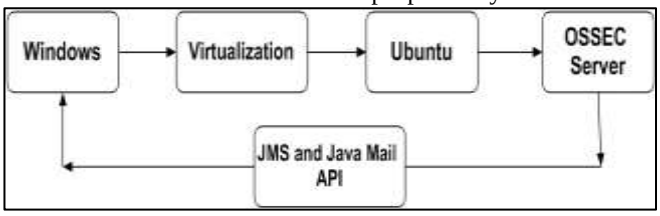

Fig 3.1 General Architecture of Proposed Methodology

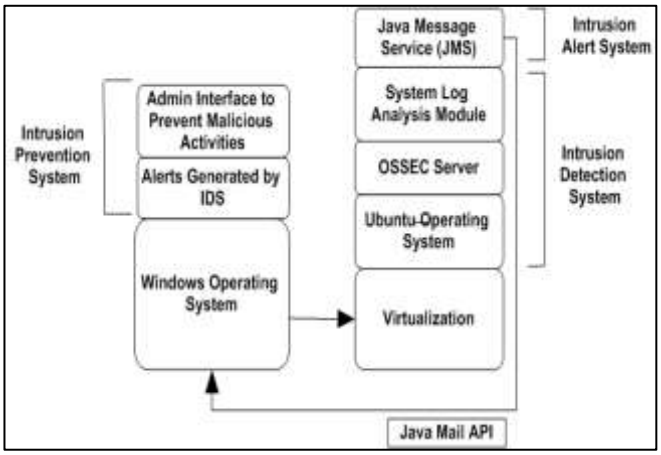

Fig 3.2 Working Architecture of Proposed Methodology

It monitors the integrity of the system. After studying various types of botnets and their operation, the knowledge of the actions taken by the controlled malware is used to define the configurations added in the HIDS for successful botnet activity detection. An overview of the event sequence is shown in Figure 3.3

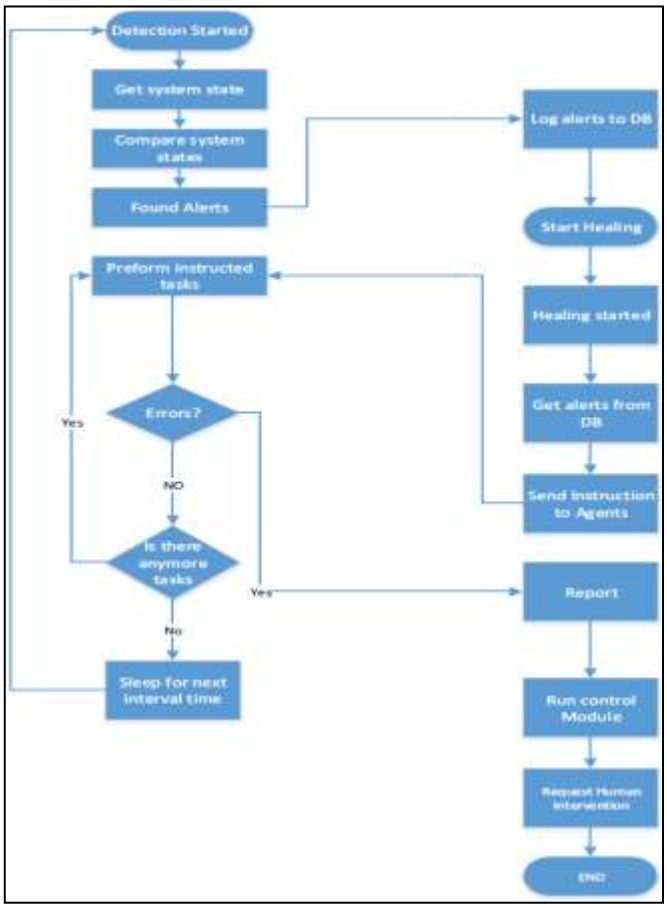

Fig 3.3 Work Flow of event sequence module of Proposed System

3.2 TOOLS USED

Here the list of tools or software, which has used to implement our proposed intrusion detection and prevention system.

i. Virtual Manager Workstation 9.0
ii. Ubuntu 14.0
iii. OSSEC
iv. JMS (Java Message Services)
v. Java Mail API

i. Virtual Manager Workstation 9.0

It is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems. It enables users to set up virtual machines (VMs) on a single physical machine, and use them simultaneously along with the actual machine. VMware Workstation allows for the installation of multiple instances of different operating systems, including client and server operating systems. It helps the network or system administrators to check, test and verify the client server environment.

ii. Ubuntu

Ubuntu is an operating system that is developed by a worldwide community of programmers as well as by employees of Ubuntu's commercial sponsor, Canonical. Ubuntu is based on the concept of free or open-source software, meaning that you do not pay any licensing fees for Ubuntu. Ubuntu is built on Debian's architecture and infrastructure, to provide Linux server, desktop, phone, tablet and TV operating systems.

iii. OSSEC

OSSEC is intrusion detection and active response application designed for use with Linux and Windows systems. It can be setup to run on a single system or with a server/client relationship. OSSEC uses log analysis to detect intrusions

and is highly customizable, capable of analyzing custom logs. OSSEC is an open source HIDs, or Host-based Intrusion Detection software. OSSEC watches your logs for signs of intrusion, system availability problems or things that just don't look right. It monitors your files and network devices for changes, your system for root kits and can even block attackers in near real-time. OSSEC supports dozens of log formats and has hundreds of rules to detect the suspicious activities.

iv. JMS (Java Message Services)

Java Message Service (JMS) is an application program interface (API) from Sun Microsystems that supports the formal communication known as messaging between computers in a network. Sun's JMS provides a common interface to standard messaging protocols and also to special messaging services in support of Java programs. Sun advocates the use of the Java Message Service for anyone developing Java applications, which can be run from any major operating systemplatform.

v. Java Mail API

The Java Mail API is an optional package (standard extension) for reading, composing, and sending electronic messages. You use the package to create Mail User Agent (MUA) type programs, similar to Eudora, Pine, and Microsoft Outlook. Its main purpose is not for transporting, delivering, and forwarding messages like send mail or other Mail Transfer Agent (MTA) type programs. In other words, users interact with MUA-type programs to read and write emails. MUAs rely on MTAs to handle the actual delivery.

The Java Mail API is designed to provide protocol-independent access for sending and receiving messages by dividing the API into two parts:

- The first part of the API is the focus of this course. Basically, how to send and receive messages independent of the provider/protocol.
- The second part speaks the protocol-specific languages, like SMTP, POP, IMAP, and NNTP. With the Java Mail API, in order to communicate with a server, you need a *provider* for a protocol. The creation of protocol-specific providers is not covered in this course as Sun provides a sufficient set for free.

IV. IMPLEMENTAION AND RESULTS

INSTALLATION PROCESS OF OSSEC – OSSEC is an Open Source Host-based Intrusion Detection System that performs log analysis, file integrity checking, policy monitoring, root kit detection, real-time alerting and active response. It runs on most operating systems, including Linux, MacOS, Solaris, HP-UX, AIX and Windows. It also includes agentless monitoring. Requirements for installing OSSEC server:

- An Ubuntu 14.04 server
- Apache2, PHP, MySQL and development packages
- OSSEC clients to monitor

Installing development packages

OSSEC is set up from source, hence you require development packages. This is both for the OSSEC clients as well as for the OSSEC server:

apt-get install build-essential make libssl-dev git

Installing Apache, MySQL and PHP- We have already install all the required software while LAMP installation. Now the only requirement is to installation and configuration of OSSEC.

Installing OSSEC Web UI - This is as well quite uncomplicated. Since we've already set up Apache and PHP, we can now download the web UI and take out to /var/www/html.

Client installation - Download and authenticate the OSSEC 2.8 .tar.gz file as described. Don't disregard to install the development packages. This time, do an agent installation. Adding a client to OSSEC is moderately simple. Initial you add the client to the server, which provides you a key. After that you put in this key to the client, modify the config file on the client and that's it. First we require generating a key on the OSSEC server for this client. We execute this by running

/var/ossec/bin/manage_agents

Next, entering the hostname, IP and ID for the client we want to add. Do this on the OSSEC server.

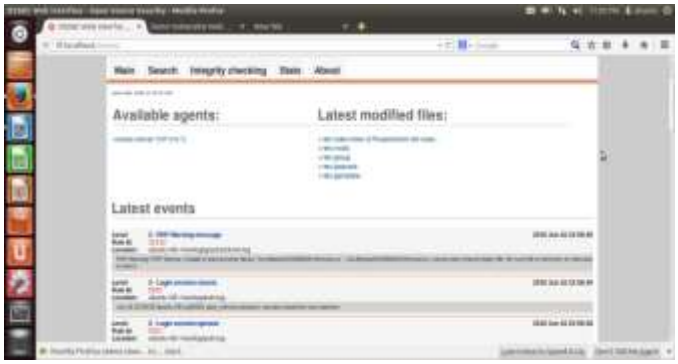

Fig. 4.1 OSSEC Starter Dashboard

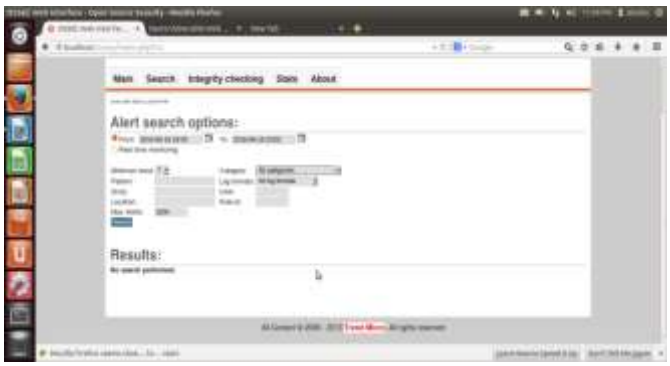

Fig. 4.2 IDS Alert Search option

Fig. 4.3 List of Modified Files

Fig. 4.4 List of Various Events

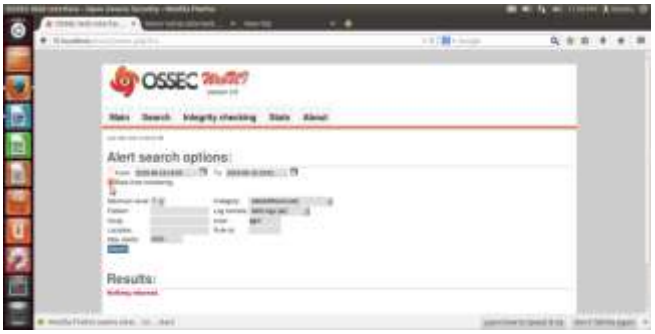

Fig. 4.5 Alert Search Option (Real Time or Duration Wise)

Fig. 4.6 Intrusion Alert Statistics

4.1 COMPARISIONS WITH EXISTING SYSTEM

S. No.	Parameters	Existing System	Proposed System
01.	**Notification (Contains Information of Activities)**	Not Containing Complete Activity Information	Containing Complete Activity Information
02.	**Real Time Notification**	Notification not Time Bound	Real Time Notification Available
03.	**Interface for Block/Prevent Malicious Activities**	Not Available	User Interface Available
04.	**Real Time Prevention Mechanism**	Not Implemented	Implemented
05.	**Real Time Alert Mechanism**	Real Time Alerts Not Available	Real Time Alerts Available
06.	**Multi-Platform Support**	Single Platform	Multi-Platform

Table 4.1 Comparison Table

4.3.1 Comparisons Graph between Existing System and Proposed System

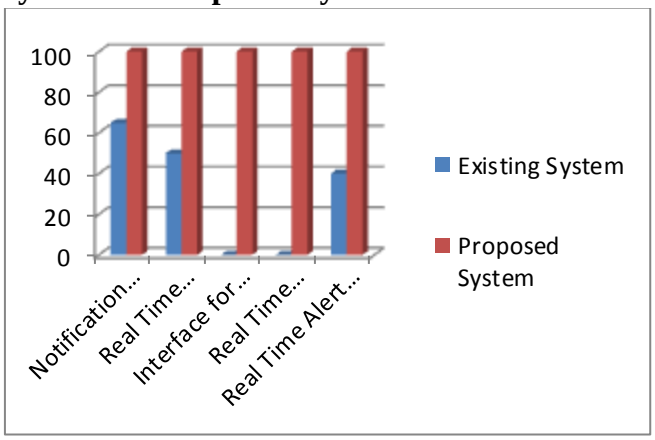

Fig.4.7 Comparison Graph of Existing and Proposed System

V. CONCLUSION

In this work, we have proposed intrusion detection and prevention system using OSSEC, JMS which is implemented on Ubuntu-12.04, Apache-2.0 and Mysql Server. This system has been tested by simulating various types of attack. The proposed system detects the attacks correctly and efficiently. First the proposed system takes into account the scenario approach and the behavioural approach. Evaluating an intrusion detection system is a difficult task. Indeed, it can be difficult even impossible to identify the set of all possible intrusions that might occur at the site where a particular intrusion detection system is employed. To start with, the number of intrusion techniques is quite large. Then, the site may not have access to information about all intrusions that have been detected in the past at other locations. Also, intruders can discover previously unknown vulnerabilities in a computer system, and then use new intrusion techniques to exploit the vulnerabilities. Another difficulty in evaluating an intrusion detection system is that although it can ordinary detect a particular intrusion, it may fail to detect some intrusion when the overall level of computing activity in the system is high. This complicates the task of thoroughly testing the intrusion detection system.

VI. FUTURE WORK

The proposed network intrusion detection and prevention system is extensible and portable and much other functionality can be implemented. Implementation of HIDS was but still there are many features which can be incorporated into that. Even NIDS tools can be combined along with the HIDS. There are few limitations for this implementation i.e. its maintenance in a bigger network etc. For these kinds of problems an alternate solution must be found. This can be improved much more from the administrator point of view i. e high monitoring based on other parameters and access on remote system.

REFERENCES

[1]. Jabez J, Dr.B.Muthukumar, "Intrusion Detection System (IDS): Anomaly Detection using Outlier Detection Approach", International Conference on Intelligent Computing, Communication & Convergence (ICCC-2015), ScienceDirect Procedia Computer Science 48 (2015) 338 – 346.

[2]. Muhammad Imrana, Farrukh Aslam Khanb, Tauseef Jamala, Muhammad Hanif Durada, "Analysis of Detection Features for Wormhole Attacks in MANETs", Procedia Computer Science 83 (2016) 1090 – 1095, 1877-0509 © 2015 Published by Elsevier.

[3] Naila Belhadj Aissa, Mohamed Guerroumi, "Semi-Supervised Statistical Approach for Network Anomaly Detection", Procedia Computer Science 83 (2016) 1090 – 1095, 1877-0509 © 2016 The Authors. Published by Elsevier.

[4]. Malik Shahzad Kaleem Awan, Pete Burnap, Omer Rana, "Identifying cyber risk hotspots: A framework for measuring temporal variance in computer Network risk", 0167-4048/© 2016 The Authors. Published by Elsevier Ltd.

[5]. Sreenivas Sremanth Tirumala, Hira Sathu, "Free and Open Source Intrusion Detection System: A Study", 978-1-4673-7220-6/15/$31.00©2015 IEEE.

[6]. C. B. Westphall and F. R. Lamin. SLA Perspective in Security Management for Cloud Computing. In Proc. of the Int. Conf. on Networking and Services (ICNS), 2010. Pp. 212-217.

[7]. Hisham A. Kholidy, Fabrizio Baiardi CIDS: A framework for Intrusion Detection in Cloud Systems, 2012 Ninth International Conference on Information Technology- New Generations, 978-0-7695-4654-4/12 $26.00 © 2012,pp 379-385.

[8]. Karen Scarfone and Peter Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)", National Institute of Standards and Technology(NIST), Special Publication 800-94, Feb. 2007.

[9]. J.H. Lee, M.W. Park, J.H. Eom, T.M. Chung, "Multi-level Intrusion Detection System and Log Management in Cloud Computing", In 13th International Conference on Advanced Communication Technology, pp.552-555, 2011.

[10]. H. Jin, G. Xiang, D. Zou et al., "A VMM-based intrusion prevention system in cloud computing environment," The Journal of Supercomputing, pp. 1–19, 2011.

[11]. 18. J. Amudhavel1, V. Brindha, B. Anantharaj, P. Karthikeyan, B. Bhuvaneswari, M. Vasanthi, D. Nivetha and D. Vinodha, "A Survey on Intrusion Detection System: State of the Art Review", ISSN (Print) : 0974-6846, ISSN (Online) : 0974-564, Indian Journal of Science and Technology, Vol 9(11), March 2016.

[12]. Joseph Mbugua Chahira, Jane Kinanu Kiruki, Chuka, Peter Kiprono Kemei, "A Review of Intrusion Alerts Correlation Frameworks", International Journal of Computer Applications Technology and Research Volume 5– Issue 4, 226 - 233, 2016, ISSN:- 2319–8656.

