# A Real Time DOS Attack Detection in IP Networks Based on Bandwidth Utilization Pattern and Rule Based Pattern Matching

[1]Revathi Cherukuri, [2]Thadoor Shobha Rani, [3]Challa Madhavi, [4]Dr.Manjunath Gadiparthi
[1,2,3,4]Dept. of CSE, JNIT, Hyderabad, India

## Abstract

An DOS(denial of service) is an attempt by unauthorized processes or users to use the system resources like bandwidth thus denying the fare access of the valid peer. DOS is sub type of Intrusion. Various types of Such Intrusion detection systems are Proposed. Such systems are basically depending upon checking the network behavior and matching the access pattern with a predefined rule pattern. The intruders introduces techniques to break the firewall and such rules. Hence Heirarchial pattern matching sches are proposed. But such schems suffers from matching overhead. In this work we detect the intrusion based on Bandwidth usage Pattern analysis combined with protocol headers pattern matching of the packets that are being exchanged from the system with the internet or network. The system comprises of mainly two component: a Monitor which senses and extracts the packet information from the packets being exchanged, classifier: classifies the packets as being intruding and non intruding and performance analyzer to analyze the system. The system is tested in a real time network an intruding system which attacks another system resources as deniel of service attack.Performance shows significant quick and efficient detection and the detection time is merely. 2 milliseconds. False detection rate is 0%.

## Keywords

Intrusion Detection System, IP Security, Bandwidth.

## I. Introduction

### A. Conventional Attacks over IP

An intrusion is the phenomenon of unauthorized access of the resources. In the context of an IP network consider the following fig. 1.

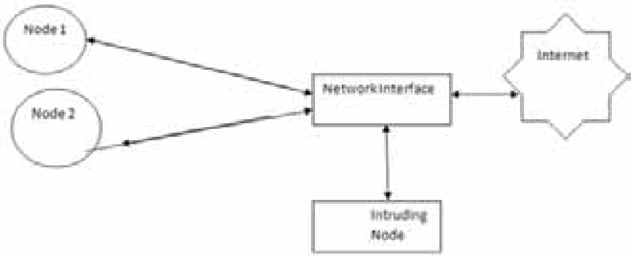

Fig. 1: A typical architecture of the intrusion system.

The fig. explains how the intrusion occurs. In a network few valid nodes access the internet exchanging various types of application specific IP packets. An intruder node is a node which sits between the interfaces of the valid nodes.

The function of intrusion is divided into mainly two categories:

#### 1. Resource Intrusion

Resource intrusion is also called a denial of service. This is a mechanism by means of which the intruding node access the interfaces of the valid nodes to send and receive packets from and to internet. Though this type of attack is considered to be less severe than the second category discussed bellow, in some extent the attack causes many practical problem like bandwidth bottleneck which minimizes the overall resources available to the valid nodes.

#### 2. Eavesdropping

This is nothing but sniffing or reading the packets transmitted between two authenticated or valid nodes by intruding nodes. The process may extract payload from the packets to extract information exchanged between the peers. Though both the attacks are performed through same mechanism, it is important to note that several SSL(secured Socket Layer) and Encryption algorithms are already developed for secured data exchange between the nodes. Therefore in this paper the other type of attack is emphasized onto which is DOS attack and a mechanism is proposed to detect such attacks in real time.

### B. Anomaly Detection System

The basic principal of the work is to distinguish the behavior of the intruding nodes from that of the normal behavior. If a normal profile can be established than easily the intrusion pattern can be recognized. In this sub section we discuss about anomaly detection in detail for better understanding of the proposed work. Anomaly detection techniques assume that all intrusive activities are necessarily anomalous. This means that if we could establish a "normal activity profile" for a system, we could, in theory, flag all system states varying from the established profile by statistically significant amounts as intrusion attempts. However, if we consider that the set of intrusive activities only intersects the set of anomalous activities instead of being exactly the same, we find a couple of interesting possibilities: (1) Anomalous activities that are not intrusive are flagged as intrusive. (2) Intrusive activities that are not anomalous result in false negatives (events are not flagged intrusive, though they actually are). This is a dangerous problem, and is far more serious than the problem of false positives. The main issues in anomaly detection systems thus become the selection of threshold levels so that neither of the above 2 problems is unreasonably magnified, and the selection of features to monitor. Anomaly detection systems are also computationally expensive because of the overhead of keeping track of, and possibly updating several system profile metrics. Some systems based on this technique are discussed in Section 4 while a block diagram of a typical anomaly detection system is shown in Fig. below.

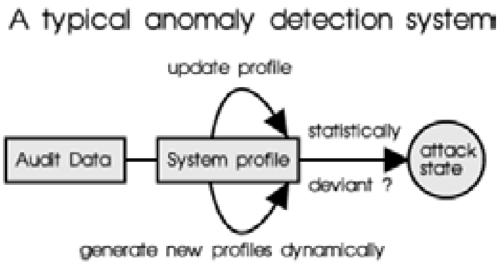

Fig. 2: A typical anomaly detection system

### C. Contribution

Various papers have proposed various techniques of intrusion detection system, including DOS attack detection system. Most of these techniques are illustrated and elaborated with the help of simulators. But in a real time network, many parameters varies in a great deal in comparison to the simulating environment. Hence we propose an intrusion detection system based on real time anomaly detection scheme based on the bandwidth observation. The system is developed using java and can be adopted flawlessly without any significant changes in any real network.

## II. Related Work

Rong-Tai Liu, Nen-Fu Huang, Chia-Nan Kao, Chih-Hao Chen, Chi-Chieh Chou, presented FNP2, an efficient pattern-matching engine designed for Network Processor platform which conducts matching sets of patterns in parallel and showed that combining their string matching methodology, hashing engine supported by most Network Processors, and characteristics of current Snort signatures frequently improves performance and reduces number of memory accesses compared to current NIDS pattern matching algorithms. Besides total number of searching patterns, shortest pattern length is also a major influence on NIDS multipattern matching algorithm performance [1].

Yi Tang, Junchen Jiang , Xiaofei Wang , Bin Liu and Yang Xu extended the classic longest prefix principle from single-character to multi-character string matching and proposed a multi-string matching acceleration scheme named Independent Parallel Compact Finite Automata (PC-FA), also showed that seven times of speedup can be practically achieved with a reduced memory size than up-to-date DFA-based compression approaches [2].

Mohammad A. Alia, Adnan A. Hnaif, Hayam K. Al-Anie, Khulood Abu Maria, Ahmed M. Manasrah, M. Imran Sarwar proposed a novel algorithm to detect the intruders, who's trying to gain access to the network using the packets header parameters such as; source/destination address, source/destination port, and protocol without the need to inspect each packet content looking for signatures/patterns [3].

Lambert Schaelicke, Thomas Slabach, Branden Moore, and Curt Freeland worked on measures and compares two major components of the NIDS processing cost on a number of diverse systems to pinpoint performance bottlenecks and to determine the impact of operating system and architecture differences [4].

Miyuki Hanaoka , Kenji Kono, Toshio Hirotsu, and Hirotake Abe proposed Brownie, a system for improving performance by coordinating configurations of alreadyexisting, independently-managed NIDSs in an organization.Brownie achieves performance improvement by 1) offloading overloaded NIDS, and 2) eliminating redundant rules [5].

M. Nourani and P. Katta presented a hardware architecture for string matching and their solution based on using a Bloom filter based pre-processor and a parallelized hashing engine is capable of handling wire line speeds with zero false-positive probability and also showed the system is capable of matching 16000 strings and achieves in excess of 100 Gbps throughput [6].

C. Jason Coit, Stuart Staniford, Joseph Mcalerney described the effectiveness of a significantly faster approach to pattern matching in the open source NIDS Snort [7].

Hongbin Lu, Kai Zheng, Bin Liu, Xin Zhang, and Yunhao Liu proposed a memory-efficient multiple-character-approaching architecture consisting of multiple parallel deterministic finite automata (DFAs), called TDP-DFA [8].

Amitava Biswas, Purnendu Sinha implemented an user space network interface (DMA ring) to capture packets under high network load on a modest commodity platform [9].

D.Jeyabharathi, D.Sasireka and D.Kesavaraja implemented a simulation tool to handle intrusion attacks in Mobile Ad Hoc Network (MANET) [10].

## III. Proposed Work

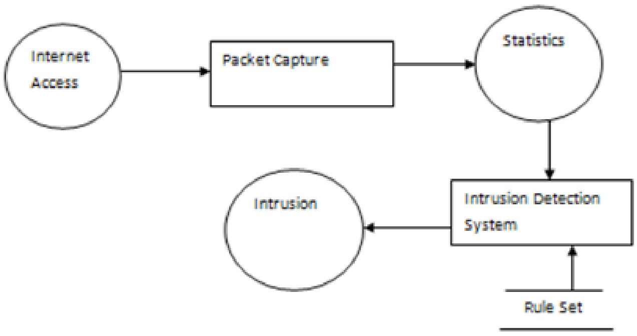

Fig. 3: Simplified DFD of the proposed system.

The diagram explains the overall system. A Packet sniffer interface is designed to sniff all the accessible ports of all the open network interfaces. The statistics like type of packet and port pair, Bandwidth are used to measure intrusion by a separate intrusion detection system. This system detects the anomaly in the access pattern by comparing the result with standard rule set specially designed for the work.

The core of the detection system is a packet capture interface which is developed using JpCap-WinPcap library. WinpCap is a windos interface for communicating and monitoring the network interfaces and devices. Jpcap is the java extension that utilizes the WinpCap functionality. The interface monitors all the packets that are exchanged over the network. It then classifies the access into normal bandwidth and abnormal bandwidth. Based on this detection, intrusion is detected.

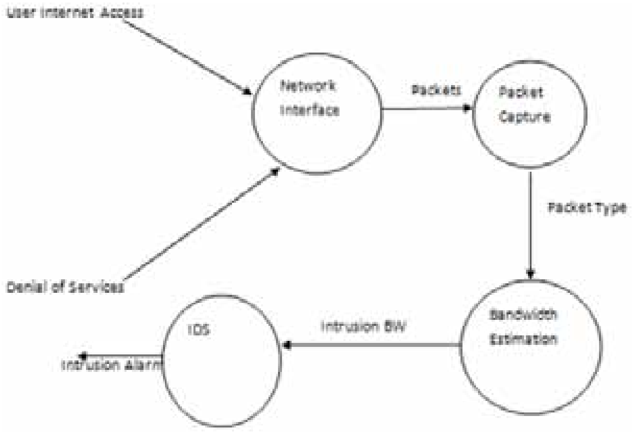

Fig. 4: Detailed DFD of the proposed system.

For validation and verification of the efficiency of the system, the intruding system is also designed by us. The intruding system here is a background process that connects to any of the web sites and transmits data ( either ICMP, or TCP or IP) with the website. Now a day's many software access web services for updating the products. Therefore not only user access the information, many background process may also access the same. Identifying any such activity is difficult by the normal comparison based profiles.

Hence the proposed system is developed to effectively and efficiently detecting the Intrusion.

## IV. Methodology

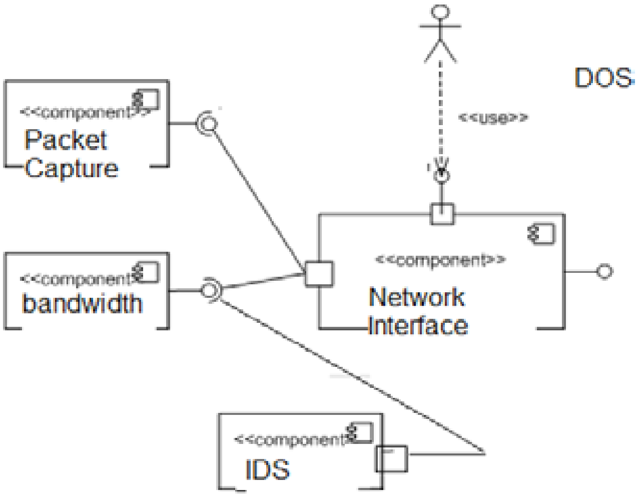

Fig. 5: Component Diagram of the system.

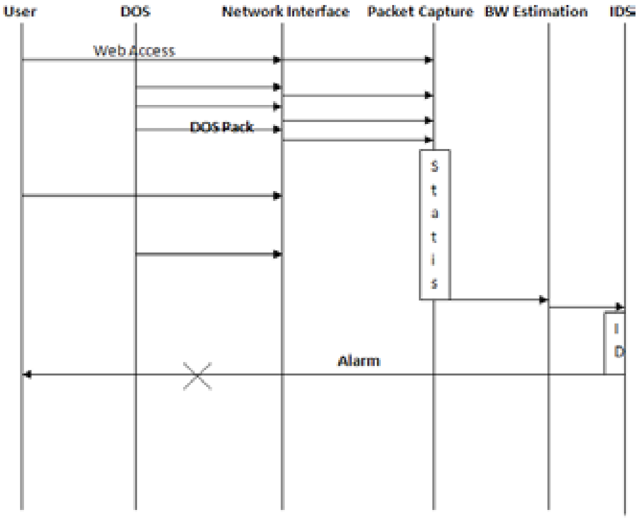

Fig. 6: Sequence diagram of the overall process.

Fig. 5 and 6 explains the entire methodology in detail. Refer fig. 6 for understanding the actual procedure for intrusion detection. It is clear that the packets are exchanged from the same interface by the valid applications or user programs and also exchange packets through the same interface that the intruding nodes access the packets from. These packets are periodicallychecked by a packet capture interface designed as the sensor of the system. The period of monitoring the interface varies unequally based on poisson's distribution so that the overhead due to packet hearing by the sensor do not affect the performance of the system. The statistical output from this interface is given as input to the intrusion detection system where depending upon the access, anomly bandwidth is estimated. If it is higher than the threshold, which is 10% of the actual bandwidth utilized by valid applications, then intrusion detection alarm is generated.

## V. Results

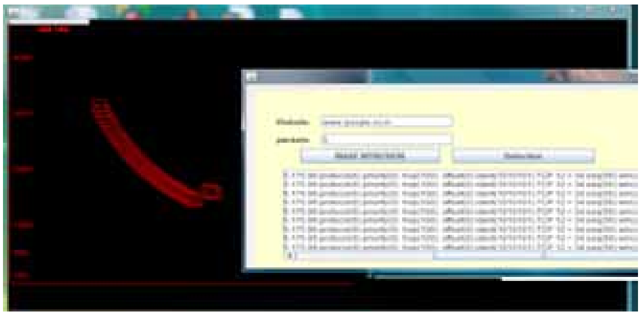

Fig. 7: A typical screenshot of the working of the proposed system.

Fig. 7 Demonstrate the variation in bandwidth of user program under no attack. The variation in bandwidth is visible which is due to various web application access.

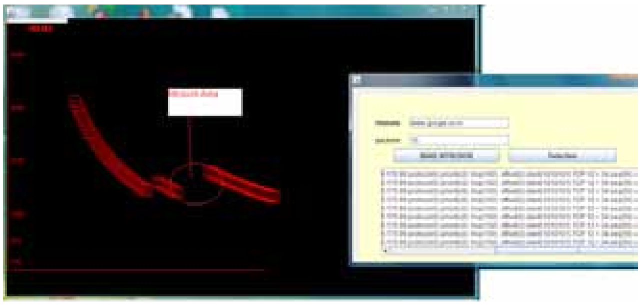

Fig. 8: Response of the system under the presence of Intrusion.

Fig. 8 demonstrates the intrusive process. When the intrusive nodes carry attack on a port and generate traffic from the same port, the application bandwidth is denied for some period of time. Generally such bursts are very short so that the IDS system do not detect such abnormalities. But results shows that even short burst intrusion is also correctly detected by the proposed system.

Comparative Analysis

Conventionally the intrusion detection system woks hand to hand with the firewall. The role of the Intrusion detection system is to detect the patterns which firewalls set the rule for. For example a particular IP address X.Y.Z.K is attacking a specific Network. Even if the intrusion detection system detects the IP address of the same and the firewall prevents the same IP, the attacker may opt for a proxy address with a variation in the IP. Therefore Intrusion detection system must not only detects the common rule but also mis-behevior. For example a web site gets 1000 clicks every day on average. If suddenly it starts receiving 10000 hits a day, than that is called a mis-behevior. A firewall can not take the site offline just because, there are abnormal activity, it needs to have the problem zone traced and protected. Therefore often intrusion detection systems are identified as Pattern Matching Algorithms where it matches the network activity with the preset rules. The Rules are also evolved with time alongside every detection. But in a real time network number of rules are quite complex and hence dedicated server resources are allocated to IDS for the same. When the same problem is traced back to the individual computers or personal networks, attacks are less complicated. Mainly the intruder utilizes the system resources. Hierarchial Pattern Matching based systems here takes a lot of bandwidth and processing power itself. Therefore mitigating to simple rule oriented yet effective IDS for detection of DOS attack is important. We summarize our rule set as follows for Various Internet access activities for a broadband connection with maximum speed 1Mbps.

ISSN : 2229-4333(Print) | ISSN : 0976-8491(Online)

Table 1: Rule Set

| Activity | Rule Name | Average Value | Minimum | Maximum | Variance |
|---|---|---|---|---|---|
| Mail Access | Bandwidth | 200kbps | 130kbps | 360kbps | 20kbps |
| Mail Access With Attachment | Attachment size (5 Mb) | Upload: 600kbps<br>Download: 780kbps | Upload: 200kbps<br>Download: 221kbps | Upload: 900kbps<br>Download: 1Mbps | 61kbps |
| Ideal Internet (Software Updates) | Bandwidth | 160kbps | 110kbps | 266kbps | 12kbps |
| Average Web Page Access ( based on YSlow Test) | Delay in web page Opening | 1.4 seconds | 3 milli seconds | 4 Seconds | 750 ms |
| Bit Torrent Client | Seeding Bandwidth | 250kbps | 200kbps | 1mbps | 100kbps |

The rule set above gives the internet access pattern of a personal computer. The system should detect the Mis behavior or intrusion based on the Rule Matching. A deviation of more than 33% of the rule set is considered to be a case of "possible Intrusion." More than 3 such cases will make the case as high probable intrusion and matching of all the criteria makes it as a sure intrusion case.

Experiments were conducted based on Following test cases.

(i) Enabling Agent based chat for the system ( with Zopim Chat)
(ii) Generating Random Traffic for a particular web sites.
(iii) Scheduling Windows and Antivirus updates to specific intervals
(iv) Activating a download with the Bit Torrent Client.

Out of the cases, the second case generates intrusion of DOS attack. We further compared the performance of our algorithm with Hierarchical pattern matching.

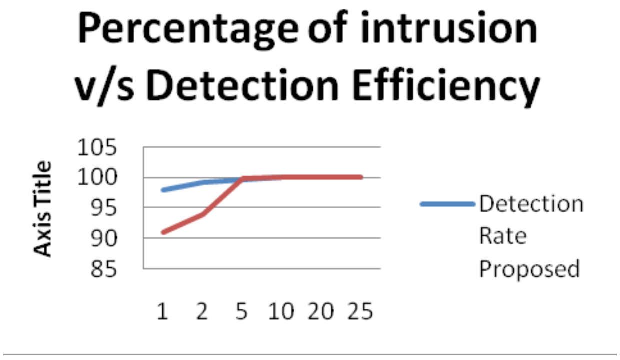

Fig. 9: % of Intrusion v/s Detection Efficiency.

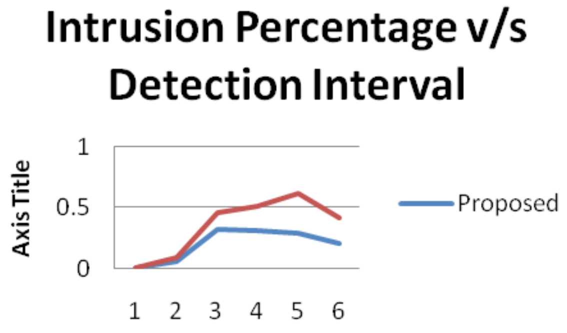

Fig. 10: Intrusion % v/s Detection Interval.

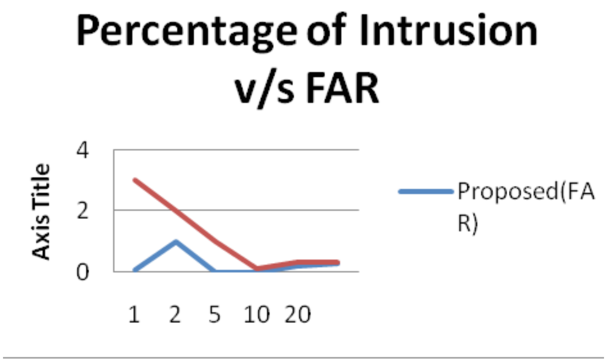

Fig. 11: % of Intrusion v/s FAR.

In above figs. 9-11, percentage of intrusion suggests number of Intruding packets in correspondence to number of overall internet packet. FAR is calculated as number of intrusive packets accepted by the system as valid packets. Detection Time is defined as the interval required by the system for the detection of the first packet. It is clear that short burst of intrusion is difficult to detect. Hierarchial patter matching needs to analyze the packets for significant iterations before detecting the intrusion. But rule based direct thresholding technique detects the intrusion fast and accurately for a smaller network or personal computer. This is due to less number of iteration required by the system to detect the intrusion.

## VI. Conclusion

There are several intrusion detection systems available for simulation and some of the tools are available at the real time. The systems are used in wide range by various applications. Most of such systems require external hardware called intrusion sensors and are not of use by common users. Therefore in this project we developed a real time intrusion detection system which detects intrusion of type denial of services which is considered to be one of the most significant type of intrusion system. The system is developed in java with collaboration of winPCap and JpCap libraries which are entirely open source. Therefore the system is an open source system. The experiments shows that the system is capable of detecting intrusion in all possible scenarios. The system can be further improved by incorporating classifier based intrusion detection mechanisms like one with hierarchical matching.

## References

[1] Rong-Tai Liu, Nen-Fu Huang, Chia-Nan Kao, Chih-Hao Chen, Chi-Chieh Chou, " A Fast Pattern-Match Engine for Network Processor-based Network Intrusion Detection System", International Conference on Information Technology: Coding and Computing (ITCC'04), IEEE, 2004.

[2] Yi Tang, Junchen Jiang , Xiaofei Wang , Bin Liu, Yang Xu, "Independent Parallel Compact Finite Automatons for Accelerating Multi-String Matching", IEEE 2010.

[3] Mohammad A. Alia, Adnan A. Hnaif, Hayam K. Al-Anie, Khulood Abu Maria, Ahmed M. Manasrah, M. Imran Sarwar, "A NOVEL HEADER MATCHING ALGORITHM FOR INTRUSION DETECTION SYSTEMS", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.4, pp. 59-73, July 2011.

[4] Lambert Schaelicke, Thomas Slabach, Branden Moore, Curt Freeland, "Characterizing the Performance of Network Intrusion Detection Sensors", pp. 155–172, Springer-Verlag

Berlin Heidelberg 2003.

[5] Miyuki Hanaoka , Kenji Kono, Toshio Hirotsu, Hirotake Abe, “Performance Improvement by Coordinating Configurations of Independently-managed NIDS”, IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.5, May 2011.

[6] M. Nourani, P. Katta, “Bloom Filter Accelerator for String Matching”, IEEE, 2007.

[7] C. Jason Coit, Stuart Staniford, Joseph Mcalerney, “Towards Faster String Matching for Intrusion Detection or Exceeding the Speed of Snort”, IEEE, 2001.

[8] Hongbin Lu, Kai Zheng, Bin Liu, Xin Zhang, Yunhao Liu, “A Memory-Efficient Parallel String Matching Architecture for High-Speed Intrusion Detection”, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 24, NO. 10, OCTOBER 2006.

[9] Amitava Biswas, Purnendu Sinha, “On improving performance of Network Intrusion Detection Systems by efficient packet capturing”, IEEE, 2006.

[10] D.Jeyabharathi, D.Sasireka, D.Kesavaraja, “Implementation of Mobile Intrusion Detection Controller [MIDC] for Affording Secure Service in MANET Environment”, IJCA Special Issue on “Mobile Ad-hoc Networks” MANETs, 2010.

Revathi Cherukuri is currently pursuing B-Tech Final year in Computer Science Engineering at Jawaharlal Nehru Institute of Technology, Andhra Pradesh, India. Her areas of interests are Network Security, Data Mining.

Thadoor Shobha Rani is currently pursuing B-Tech Final year in Computer Science Engineering at Jawaharlal Nehru Institute of Technology, Andhra Pradesh, India. Her areas of interest include Software Engineering, Network Security.

Challa Madhavi is currently pursuing B-Tech Final year in Computer Science Engineering at Jawtaharlal Nehru Institute of Technology, Andhra Pradesh, India. Her areas of interest include Software Engineering, Network Security.

Dr.Manjunath Gadiparthi received his B.Tech Degree in Computer Science & Engineering, and M.Tech Degree in Computer Science & Engineering. He Completed his Ph.D. in Computer Science & Engineering.He contributed several research articles in International journals &International Conferences. He is currently working as an Associate Professor and Head in the Department of Computer Science & Information Technology in Jawaharlal Nehru Institute of Technology (JNIT). His areas of interest are Image Processing, Data Mining & Network security.