International Journal of Advanced Trends in Computer Applications
*www.ijatca.com*

# Distributed Intrusion Detection System Using IDMEF

**Manish Kumar[1], Dr. M. Hanumanthappa[2]**
[1]Asst. Professor, Dept. of Master of Computer Applications
M. S. Ramaiah Institute of Technology, Bangalore-54
Research Scholar, Department of Computers Science and Applications,
Bangalore University, Bangalore, INDIA
*manishkumarjsr@yahoo.com*
[2] Professor, Dept. of Computer Science and Applications,
Jnana Bharathi Campus, Bangalore University,
Bangalore -560 056, INDIA,
*hanu6572@hotmail.com*

**Abstract:** *Intrusion is defined as a set of actions that attempt to compromise the integrity, confidentiality or availability of a information resources. An intrusion detection system (IDS) monitors network traffic or system logs for suspicious activity and alerts the system or network administrator. The current intrusion detection systems have a number of problems that limit their configurability, scalability and efficiency. There have been some propositions about distributed architectures based on multiple independent agents working collectively for intrusion detection. A Distributed IDS (DIDS) consists of several IDS over a large network (s), all of which communicate with each other, or with a central server that facilitates advanced monitoring. In a distributed environment, DIDS are implemented using cooperative intelligent agents distributed across the network(s). On the basis of analyzing the existing intrusion detection system (IDS) based on agent, this paper proposes architecture for distributed Intrusion Detection System where comprehensive data analysis is executed in a centralized computing environment. The proposed architecture is able to efficiently handle large volumes of collected data and consequent high processing loads. Experiments proved that the system could complete the intrusion detection tasks by making full use of various resources collaboratively, and thus the detection speed and accuracy of the system could be improved.*

**Keywords:** *Intrusion Detection System (IDS), Distributed Intrusion Detection System (DIDS), Intrusion Detection Message Exchange Format (IDMEF).*

## 1. Introduction

Intrusion Detection Systems (IDS) are important mechanisms which play a key role in information security. The most popular way to detect intrusions has been by using the audit data generated by the network or operating system. An audit trail is a record of activities on a system that are logged to a file in chronologically sorted order. Since almost all activities are logged on, it is possible that a manual inspection of these logs would allow intrusions to be detected. However, the incredibly large sizes of audit data generated (on the order of 100 Megabytes a day) make manual analysis impossible. IDSs automate the audit data analysis.Such systems perform automatic detection of intrusion attempts and malicious activities through the analysis of network traffic or using a system log analysis. Such data is aggregated, analyzed and compared to a set of rules in order to identify attack signatures, which are traffic patterns present in captured traffic or security logs that are generated by specific types of attacks. In the process of identifying attacks and malicious activities an IDS parses large quantities of data searching for patterns which match the rules stored in its signature database. Such procedure demands high processing power and data storage access velocities in order to be executed efficiently in large networks.

## 2. Distributed Intrusion Detection System

A distributed IDS (DIDS) consists of multiple Intrusion Detection Systems (IDS) over a large network, all of which communicate with each other, or with a central server that facilitates advanced network monitoring, incident analysis, and instant attack data. By having these co-operative agents distributed across a network, incident analysts, network operations, and security personnel are able to get a broader

view of what is occurring on their network as a whole[11]. A number of IDSs have been proposed for a networked or distributed environment. Early systems included ASAX [1], DIDS[14] and NSTAT [13]. These systems require the audit data collected from different places to be sent to a central location for an analysis. However, the main problem with such an approach is that if two or more IDSs that are far apart in the hierarchy detect a common intruder, the two detection cannot be correlated until the messages from the different IDSs reach a common high-level IDS.

### 2.1 Advantages of Distributed Intrusion Detection Systems

As attackers, and attack methods become increasingly complex, the need for a DIDS system in large corporate, and military networks increases drastically. With the increased complexity of these attacks, analysts are leaving themselves open to the problems of communications breakdowns, where one analyst sees a single attack on his segment, and dismisses it as nothing. While several other segments receiving the same attacks in a coordinated manner, their analysts may be dismissing the seriousness of the attack. However, when all the attack data is viewed together, a dramatically different perspective the attack may emerge. The DIDS system gives the analyst a quicker, easier, more efficient method to identify coordinated attacks across multiple network segments, and to trace back the activities of the attackers. The system also, ultimately, saves the corporation whose networks it is deployed on money by reducing the number of Incident Analysts needed, as well as the amount of time required to gather logs from the various IDS systems setup in a large corporate network. By having all of these attack records stored in a single place, it allows the analyst much more flexibility in discovering attack patterns, and other attack issues which may have otherwise gone unnoticed.

### 2.2 Incident Analysis With Distributed Intrusion Detection Systems

Incident analysis using the DIDS system is really what it is all about. This is where all the power, potential, flexibility, and strength of the system as a whole lies. It is the reason why the DIDS was first conceptualized, to allow for advanced analysis of attacks occurring over multiple network segments, and at an advanced level[11].

Aggregation is one of the common approach which is used to facilitate this advanced method of analysis across a networks multiple segments. By aggregating similar or related data, the analyst is able to easily see how an attack progressed through the different stages: from active network reconnaissance, to the final attack. It is possible for the incident analyst to see what kind of time frame the attacker was working within and to correlate other attack attempts against the networks to determine if there were multiple co-operative attackers. The most common methods of aggregation are according to attacker IP, destination port, agent ID, date, time, protocol, or attack type.

- Aggregating by attacker IP allows the analyst to view the steps of an attacker's attempt from start to finish across the multiple network segments.
- Aggregating by destination port allows an analyst to view new trends in attack types, and to be able to identify new attack methods, or exploits being used.
- Aggregating by agent ID allows an analyst to see what variety of attacks and attackers have made attempts on the specific network segment the agent is on. Consequently, the analyst can determine if there are multiple attackers working in conjunction, or if there are network segments that are of more interest to attackers than others, thereby giving the security team a list of common targets to work on.
- Aggregating by date and time allows the analyst to view new attack patterns, and to potentially identify new worms or viruses that are only triggered at certain times.
- Aggregating by protocol helps in a purely statistical manner, which could allow an analyst to identify new attacks in particular protocols, or identify protocols on a network segment that should, under no circumstances, be there anyhow.
- Aggregating by attack type also allows for attack pattern matching and to correlate coordinated attacks against multiple network segments.

By utilizing all of these aggregation methods, the analyst is given an unlimited number of different sets of data to correlate against other attacks, detect coordinated distributed attacks, from within their own network, and to detect new exploits and vulnerabilities being deployed by the underground hacking community.

## 3. Intrusion Detection Message Exchange Format (IDMEF)

As DIDS has many advantages, it also has some difficulties and challenges in implementation. DIDS need the alert correlation of multiple IDS. Interoperability of different type of Intrusion Detection System and alerts correlation is one of the important issues. The Internet Engineering Task Force (IETF) has been working on an intrusion alert data model and accompanying message format standard called the Intrusion Detection Message Exchange Format (IDMEF). One of the main design goals of the IDMEF data model is to be able to express relationships between alerts and alert correlation. The purpose of the IDMEF [7] is to define data formats and exchange procedures for sharing information of interest to intrusion detection and response systems and to the management systems that may need to interact with them. The purpose of the IDMEF is to define data formats and exchange procedures for sharing information of interest to Intrusion Detection and Response Systems (IDRS), and to the management systems that may need to interact with them [6].

## 4. DIDS Prototype Implementation

Goals of our research are to implement a prototype of DIDS that works upon standardized communication and alerting formats. Our approach is to collect the alerts from

HIDS, NIDS or through other sensors installed in the environment in IDMEF format for alert correlation at the central system. If the HIDS, NIDS or the sensors do not support the IDMEF format, then the native alerts format are converted in IDMEF format, store these IDMEF alerts in a central database, and perform analysis on the data both with native SQL queries and custom algorithms. The prototype of DIDS (Figure 1) is composed of a three different sub-networks. In first sub-network, Host Based Intrusion Detection System (HIDS) configured on each nodes. In second sub-network, Network Based Intrusion Detection System (NIDS) is installed. In third sub-network, HIDS and NIDS both is configured. The purpose of our prototype is to correlate the alert generated by the different IDS from different network cluster or host systems and detect the intrusion. As it's briefly discussed in the above section that IDMEF is the standard format that enable interoperability among different types of IDS, we have implemented the prototype using the IDS and log analyzer which support IDMEF. The detail descriptions of these software are as follows:

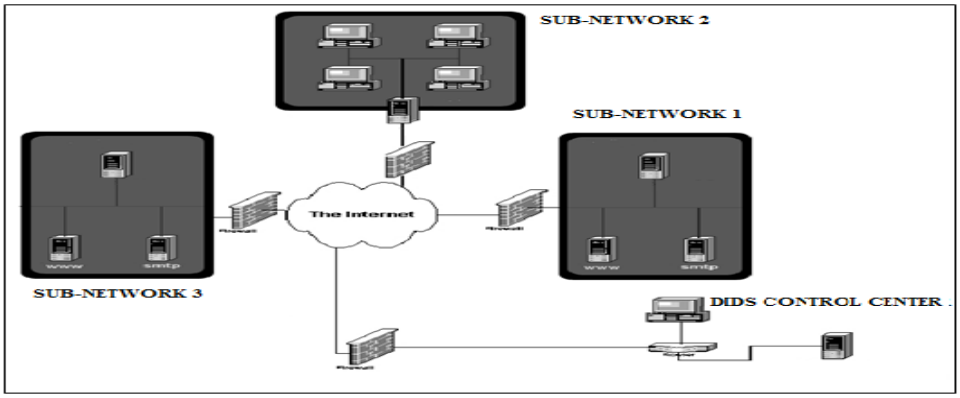



***Fig 1:-*** *Distributed IDS Architecture*

**i) Host Based Intrusion Detection System (HIDS)**

The host-based IDS looks for signs of intrusion on the local host system. These frequently use the host system's audit and logging mechanism as a source of information for analysis. They look for unusual activity that is confined to the local host such as logins, improper file access, unapproved privilege escalation, or alterations on system privileges. In our prototype implementation first sub-network is having three nodes and each nodes are configured with different types of HIDS. The detail of these HIDS are as follows:

**i) OSSEC: -** It's an Open Source Host-based Intrusion Detection System that performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response. OSSEC is installed on Node 1 and Node 2.

**ii) SamHain:-** The SamHain host-based intrusion detection system (HIDS) provides file integrity checking and log file monitoring/analysis, as well as rootkit detection, port monitoring, detection of rogue SUID executable, and hidden processes. SamHain is configured on Node 3.

**ii) SNORT- Network Based Intrusion Detection System (NIDS)**

Snort is open source network-based intrusion detection system (NIDS) has the ability to perform real-time traffic analysis and packet logging on Internet Protocol (IP) networks. Snort can be configured in three main modes: sniffer, packet logger, and network intrusion detection. In sniffer mode, the program will read network packets and display them on the console. In packet logger mode, the program will log packets to the disk. In intrusion detection mode, the program will monitor network traffic and analyze it against a ruleset defined by the user. The program will then perform a specific action based on what has been identified. In our prototype Snort is installed and configured on second sub-network.

**iii) Prelude IDS**

Prelude collects, archives, normalizes, sorts, aggregates, correlates and reports all security-related events independently of the product brand or license giving rise to such events. Prelude IDS uses IDMEF format. It's a good tool to get familiar with IDMEF format as all attribute values are visible from the Graphical User Interface. Prelude is a capable of handling large number of connections, and processing large amounts of alerts. It uses per client scheduling queues in order to process alerts by severity fairly across clients. The Prelude Manager comes with multiple plugins like filtering plugins (idmef-criteria, thresholding, etc.) or reporting plugins like the SMTP plugin which automatically sends emails containing a textual description of alerts to a configured list of recipients. In our prototype we are using prelude to collect the alerts logs from all the HIDS and NIDS for correlation and generating the aggregate alert log.

## 5. Implementation and Performance Analysis

If any activity that triggers an alert by HIDS or NIDS, an alarm is generated by the respective IDS. If the log generated by the IDS is not in IDMEF format it is converted in IDMEF format using plugins. The log is then transferred to the Prelude IDS for alert correlation. The Prelude

collects, normalizes, sorts, aggregates, correlates and generate the log in IDMEF format. The objective of our implementation was to collect the alerts from different types of IDS running on different nodes. Then the native alerts format are converted in IDMEF format and passed it to the central database for alert correlation, in our prototype the alert correlation is done by Prelude. In order to evaluate the performance of implementation, we did the experimental analysis using KDD99 dataset. The results are tabulated in Table 1. The KDD99 dataset was used in the 3rd International Knowledge Discovery and Data Mining Tools Competition for building a network intrusion detector, a predictive model capable of distinguishing between intrusions and normal network connections [11]. It was operated like a real environment, but being blasted with multiple intrusion attacks and received much attention in the research community of adaptive intrusion detection. In KDD99 dataset, each example represents attribute values of a class in the network data flow, and each class is labeled either normal or attack. The classes in KDD99 dataset categorized into five main classes (one normal class and four main intrusion classes: probe, DOS, U2R, and R2L).The results of detection rate and false positives are shown in the table 1, figure 2 and Figure 3.

1 Normal connections are generated by simulated daily user behavior such as downloading files, visiting web pages.
2 Denial of Service (DoS) attack causes the computing power or memory of a victim machine too busy or too full to handle legitimate requests. DoS attacks are classified based on the services that an attacker renders unavailable to legitimate users like apache2, land, mail bomb, back, etc.
3 Remote to User (R2L) is an attack that a remote user gains access of a local user/account by sending packets to a machine over a network communication, which include sendmail, and Xlock.
4 User to Root (U2R) is an attack that an intruder begins with the access of a normal user account and then becomes a root-user by exploiting various vulnerabilities of the system. Most common exploits of U2R attacks are regular buffer overflows, load-module, Fd-format, and Ffb-config.
5 Probing (Probe) is an attack that scans a network to gather information or find known vulnerabilities. An intruder with a map of machines and services that are available on a network can use the information to look for exploits.

| | **Normal** | **Probe** | **DOS** | **U2R** | **R2L** |
|---|---|---|---|---|---|
| DR% | 98.21 | 97.25 | 95.45 | 45.34 | 95.45 |
| FP% | 0.77 | 0.35 | 0.29 | 0.71 | 0.51 |
| ***Detection Rate- DR*** ***False Positive- FP*** | | | | | |

**Table 1:-** Attacks Detection Rates

In our other experiment with the same prototype implementation we wanted to analyze that what's the benefit of using IDS alert log in IDMEF format in terms of size of alerts log. It's one of the important analysis as in distributed environment or in large network the IDS and sensors are installed on multiple hosts and the alerts log generated by these IDS and sensors are communicated to central database for alert correlation. If the size (storage size) of these alerts are big it will consume more bandwidth of the communication network.

***Fig 2:-*** *Intrusion Detection Rate by DIDS*

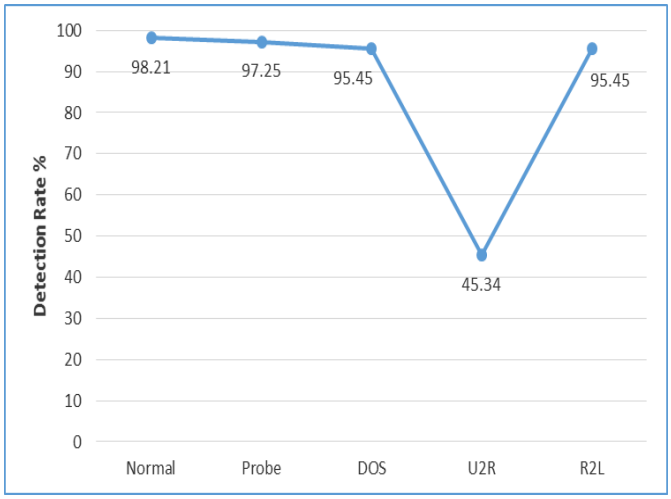
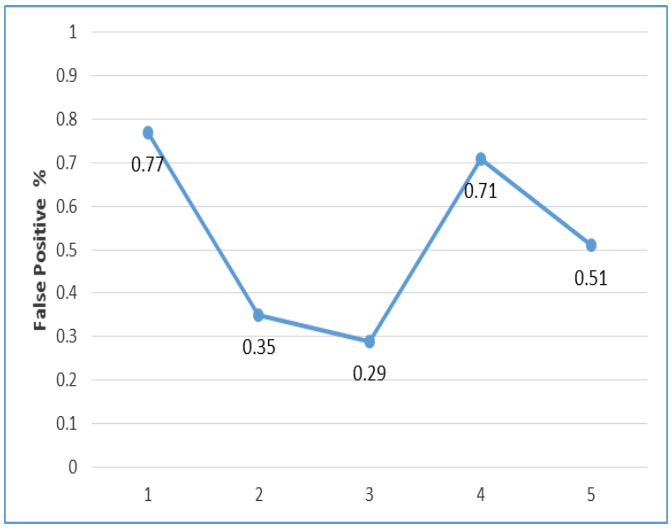
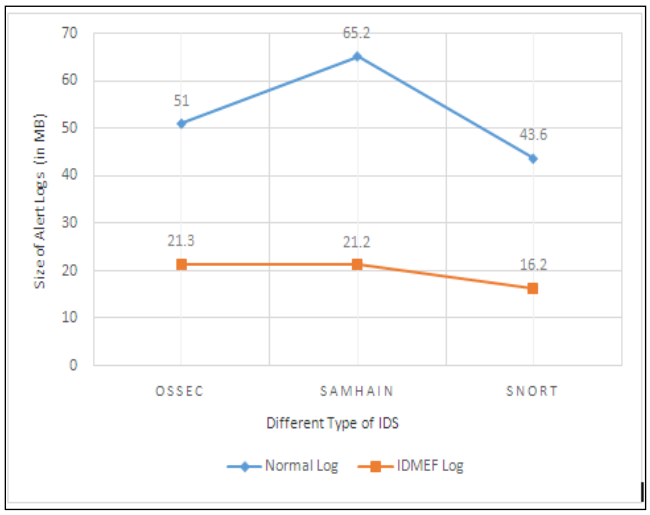



***Fig 3:-*** *False Positive Rate*




In our experiment we collected the 10000 alerts generated by OSSEC, SamHain and Snort in the native format as well as in IDMEF format. The details are shown in the Table 2. The experimental analysis shows that the size of alert logs generated in IDMEF format is on average 2.5 times (Figure 4) less than the logs generated in native format. This clearly shows that such kind of DIDS architecture which uses the log's generated in IDMEF format also save the bandwidth and feasible for realistic implementation.

***Fig 4:*** *- Analysis of Alert Log Size Generated by Different IDS*




| | OSSEC | SamHain | SNORT |
|---|---|---|---|
| **Native Format** | 51 MB | 65.2 MB | 43.6 MB |
| **IDMEF Format** | 21.3 MB | 21.2 MB | 16.2 MB |

**Table 2:-** Alerts Log Size Analysis

## 6. Conclusion

In this paper we have explained the architectural design and performance analysis of a DIDS system. We presented a Distributed Intrusion Detection System for a large scale network environment where multiple IDS exchange the logs in IDMEF format. A prototype implementation has been shown. The system provides ease of management and high detection rate with less false positive ratio. The experimental result was positive and we found that this work can be continued with several other improvement and performance analysis.

## References

[1] A Mouinji, B L Charlier, D Zampunieris, N Habra, "Distributed Audit Trail Analysis", Proceedings of the ISOC 95 Symposium on Network and Distributed System Security", pp. 102- 112, 1995

[2] ApacheTM Hadoop@ homepage, http://hadoop.apache.org/.

[3] Axelsson, Stefan. Intrusion detection systems: A survey and taxonomy. Vol. 99. Technical report, 2000.

[4] E H Spafford, D Zamboni, "Intrusion detection using autonomous agents", Computer Networks, 34, pp. 547-570, 2000

[5] H.Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of Intrusion Detection Systems", Computer Networks, vol 31, n0. 8, pp. 805-822, 1999.

[6] Hadoop, http://hadoop.apache.org/.

[7] Holtz, Marcelo D. ; Bernardo David ; Sousa Jr., R. T. . Building Scalable Distributed Intrusion Detection Systems Based on the MapReduce Framework. Telecomunicacoes (Santa Rita do Sapucai), v. 13, p. 22-31, 2011.

[8] J. Dean and S. Ghemawat, MapReduce: Simplified Data Processing on Large Cluster, USENIX OSDI,2004.

[9] Jeong Jin Cheon and Tae-Young Choe, "Distributed Processing of Snort Alert Log using Hadoop", IJET,Vol 5, No-3, Page 2685-2690, Jun-Jul 2013,

[10] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler, "The Hadoop Distributed File System," IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), pp.1-10, 2010.

[11] Nathan Einwechter," An Introduction To Distributed Intrusion Detection Systems", Security, Endpoint Protection (AntiVirus), SecurityFocus, 2001 (http://www.symantec.com/connect/articles/introduction-distributed-intrusion-detection-systems)

[12] P A Porras, P G Neumann, "EMERALD: event monitoring enabling response to anomalous live disturbances", Proceedings 20th National Information Security Conference, NIST 1997.

[13] R A Kemmerer, "NSTAT: a Model-based Real-time Network Intrusion Detection System", Technical Report TRCS97-18, Reliable Software Group, Department of Computer Science, University of California at Santa Barbara, 1997.

[14] S R Snapp, J Bretano, G V Diaz, T L Goan, L T Heberlain, C Ho , K N Levitt, B Mukherjee, S E Smaha, T Grance, D M Teal, D Mansur, "DIDS (Distributed Intrusion Detection System) – motivation architecture and an early prototype", Proceedings 14th National Computer Security Conference, Washington DC, October, pp. 167-176, 1999.

[15] S Staniford-Chen, S Cheung, R Crawford, M Dilger, J Frank, J Hoagland, K Levitt, C Wee, R Yipi, D Z Erkle, "GriDS – a large scale intrusion detection system for large networks", Proceedings 19th National Information Security Conference, Vol. 1, pp. 361-370, 1996.

[16] S. Ghemawat, H. Gobioff, and S. Leung, The Google file system, ACM SOSP, 2003.

[17] Yeonhee Lee and Youngseok Lee. 2012. Toward scalable internet traffic measurement and analysis with Hadoop. SIGCOMM Comput. Commun. Rev. 43, 1 (January 2012), 5-13. DOI=10.1145/2427036.2427038 http://doi.acm.org/10.1145/2427036.2427038

**Authors Profile**

**Manish Kumar** is working as Assistant Professor in the Department of Computer Applications, M. S. Ramaiah Institute of Technology, Bangalore, India. He is pursuing his PhD from Bangalore University, Bangalore. His specialization is in Network and Information Security. He has worked on the R&D projects related on theoretical and practical issues about a conceptual framework for E-Mail, Web site and Cell Phone tracking, which could assist in curbing misuse of Information Technology and Cyber Crime. He has published many research papers in National, International Conferences and Journals. He is also the active member of various professional societies.

**Dr. M Hanumanthappa** is currently working as Professor in the Department of Computer Science and Applications, Bangalore University, Bangalore, India. He has over 15 years of teaching (Post Graduate) as well as Industry experience. He is member of Board of Studies /Board of Examiners for various Universities in Karnataka, India. He is actively involved in the funded research project and guiding research scholars in the field of Data Mining and Network Security. He has published many research papers in National, International Conferences and Journals. He is also the active member of various professional societies.