An Approach for Collaborative Decision in Distributed Intrusion Detection System

Deepak Kumar Sharma
Assistant Professor
University of Petroleum and Energy Studies
Dehradun, India

Nikhil Kumar Singh
Assistant Professor
U. V. Patel College of Engineering
Ganpat University, Kherva, Gujrat, India

ABSTRACT
Computers have virtually changed every aspect of our life. The rapid growth in the development of computers was focused on making the computer easy to use for all. The rapid growth did not give as much importance on the security of the computer system thereby leaving system as vulnerable to attacks. As internet and its applications are increasing, complex and hybrid networks are being used for communication. So many loopholes are being explored to intrude into other systems. There are many tools and techniques available for securing networks like Firewalls, IDS etc. and until now they are used very frequently by nearly all the organizations to safeguard information and other critical data but these are not sufficient for implementing complete security because the intruders have become smarter.

Higher security being the priority of many organizations has led to the importance and promoting active research on efficient Intrusion Detection Systems. To deal with various types of attacks we need to have information of attacks from other sources as well. This can be done by sharing intrusion information with all. As hackers are becoming more intelligent we need to have collaborative decision making system where intrusion activity is decided by knowing other's opinion as well. We have proposed an approach to enhance the collaborative decision making by conducting polls between registered intrusion detection systems in the network. Intrusion activity for new packets and false positives is decided based on all opinions gathered from registered intrusion detection systems.

General Terms
Intrusion detection systems, collaborative decision making system

Keywords
Distributed IDS, Anomaly detection

1. INTRODUCTION
Information security plays an important role in all aspects of life, in particular the protection of an organization's valuable resources, such as information, hardware, and software. Therefore, information security is defined as a process of protecting data from unauthorized access, use, disclosure, destruction, modification, or disruption. It is concerned with ensuring that information related risks are assessed, appropriate controls are implemented to manage those risks, and that the adequacy of those controls are monitored on a regular basis. Generally, discussion of information security falls under three generic headings:

a) *Confidentiality:* This is a requisite for maintaining the privacy of people whose personal information the organization holds.

b) *Integrity:* This means that data cannot be created, changed, or deleted without authorization. It also means that data stored in one part of a database system are in agreement with other related data stored in another part of the database system (or on another system).

c) *Availability:* This means that the information, the computing systems used to process the information, and the security controls used to protect the information are all available and functioning correctly when the information is needed.

The field of information security has evolved rapidly in recent years because of the swift growth and widespread use of electronic data processing, and also of business conducted through the Internet and other computer networks (LAN, WAN, etc.). These application areas make networks an attractive target for abuse and thus an area of vulnerability. At the same time, the tools of the intruder and the hacker have improved substantially. In order to both combat the growing number of attacks and to maintain critical information services, both academic and industry groups have been developing systems to monitor networks and to raise alarms over suspicious activities. These systems are called Intrusion Detection Systems.

Intrusion Detection is defined as "the problem of identifying individuals who are using a computer system without authorization (i.e., crackers) and those who have legitimate access to the system but are abusing their privileges (i.e., insider attack: threat)" [1]. An Intrusion Detection System gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). An IDS is designed to detect unscrupulous activities that compromise the confidentiality, integrity, or availability of network or computer systems and to analyze what happens – or what has happened – to indicate that the computer has been misused. The IDS does not eliminate the use of a preventive mechanism, but rather works as a second defense mechanism behind a firewall, which can monitor the network while not affecting network performance. In conclusion an IDS is the whole process that detects, audits, tracks, and identifies unauthorized access and abnormal phenomena actions or events in the system. It can identify whether the system is being accessed as it happens and take the appropriate actions to cut off network connections, record events, and raise an alarm. It can also remind the system administrators to take proper measures. More details on IDS are given in the next chapter.

Recently, a number of innovative approaches and new models for IDS have been proposed. But while many of the proposed

techniques have relatively improved some of the shortcomings of the earlier approaches, still a number of issues remain: low detection accuracy, low real-time performance, and limited scalability. These problems make the area of IDS an attractive and open research field. In recent years, researchers have investigated a variety of different computational tools to improve IDS performance and overcome some of its limitations, such as Soft Computing (SC) techniques [2], [3], [4], distributed systems [5], [6], [7], and autonomous agents (AA) [8], [9]. Still, a lot more needs to be done to deal with new technologies and tools developed by intruders to break the systems.

In this paper, we have overcome some of IDS limitations by proposing new collaborative decision making approach CDdIDS in distributed IDS architecture.

1.1 Problem Statement

The field of information security has grown and evolved substantially in recent years because of the rapid growth and widespread use of electronic data processing, and of business conducted through the Internet and other computer networks (LANs, WANs, etc.). These application areas make networks attractive targets for abuse. At the same time, the tools of the intruder and the hacker have improved substantially. Facing these daunting challenges, industry and academic institutions are working hard to develop new devices, new approaches, and new security mechanisms to counter the challenges from malicious intruders. These efforts have resulted in a great variety of security products such as firewalls, encryption, authentication, vulnerability checking, and other measures. Nevertheless, most computer systems are still susceptible to attacks from hackers, so it is essential to establish a second line of defense for these systems in the form of an Intrusion Detection System (IDS).

IDS [10], [11] play an important role in achieving the survivability of information systems and ensuring their safety from attacks. They aim to protect the availability, confidentiality, and integrity of critical network information systems by analyzing what happens or has happened during an intrusion, and attempting to identify signs that a computer has been misused. They can also take appropriate actions to sever network connections, record events, raise alarms, and remind system administrators to take proper measures.

IDS are usually classified as host-based or network-based. Host-based systems [12], [13], base their decisions on information obtained from a single host (usually log files, network traffic to and from the host, or information on processes running on the host), while network-based systems [20] obtain data by monitoring network traffic between hosts, and are usually run on a separate machine. Most current IDS technology still suffers from three main problems which limit their detection ability: low detection accuracy (registering high False Positive alarms and False Negative); low real-time performance (processing large amounts of traffic data in real time); and limited scalability (storing a large number of user profiles and attack signatures).

Our proposed approach overcomes these limitations by having a collaborative and distributed architecture for the IDS. Another key effort in our approach is that directed towards improving system robustness, extensibility, configurability, and security.

1.2 Motivation and Contributions

The ideal approach for computer security is to establish and implement a security policy that prevents any intrusion through the use of security measures. However, traditional preventive measures are not always sufficient, for the following reasons:

- Bug-free software is seldom attainable.
- It is difficult to change user and organization behavior, to oblige all users to follow diligently security policy.
- Human errors in operations and maintenance are unavoidable; these errors can cause serious security loopholes.
- The security measures and controls themselves can be compromised: for instance, the cryptographic algorithms can be cracked, given sufficient time and computing power.
- It is almost impossible to prevent insider attacks because inside users naturally have greater access to the system than do outside attackers.
- The cost of setting up a totally secure system is very high, which discourages their implementation. Because of the above difficulties, we need to use other alternative or complementary techniques to protect and secure our systems. One of the major techniques is the Intrusion Detection System (IDS).

Intrusion Detection is another type of security tool that must be created to protect and secure the information resources in the system. It complements firewalls by allowing a higher level of analysis of traffic on a network, and by monitoring the behavior of the sessions on the servers. In addition, it possesses some special characteristics and benefits as:

- Networks are complex and difficult to monitor: an IDS can help reveal potential network security problems by documenting the network status.
- An IDS highlights intrusion traces, which help to identify and eliminate the security flaws that enabled these intrusions in the first place.
- An IDS can assess the integrity of critical system and data files.
- An IDS provides real-time reporting of break-ins, allowing the system administrator to take immediate action, lessening potential damage.
- In contrast to a firewall, an IDS is a passive system that does not influence network traffic.

Thus, most people attacking or trying to circumvent a system will not recognize the intrusion detection node. In addition, an authorized user can log on without interruption. The current state of IDS technology is not yet fully reliable, which makes the area of IDS an attractive and still open research field. A major problem with current IDS is their inability to guarantee intrusion detection (low accuracy): the current IDS technology is not accurate enough to provide reliable detection. This problem will lead to a high rate of false alarms (False Positives), and missed alarms (False Negatives). A common complaint is that the large number of False Positives and Negatives generated by Intrusion Detection Systems makes it hard to filter out false attacks without potentially missing genuine attacks. Moreover, this low accuracy can lead to an incident handling problem: that is, security administrators are uncertain how to respond to mitigate the

risks if a certain degree of accuracy cannot be achieved. There is no decision rule associated with each alert to tell the security administrator whether he should ignore the alert or simply terminate the suspicious session.
Another major problem is the speed of detection (low efficiency). The size of the feature space is obviously very large, which leads to slow training and testing processes, heavy computational resources, and low detection accuracy. Moreover, computer networks have a dynamic nature in the sense that the data within them are continuously changing. Therefore, in order to detect an intrusion accurately and promptly, the system has to operate in real time. In addition to the problems outlined above, there are some other limitations, such as:

a) *Inability to detect new attacks:* The ability to recognize new attacks when they are launched for the first time is very low; this reduces the overall system performance.

b) *Limited scalability:* The IDS is unable to achieve reliable scalability to gather and analyze the high volume of audit data correctly from the distributed host, which may cause severe network performance degradation.

c) *Lack of extensibility:* It is difficult to extend the scope of IDS or reconfigure/add capabilities to the IDS.

d) *Difficult configurability:* The IDS is unable to configure itself easily to the local requirements of each host or each network component.

e) *Monotonic analysis:* Many network intrusions exploit the multiple points of a network. Thus, from a single host, they might appear to be just a normal mistake. But if they are collectively monitored from multiple points, they can be clearly identified as a single attack attempt.

f) *Low robustness:* In many cases, the IDS itself may fall under attack from a threat seeking to disable it. An IDS should itself be resistant to attacks, should exhibit a high degree of fault tolerance, and allow for graceful degradation.

g) *Low reliability (Point of Failure):* For most single IDS, if an intruder can somehow prevent the IDS from working, the whole network is without protection.

Recently, a number of innovative approaches and new models for IDS have been proposed to improve IDS efficiency and performance, such as Distributed IDS (dIDS). The dIDS [21], [22] is one of several options that allow the computation load and diagnostic responsibilities to be distributed throughout the network. It performs distributed data collection (and some pre-processing) by using modules distributed in different hosts, which monitor separately and communicate and cooperate with each other. The dIDS can provide the foundation for a complete solution to the complexities of real time detection, while maintaining fault-resistant behavior. It has the scalability to detect general attacks or a specific attack. In addition, each module can be added to or removed from the system without altering other system components, because they operate independently. Also, the system's modules can be configured or upgraded without disturbing the rest of the system, as long as their external interface remains the same.

Another approach used to improve IDS efficiency is Soft Computing (SC). In general, applications of SC are widely used by IDS, either for a detection model or for the generation of intrusion features selection. They are suitable for handling such subjective estimates for a number of reasons:

- Fast recognition and classification;
- Learning abilities;
- Adaptability;
- Flexibility;
- Low solution cost;
- Fast computing;
- Ease of design;
- The ability to generalize from learned data;
- Not easily misled by small variations in intrusion patterns;
- Modular with both misuse and anomaly detection components.

2. LITERATURE SURVEY

With the increasing connectivity and complexity of heterogeneous computer systems, it is likely unrealistic to expect that an IDS should be capable of correctly classifying every event that occurs on a given system. In addition, there are the limitations of centralized IDS, such as: a single point of failure; limited scalability; frequent overload; vulnerability to subversion; and difficulty in configuring or adding capability to the IDS.

2.1 Centralized IDS

The current research directions in detecting coordinated attacks using CIDSs are summarized in [17]. In particular, two main challenges in CIDS research: CIDS architectures and alert correlation algorithms are highlighted and analyzed.

In [18], a decentralized, multi-dimensional alert correlation algorithm for CIDSs is proposed. A two-stage algorithm, implemented in a fully distributed CIDS, first clusters alerts locally at each IDS, before reporting significant alert patterns to a global correlation stage.

An IDS should consist of multiple entities working independently to cover the huge amount of data and traffic in the system, and should allow changes to these entities without any modifications made to other entities; this is accomplished by using an IDS with distributed architecture.

2.2 Distributed IDS

Distributed IDSs (dIDSs) are based on distributed IDS entities located in different locations within the network, which monitor separately and communicate and cooperate with each other. The dIDS allows computation load and diagnostic responsibilities to be distributed throughout the network. It can provide the foundation for a complete solution to the complexities of real-time detection, while maintaining fault tolerance behaviour. It allows early detection of planned and coordinated attacks, thereby allowing network administrators to take preventive measures. dIDS also helps to control the spreading of worms, improves network monitoring, incident analysis, attack tracing and so on. Also, it has the scalability to detect general attacks or a specific attack, in addition to providing significant advantages in flexibility, extendibility, and resistance to compromise.

A number of distributed IDS have been proposed for a distributed environment. Early systems included dIDS [5], NADIR (Network Anomaly Detector and Intrusion Reporter) [14], CSM (Cooperative Security Managers) [19], EMERALD (Event Monitoring Enabling Response to Anomalous Live Disturbances) [20], AAFID (Autonomous Agents for Intrusion Detection) [8], CIDF (Common Intrusion Detection Framework) [21] and MAIDS (Mobile Agent Intrusion Detection System) [22]. The rest of this section briefly introduces some of these projects.

dIDS [5] incorporates Haystack and NSM (Network Security Monitor) in its framework. This system requires the audit data collected from different places to be sent to a central location for analysis. The DIDS operates on a local area network (LAN) and consists of three major components: the host monitor, the LAN monitor, and the central manager. Each host is monitored by a host manager. This manager is a collection of processes running in the background of the host. Also, each LAN is monitored by a LAN manager, which operates just like a host manager except that it analyzes LAN traffic. Finally, there is a central manager which is placed at a single secure location and controls the entire system. This central manager receives reports from various host and LAN managers and by processing and correlating these reports, it detects intrusions. The DIDS itself is not fully distributed because it relies on both distributed and centralized resources to detect intrusions. This technology faces a number of challenges such as its centralized nature, arbitrary definitions of abnormal activities, and ineffective coordination between the DIDS modules.

The NADIR system [14] performs distributed data collection by employing the existing service nodes in the Los Alamos National Laboratory's Integrated Computer Network (ICN) to collect audit information. The NADIR examines the network traffic at the service and protocol level by using a statistic-based anomaly detector and an expert system, which is then analyzed by a central expert system. The major drawback of NADIR is its centralized analysis, which severely limits the scalability of the detection algorithm. Moreover this system, NADIR, would not easily be ported to an internetworked environment with many heterogeneous systems.

The CSM [19] are employed to perform dIDS that does not need a hierarchical organization or a central coordinator. Each individual CSM detects malicious activity on the local host. When suspicious activity is detected, each CSM will report any noteworthy activity to the CSM on the host from which the connection originated. The local CSM will not notify all networked systems, but rather only the system immediately before it in the connection chain. The architecture of the system allows for CSM to take reactive actions when an intrusion is detected. Unclear aspects are the mechanisms through which CSM can be updated or reconfigured, and the intrusion detection mechanisms that are used locally by each CSM.

EMERALD [20] is intended as a framework for distributed, interoperable computer and network intrusion detection. It employs entities called service monitors that are deployed to host and perform monitoring functions. They define several layers of monitors for performing data reduction in a hierarchical fashion. Monitors can be programmed to perform any function. However, this model does not scale well for large networks. The large number of events and devices distributed across the network can generate too much network traffic and too much data to be stored in one location efficiently. It also does not cover distributed services (e.g., DNS, firewalls).

AAFID [8] is a distributed intrusion detection architecture and system, developed in CERIAS at Purdue University. It is agent-based, employs a hierarchical structure and the data are collected and analyzed locally. Nevertheless, there is still a highest-level entity in the AAFID architecture, which is the bottleneck of this system and leads inevitably to the matter of a single point of failure. Also, if the two or more IDS that are far part in the hierarchy detect a common intruder, the two detections cannot be correlated until the messages from the different IDS reach a common high-level IDS. This will require the messages to traverse multiple IDS resulting in communication overheads. In addition, it has limited scalability, performance, user interface and security.

CIDF [21] was an effort to standardize intrusion detection to some degree by enabling different intrusion detection and response components to inter operate and share information and resources in a distributed environment. The intrusion detection inter-component adaptive negotiation protocol helps cooperating CIDF components to reach an agreement on each other's needs and capabilities.

MAIDS [22] are also typical distributed IDS. It is an end-to-end procedure for intrusion detection. Known vulnerabilities of a system are expressed in an abstract "Software Fault Tree" (SFT), then converted to a Colored Petri Net (CPN), and finally into a system of independent agents. These systems suffer from a number of problems such as a lack of an effective coordination mechanism to detect a complicated attack, and the security of the system itself is almost unconsidered.

Paper [23] presents a collaborative architecture for multiple IDSs to detect real-time network intrusions. The architecture is composed of three parts: Collaborative Alert Aggregation, Knowledge-based Alert Evaluation and Alert Correlation to cluster and merge alerts from multiple IDS products to achieve an indirect collaboration among them.

The research on dIDS [9], [6], [17], [18], [24] is a rapidly growing area of interest because the existence of dIDS techniques is increasingly unable to protect the global distributed information infrastructure. So, the existing dIDS must be updated and improved constantly to adapt to the ever-changing environment and they should be studied in greater depth in order to ensure better system security.

3. PROPOSED WORK

Generally large networks are divided into smaller sub-networks having simple/hybrid topologies. These sub-networks will have its own IDS to monitor network traffic. All the new intrusion systems (IDS/IPS) need to register themselves with existing IDS/IPS group to be part of this distributed and collaborative decision making system.

3.1 Signatures

Each IDS/IPS registered with the group will maintain two classes of signature rules as defined below:

Positive Signatures – this indicates packet/source is suspicious. All the network traffic will be compared against these signatures and if a match is found, alarm for intrusion activity will be sent to the administrator.

Negative Signatures – this indicates packet/source is reliable and is known to it. All the network traffic will be compared

against these signatures and if a match is found, it will be notified as safe.

After successful registration, initially it will update its signature database for both classes from any of the registered IDS/IPS. All individual IDS will look for new signatures. If it finds any new signature or case of false positive, the administrator will broadcast poll message to all the registered IDS. Poll message will contain (source address, packet details).

3.2 Polling

As we know that in a network there could be heterogeneous environment and IDS/IPS having different software/hardware configurations. To deal with this situation we can have a poll and a response message in XML format, a well known universal data sharing language.

Packet details will be sent in XML format. An example is given below for TCP packet (**Fig 1**):

```
<poll>
<protocol>TCP</protocol>          // Protocol used
<SIP> 202.119.81.182</SIP>       // Source IP address
<DIP>192.16.8.18</DIP>           // Destination IP address
<SP>1880</SP>                     //Source Port
<DP>3128</DP>                     //Destination Port
<SMAC> </SMAC>                    //Source MAC address
<date_time>10-05-2013 22:12:20</date_time> //Time stamp
</poll>
```

Fig 1: Poll Message in the form of XML

On receiving any poll message IDS will extract the packet details and compare with its both signature databases. Based on the type of signature found it will generate response message.

There will be three types of response messages that a source IDS can get back from other registered IDS/IPS:

- Positive response
- Negative response
- Neutral response

3.2.1 Positive response

On comparing packet information with both classes of signatures it was found in positive signature class (suspicious). An example of the response message for the suspicious category of message is given below (**Fig 2**).

```
<Response>
   <Rtype>positive</Rtype>
            <poll>
 <protocol>TCP</protocol>
<SIP> 202.119.81.182</SIP>
<DIP>192.16.8.18</DIP>
<SP>1880</SP>
<DP>3128</DP>
<SMAC>00:2d:cb:ca:5b:38</SMAC>
<date_time>10-05-2013 22:12:20</date_time>
              </poll>
</Response>
```

Fig 2: Positive response message

3.2.2 Negative response

On comparing packet information with both classes of signatures it was found in negative signature class (reliable). An example of the response message for known safe packets class, is given below (**Fig 3**)

```
<Response>
   <Rtype>negative</Rtype>
            <poll>
 <protocol>TCP</protocol>
<SIP> 202.119.81.182</SIP>
<DIP>192.16.8.18</DIP>
<SP>1880</SP>
<DP>3128</DP>
<SMAC>00:2d:cb:ca:5b:38 </SMAC>
<date_time>10-05-2013 22:12:20</date_time>
              </poll>
        </Response>
```

Fig 3: Negative Response

3.2.3 Neutral response

On comparing packet details it was not found in any of the classes. The intrusion detection system has no information about that packet. Example of the response message for this category is given below (**Fig 4)**

```
<Response>
   <Rtype>neutral</Rtype>
            <poll>
<protocol>TCP</protocol>
<SIP> 202.119.81.182</SIP>
<DIP>192.16.8.18</DIP>
<SP>1880</SP>
<DP>3128</DP>
<SMAC>00:2d:cb:ca:5b:38 </SMAC>
<date_time>10-05-2013 22:12:20</date_time>
              </poll>
 </Response>
```

Fig 4: Neutral Response message

On getting back results from different IDS/IPS it will count the Rtype of response messages to get three values.

- Count of positive response,
- Count of negative response and
- Count of neutral response

The registered IDS group will have some predefined threshold value for deciding the new class of the packet. Based on the threshold value by the group, class of the packet will be decided. This will further decide whether the packet should be dropped or allowed in further communication. One of the deciding criteria is given below:

3.3 CDdIDS Algorithm

The proposed CDdIDS Algorithm is given below which is used for deciding class and operation to perform after polling.

1. If (count(positive) >0)
2. start
3.If(count(positive)>Threshold&&count(positive)>count(negative))
4. start
5. Add packet details in the positive class
6. Block all activities from this source.
7. end
8.Elseif(count(positive)<Threshold&&count(positive)<count(negative))
9. start
10. Add packet details in negative class.
11. Allow packets.

12. end
13. Else
14. start
15. If(Count(positive)>Threshold)
16. Block packets.
17. Else
18. Allow packets but require further monitoring
19. end
20. end
21. Else If (count(positive)=0 && count(negative)>0)
22. start
23. If(count(negative)>Threshold)
24. start
25 Add packet details in negative class
26 Allow packets.
27. end
28. Else
29. Allow packets
30. end
31 Else
32. Allow packets but require further monitoring

Here count() is the function which returns the total number of packets of type positive or negative or neutral depending on the parameter passed.

Threshold is the value decided by all Registered IDS/IPS together. Threshold value needs to be chosen appropriately depending on how much accuracy is required to decide new class of packet.

4. RESULTS AND ANALYSIS

As the precision of selecting class increases with higher value of threshold we can conclude that higher the threshold value, higher will be the probability of choosing the correct class of new packet. But with higher threshold value number of decisions will be very less. So threshold value need to chosen based on network priorities and requirements.
In this paper we have considered only 2 IDS systems. The First Host machine (IDS) successfully captured packets in communication between two virtual machines. The second IDS system successfully responded back to the host IDS.

Maintaining signatures for 2 different classes is difficult as we generally have different IDS/IPS hardware/software for different organizations. Here based on filtering settings we successfully added signatures of new packet in host IDS.

This approach reduces false positive cases by communicating with other IDS/IPS. For the new packets or false positives, the administrator will have to observe manually for conducting polling and updating.

On the basic of results obtained we can say that, our proposed CDdIDS approach reduces the false negative rate and enhances true positive rate and the attach detection capability is higher than the approaches proposed by [9], [24].

5. CONCLUSION

For the problems in the distributed intrusion detection system model and the need in practical application, this paper puts forth a distributed collaborative intrusion detection system model on the basis of the existing typical distributed intrusion detection system model. This paper also puts out that the collaborative analysis should be adopted in the distributed intrusion detection system and that the information about various system security components should be used to maintain the information safety of the system. At last, the experiment and simulation in this paper demonstrate that there is obvious improvement in the detecting rate of the system and obvious decrease in false alarm rate. Meanwhile, the collaborative analysis can spot new attack type and improve the attack detection. In future a self-organized framework can be developed for collaboration of multiple IDS systems.

6. REFERENCES

[1] R. Heady, G. Luger, A. Maccabe, and M. Servilla, "The architecture of a network level intrusion detection system", Technical Report CS90-20, Department of Computer Science, University of New Mexico, August 1990.

[2] L. Silva, A. Santos, J. Silva, and A. Montes, "A Neural Network Application for Attack Detection in Computer Networks", In the proceedings of the IEEE International Joint Conference on Neural Network, Vol. 2, pp. 1569 – 1574, July 2004.

[3] C. Zhang, J. Jiang, and M. Kamel, "Intrusion Detection using Hierarchical Neural Networks", Pattern Recognition Letters, Vol. 26, No. 1, pp. 779–791, 16 February 2004.

[4] A. Sung and Srinivas Mukkamala, "Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks", Symposium on Application and Internet (SAINT'03), pp: 209- 216, 27-31 January. 2003.

[5] S. Snapp, J. Brentano, G. Dias, T. Goan, T. Grance, L. Heberlein, C. Ho, K. Levitt, B. Mukherjee, D. Mansur, K. Pon, and S. Smaha, "A System for Distributed Intrusion Detection [C]", Proceedings of the 14th Conference on National Computer Security Conference, Vol.9, pp. 170-176, March 1991.

[6] M. A. Aydin, A. H Zaim, and K. G. Ceylan, Feb. 2009, "A hybrid intrusion detection system design for computer network security," In the journal of Computers and Electrical Engineering, Vol. 35 , No. 3, pp. 517-526, 2009.

[7] S. Snapp, J. Brentano, and G. Dias, "DIDS (Distributed Intrusion Detection System) – motivation, architecture, and an early prototype", In the proceedings of the 14th National Computer Security Conference, October 1991.

[8] E. Spafford and D. Zamboni, "Intrusion Detection using Autonomous Agents", In the International Journal of Computer and Telecommunications Networking, pp. 547-570, 2000.

[9] D. Ye, W. Hui-Qiang, and P. Yong-Gang, "Design of A Distributed Intrusion Detection System Based on Independent Agents", In the proceedings of International Conference on Intelligent Sensing and Information Processing, pp. 254 – 257, 2004.

[10] Th. Verwoerd and R. Hunt, "Intrusion Detection Techniques and Approaches", Journal in Computer Communications, pp. 1356-1365. 2002.

[11] B. Mukherjee, L.T. Heberlein, and K.N. Levitt, "Network Intrusion Detection" IEEE Network, pp. 26-41, Vo1.8, No.3, May-June 1994.

[12] P. Lichodzijewski and A. Zincir, "Host-Based Detection Using Self-Organizing Maps", In the Proceedings of International Joint Conference on Neural Networks, Vol. 2, pp. 1714-1719, 2002.

[13] M. Yasin and A. Awan, "A Study of Host-Based IDS using System Calls", In the proceedings of the International Conference on Networking and Communication 2004, pp. 36- 41, June 2004.

[14] J. Hochberg, K. Jackson, C. Stallings, J. McClary, D. DuBois, and J. Ford, "NADIR: An Automated System for Detecting Network Intrusion and Misuse", In the proceedings of the Conference on Computers and Security, pp. 235–248, May 1993.

[15] M. Treaster, "A Survey of Distributed Intrusion Detection Approaches", ArXiv Computer Science e-prints: cs/0501001. December 2005, Available at: http://arxiv.org/abs/cs/0501001 (March 2009)

[16] R. Robbins, "Distributed Intrusion Detection Systems: An Introduction and Review", GSEC Practical Assignment, version 1.4b, Option 1, January 2, 2002.

[17] C. V. Zhou, C. Leckie, and S. Karunasekera, Feb. 2009, "Decentralized multidimensional alert correlation for collaborative intrusion detection," Published by Elsevier Ltd. Journal of Network and Computer Applications 32 (2009), pp. 1106-1123.

[18] C. V. Zhou, C. Leckie, and S. Karunasekera, June, "A survey of coordinated attacks and collaborative intrusion detection ," Jonoural of Computer Security, Vol. 29, No. 1, pp. 124–140, 2010

[19] H. Debar, "An Introduction to Intrusion-Detection Systems", In the proceedings of Connect, May, 2000.

[20] G. White, E. Fisch, and U. Pooch, "Cooperating Security Managers: A Peer-Based Intrusion Detection System", IEEE Network, Vol. 10, No. 1, pp. 20–23, January/February 1996.

[21] P. Porras and P. Neumann, "EMERALD: Event monitoring enabling responses to anomalous live disturbances", In the proceedings of the 20th National Information Systems Security Conference, 1997.

[22] S. Staniford-Chen, S. Tung, D. Schnackenberg, "The Common Intrusion Detection Framework (CIDF)", In the proceedings of the information survivability workshop, October 1998.

[23] Y. Wu, B. Foo, Y. Mei, and S. Bagchi. "Collaborative Intrusion Detection System (CIDS): A Framework for Accurate and Efficient IDS", In the proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 03), pp. 234 – 244, December 2003.

[24] Chao Shen, Shengjun Xue,2010, "Design and Implementation of Distributed Collaborative Intrusion Detection System Model", In the proceeding of Fuzzy systems and Knowledge Discovery, pp. 1224-1228.