International Journal of Information & Network Security (IJINS)
Vol.2, No.2, April 2013, pp. 160~166
ISSN: 2089-3299

Defending IP Spoofing Attack and TCP SYN Flooding Attack in Next Generation Multi-hop Wireless Networks

I. Diana Jeba Jingle*, Elijah Blessing Rajsingh**
*Departement of Computer Science and Engineering, LITES
**Karunya School of Computer Science and Technology, Karunya University

Article Info

Article history:

Received Nov 10th, 2012
Revised Dec 15th, 2012
Accepted Dec 21th, 2012

Keyword:

Multi-hop
Flooding
Monitor
DoS
Spoofing

ABSTRACT

Multi-hop wireless networks are normally affected by TCP SYN flooding and IP address spoofing attacks. TCP SYN flooding occurs while establishing a TCP connection for data transmission. But, even after a TCP connection is established, TCP protocol is flooded by a novel connection flooding attack which aims at consuming the entire bandwidth allocated to a network. Although effective techniques exist to combat SYN flooding, no single standard remedy for defending this novel flooding attack on TCP has emerged. In this paper, we describe the attack and provide an overview and assessment of existing defense mechanisms and we propose a novel defense mechanism which not only involve in defending such flooding attack but also prevents IP spoofing which is the gateway for such flooding attacks. The performance analysis is carried out and we prove the effectiveness of the proposed defense mechanism in terms of time delay and false positive rates.

Corresponding Author:

I.Diana Jeba Jingle,
Departement of Computer Science and Engineering,
Loyola Institute of Technology and Science,
Loyola Nagar, Thovalai 629302, Tamil Nadu, India.
Email: dianajebajingle@gmail.com

1. INTRODUCTION

Wireless broadband Network is an emerging next generation multi-hop wireless network technology. Denial-of-service attack is a vulnerable attack at all the layers of such networks. The most common DoS attack is the TCP SYN flooding attack that occurs at the transport layer of such networks. TCP SYN flooding is a specific denial-of-service attack which exploits the normal behavior of Transmission Control Protocol (TCP) and makes the victim incapable to carry out its services. SYN flooding attacks normally occurs which establishing a three-way handshake TCP connection between a sender and a receiver. The normal three-way handshake process is shown in figure 1and an abnormal three-way handshake process is described as follows: An attacker with node A's spoofed IP address tries to connect to node B by sending a SYN segment. Node B thinks that the SYN segment belongs to node A and opens a TCP connection and replies with a SYN/ACK segment to node A. The node B sends numerous SYN/ACK segments until it receives ACK from the node A. Since node A’s IP address is spoofed, the SYN/ACK never reaches node A, but it reaches the attacker. The attacker never sends ACK to Node B. Node A also never sends ACK to Node B leaving the connection half-open and no data transfer takes place in this connection as shown in figure 2.

TCP allocates memory for each incoming SYN requests and releases the memory only after ACK is received. So the memory is exhausted and there is no space to allocate memory for legitimate SYN messages. Thus the attacker denies the victim from servicing new connection requests. This situation leads to a complete denial-of-service attack. In order to avoid this memory exhaustion, operating systems generally uses a backlog queue to clearly identify the number of half-open connections. This action protects a host's available memory resource from attack, but the backlog queue *itself* represents another resource vulnerable to attack. With no room left in the backlog, it is impossible to service new connection requests until some half-open connections are forcefully terminated.

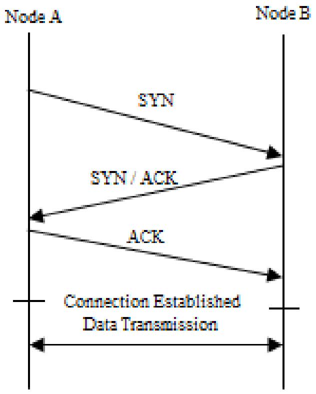

Figure 1. Normal Three-way Handshake Process

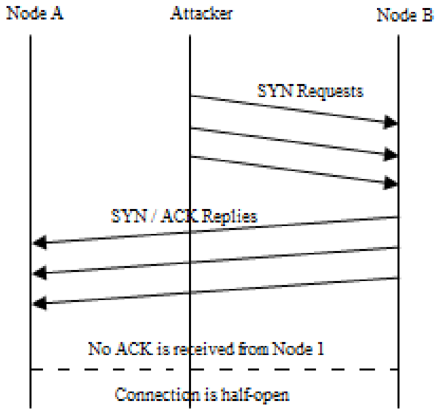

Figure 2. Three-way Handshake Process under Attack

IP spoofing is the main gateway for SYN flooding attacks which is considered as a most complex attack in which the attackers create raw IP packets with valid IP and TCP headers. An attacker might spoof a single source address or multiple source addresses. It is a difficult task for the listener to detect and filter the spoofing attacks with multiple source addresses than spoofing attacks with single source address. Spoofing attacks can be prevented by using network ingress filters [1] and egress filters in proper network locations. IP Security (IPsec) also provides an excellent defense against IP spoofing, but this protocol generally cannot be required because its deployment is currently limited. It is a quite challenging task to block spoofing attacks with multiple source addresses.

Diminishing the backlog is the goal of the TCP SYN flooding attack, which attempts to send enough SYN segments to fill the entire backlog. The Transmission Control Block (TCB) is a transport protocol data structure that holds all the information about a connection. Usually, each TCB exceeds at least 280 bytes, and in some operating systems currently takes more than 1300 bytes. The TCB is allocated based on reception of the SYN packet before the connection is fully established. SYN caches [22] operate by reducing the amount of state allocated initially for a TCB generated by a received SYN. Hosts that implements SYN cache uses a hash table with its index in a hash bucket. The hash value is computed on the incoming packet using the source and destination addresses, the source and destination port, and a randomly chosen secret. This value is then used as an index into a hash table, where SYN cache entries are kept on a linked list in each bucket. The secret is used to perturb the hash value so that an attacker cannot target a specific hash bucket and deny service to a specific machine. While on the surface it may appear that an attacker could implement a DoS by targeting a hash bucket so that a legitimate connection does not reside on the queue long enough to establish a connection, the risks are mitigated by the use of the hash secret. Additionally, since the port number of the connecting machine is used in the hash calculations, a second connection attempt from the client machine tends to result in a second hash bucket chosen, further stemming any attempt by an attacker to target a specific bucket.

In contrast to the SYN cache approach, the SYN cookies [22] technique causes absolutely zero state to be generated by a received SYN. Instead, the most basic data comprising the connection state is compressed into the bits of the sequence number used in the SYN-ACK. Since for a legitimate connection, an ACK segment will be received that echoes this sequence number, the basic TCB data can be regenerated and a full TCB can safely be instantiated by decompressing the Acknowledgement field. This decompression can be effective even under heavy attack because there is no storage load whatsoever on the listener, only a computational load to encode data into the SYN-ACK sequence numbers. To compute the SYN-ACK sequence number (that is, the TCP cookie) when using TCP cookies, a host first concatenates some local secret bits, a data structure that contains the IP addresses and TCP ports, the initial SYN sequence number, and some index data identifying the secret bits. An MD5 digest is computed over all these bytes, and some bits are truncated from the hash value to be placed in the SYN-ACK sequence number. Because the sequence number is about a fourth the size of the full hash value, this truncation is necessary, but generally at least 3 bytes worth of the hash bits are used, meaning that there should still be close to a 2^24 effort required to guess a valid cookie without knowing the local secret bits. In addition to the hash output, some of the cookie bits indicate a lower bound on the Maximum Segment Size (MSS) that the SYN contained, and the index bits identifying the local secret used within the hash. To validate a SYN cookie, first the acknowledgement number in an incoming ACK segment is decremented by 1 to retrieve the generated SYN cookie. The valid value for the set of truncated hash bits is computed based on the IP address pair, TCP port numbers, segment sequence number minus one, and the value from the secret pool corresponding to the index bits inside the cookie. If these computed hash bits match those within the ACK segment, then a TCB is initialized and the connection proceeds. The encoded MSS bound is used to set a reasonable-sized MSS that is no larger than what was originally advertised. This MSS is usually implemented as three bits whose code points correspond to eight "commonly advertised" MSS values based on typical link Maximum Transmission Units (MTUs) and header overheads.

SYN flooding attacks can be defended against by altering only the initial handshaking procedure, whereas IP spoofing attacks require additional per-segment checks throughout the lifetime of a connection. Some approaches implement both SYN cache and SYN cookies as a defense against SYN flooding attacks. Both SYN cache and SYN cookies detect and filter SYN flooding attacks while a connection is being established. But they were unable to prevent SYN flooding at the initial stage. SYN flooding denies service to new connections, without affecting in-progress connections, whereas IP spoofing attacks disrupt in-progress connections, but do not prevent new connections from starting. The main root cause for SYN flooding attack is IP spoofing where an attacker tries to spoof a node’s IP address when it joins a network. Hence by preventing IP spoofing, SYN flooding attacks can be prevented to a greater extend. Defense against these attacks would be a universal deployment of address filtering or IPsec.

Wei Chen et al [21] proposed a storage-efficient data structure and a change-point detection mechanism to distinguish normal three-way TCP handshakes from abnormal ones. This mechanism leads to large memory comsumption. Sungwon Yi et al [17] introduced a two-level cache Content Addressable Memory to dynamically detect and quarantine the unresponsive TCP flows. But it leads to large memory consumption. Dimitris Geneiatakis et al [4] proposes a new header to overcome signaling DoS attacks in SIP servers. But the proposed scheme uses a pre-shared key which when explored leads to password-based attacks and also it is vulnerable to man-in-the-middle attacks. Supranamaya Ranjan et al. [20] proposed DDoS-Shield to detect the attack packets that overwhelm the system resources such as bandwidth. DDoS Shield consist of a suspicion assignment mechanism that examines requests belonging to every TCP, UDP and ICMP session for assigning suspicion values to each session and a DDoS-resilient scheduler that schedules the sessions based on the values assigned to the sessions and decides which session to be forwarded and when and performs rate-limiting for abnormal sessions. DDoS shield improves the victim’s concert by consuming less memory for buffering requests and responses. However DDoS Shield consumes more processing time thereby unable to produce good throughput. TVA [23] uses capabilities to discard unauthorized traffic floods on a single autonomous system. TVA achieves high throughput and the problem is TVA stores all capability information of each user on routers and a router with limited number of queues may not be able to protect all legitimate users. DWARD [12] autonomously detects and filters attack traffic from legitimate traffic by dropping the excess traffic by limiting the traffic rate to and from the victim thereby reducing the overload at the victim. But DWARD cannot detect attack traffic until connection buffer fills up thereby causing increased time delay to detect an attack and it causes more communication overhead. DARB [] uses an active probing detection method and a TTL based rate-limit counteraction method to detect and filter SYN flooding attack traffic accurately and independently on the victim side. DARB consumes more amount of the victim’s bandwidth and causes computation overhead for both detecting and neutralizing methods. Ge Zhang et al [7] proposes a priority mechanism for blocking attacks on SIP proxies caused by external processing. But this mechanism causes time delay and decreased throughput when SIP proxies interact with external servers. Haidar Safa et al. [8] proposed CDMS that is implemented at the edge routers of spoofed IP address’ networks to defend the victim. CDMS also a communication protocol is used to encourage collaboration between various networks. This mechanism is very efficient and it prevents the routers from being overloaded. However this mechanism causes time delay to detect and filter an attack. Sudip Misra *et al.* [18] proposed DLSR which uses the concept of Learning Automata and prevents the server being overloaded with excess amount of illegitimate traffic from crashing and keeps the server functioning. However DLSR cannot effectively differentiate valid user’s IP address and spoofed user’s IP address and it also causes excess time delay to detect and filter an attack. Patrick P.C. *et al.* [16] proposes an online early detection algorithm based on the statistical CUSUM method for detecting signalling DoS attacks on wireless networks in a timely manner. This approach does not detect the attack traffic that has a spoofed IP address and causes signaling load on the control plane. This detection mechanism blocks both benign and malicious traffic when the signaling load reaches a threshold. Joseph Chee Ming Teo et al [13] proposes a group key agreement protocol to protect heterogeneous networks against DoS attacks. But it causes more communication overhead in heterogeneous networks. Dimitris Geneiata et al. [5] proposed a two-part bloom filter based monitor to detect and filter flooding attacks against proxy servers. The monitor's main task is to record the state of any incoming session in 3 different filters and the filter is indexed through a hash function. This mechanism uses an alarming system to trigger an alarm and report if any entries in the filter exceed the threshold value. This mechanism is very efficient and cost-effective and causes reduced time delay to detect an attack. However, hashing of entries in the filters leads to computation overhead and more CPU utilization.

The existing approaches in the area of DoS were unable to effectively differentiate valid user's IP address and spoofed user's IP address. Once the attack is identified all traffic is blocked to reduce the load on the server thereby blocking legitimate traffic also. The buffer capacity in the router's queue is not enough to accumulate all legitimate traffic requests. Also to detect an attack, the system consumes more time. Due to these limitations in the existing approaches, it is necessary to frame a novel approach to mitigate DoS attack in wireless broadband networks. It is observed that DOS attacks depend heavily on IP spoofing; therefore preventing IP spoofing might contribute to solving the problem. A common way for preventing IP spoofing is by using ingress and egress filters on firewalls. But it fails in wireless networks where legitimate packets could have topologically incorrect addresses.

2. PROPOSED METHOD

We propose a universal network address filtering deployment technique to prevent IP spoofing and SYN flooding attacks at the initial stage. The network consists of monitoring agents located at various points. A single global monitoring agent is located near the gateway router and a cluster of local monitoring agents are located near the local routers of each subnets. The global monitoring agent maintains a table containing host name, host address and secret value. The secret value is a hash of IP address and timer value. The timer value indicates the time when a node is present. The local monitoring agent maintains a table containing IP address and clock values. The clock values indicate sequence of timer values. After a node joins a network, it should periodically send its timer values to the local monitoring agent which computes the initial threshold. The proposed defense strategy is shown in figure 3 and is described as follows: The secure network adds nodes to be a part of it only if it passes the challenge test. The node first sends a join request message to the global monitoring agent in the network. The global monitor requests the node to send a challenge message. The node in turn responds with a HMAC value of the challenge response message. The challenge response message consists of the node's IP address, MAC address and the current registration time. Now the global monitor responds with a hash of HMAC value of a join success message to the node. The success message consists of N, the number of bytes the node is allowed to transmit and receive and TTL, the time limit within which N can be used up. Now the global monitor informs the local monitor to grant permission for that node to transmit and receive data. The global monitor gives half of its control to the local monitor. The nodes send clock values to the local monitor at periodic intervals and the local monitor computes the clock values to set the initial threshold as T_0-T_1 = δ. This value is set as the initial threshold and clock values failing to meet δ are predicted as spoofed nodes. Thus at this stage IP spoofing attack is banned. In order to defend TCP SYN flooding attack, the local monitor checks for each incoming SYN request, whether T_{i-1}-T_i = δ. If this condition is satisfied, then local monitor predicts that the SYN request is valid and checks whether the probability of SYN/ACK$_{out}$ = =1. If the condition is satisfied, then local monitor predicts that the traffic is normal and stores SYN and SYN/ACK in the buffer until the arrival of ACK. If both conditions are not met, the local monitor predicts that the SYN is invalid and the traffic is abnormal and blocks traffic to and from that node and informs the neighboring local monitors about the abnormality. The neighboring local monitors are now aware of this abnormality and block traffic from that particular node. The local monitor and its neighboring local monitors altogether inform the global monitor about this abnormality and global monitor removes the node from its list of valid IP address. Thus SYN flooding attack is banned through IP spoofing defense. In order to soothe the burden on a single global monitor, we propose group of local monitors at various points in the network. All nodes within a network have synchronized clock values. There is a chance for legitimate nodes being falsely detected as attack nodes by the global monitor. In such case, the global monitor selects the rejected nodes from its table and asks those nodes to re-register with a new set of challenge message. If the node is still under attacker's hold, then the attacker cannot send a successful reply message.

The SYN cache approach causes increased time delay to detect an attack. SYN cookies never maintain a state to store incoming SYN or SYN/ACK messages and retransmission of such messages requires a state which is achieved in our proposed method. Both SYN caches and SYN cookies causes more computation overhead to detect SYN flooding attacks. Our proposed method causes reasonable computation overhead to detect and filter both TCP SYN flooding and IP spoofing attacks.

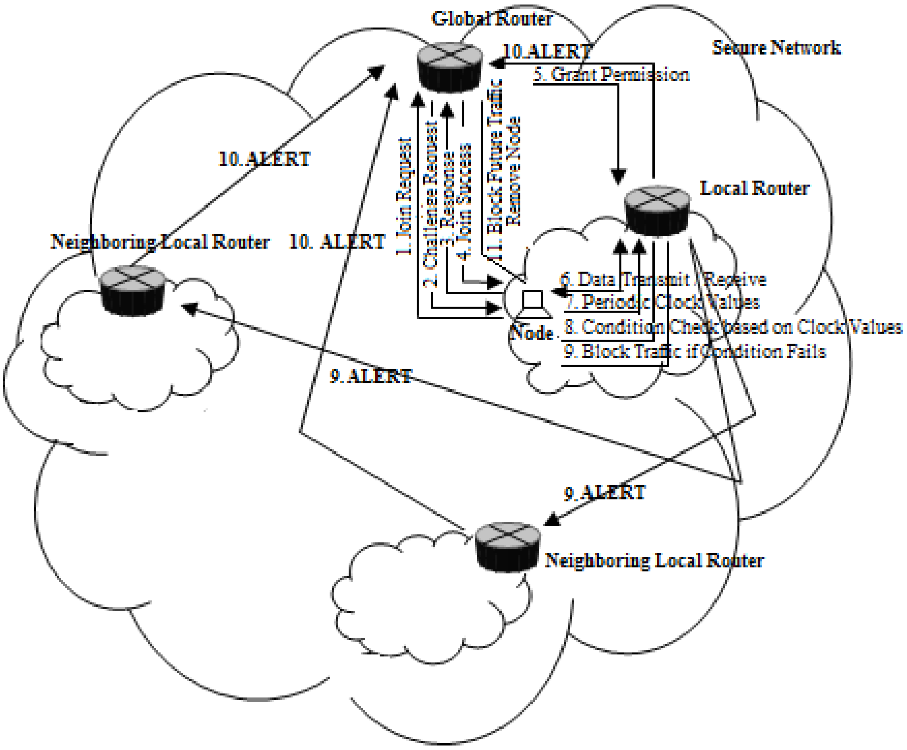

Figure 3. Proposed Defense Architecture

3. RESULTS AND ANALYSIS

The simulations are carried out using NS-2 network simulator. The network nodes are arranged in a mesh topology, the global monitors transmit and receive packets to and from other networks and its own network and the local monitors sends and receives packets to and from the source nodes in the network. The global monitors are installed near global routers and local monitors are installed near local monitors. The network contains 20% local monitors and one global monitor. All registered nodes within the network are set to be synchronized. Detection Time, Computation Overhead and False Positive Rate are the performance metrics used to measure the effectiveness of the proposed defense mechanism. Time delay is the time taken for the local monitor to detect a node under IP spoofing and SYN flooding attack. The detection time taken by the proposed method is less when compared to SYN cache and SYN cookie. Computation overhead is the number of computations required by the proposed scheme to execute the defense. The number of computations used by the proposed method is only two and it does not affect the performance of the network. False positive rate is the ratio of the number of legitimate node wrongly detected as attack nodes. The proposed defense method experiences only a less number of false positive ratio when compared to the existing defense mechanisms. The performance results are clearly stated in figures 4, 5 and 6.

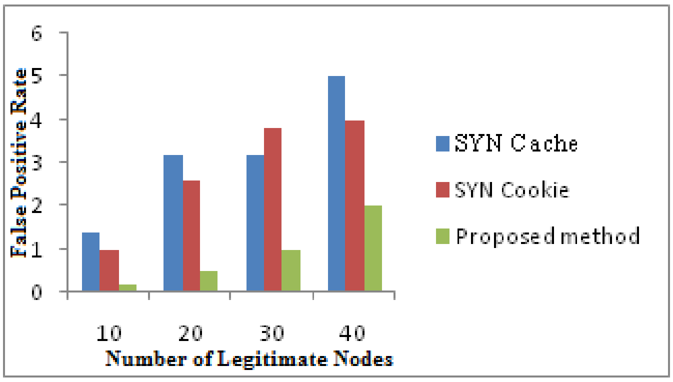

Figure 4. Number of Legitimate Nodes vs. False Positive Rate

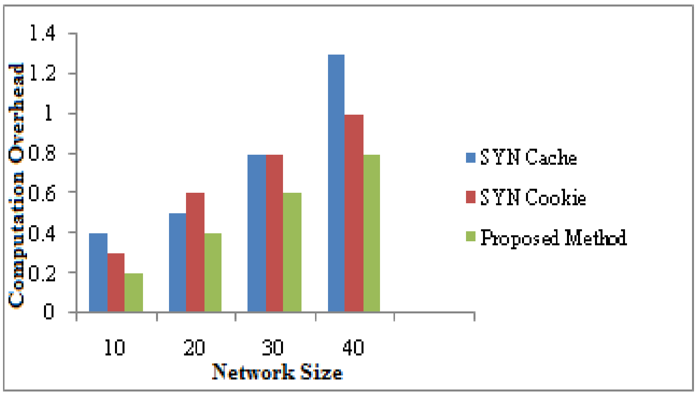

Figure 5. Network size vs. computation overhead

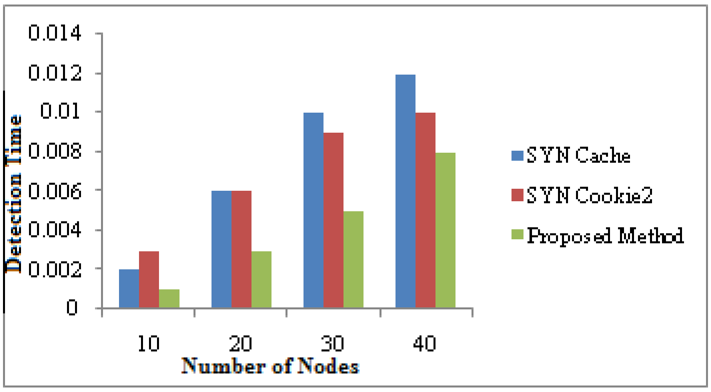

Figure 6. Number of nodes vs. detection time

4. CONCLUSION

In this paper, we have clearly described about IP address spoofing attack and TCP SYN flooding attack and provided a clear analysis about the existing approaches such as SYN cookie and SYN cache for defending SYN flooding attacks. We have proposed a novel approach which defends both flooding attack and IP spoofing attacks in multi-hop wireless networks. Our proposed method works considers the clock values of each node into account for effective mitigation of such attacks. Our proposed method is compared with existing defense methods such as SYN cache and SYN cookie and is proved to be effective over existing methods in terms of detection time delay, false positive rates and computation overhead.

REFERENCES

[1] Amey Shevtekar, Karunakar Anantharam, And Nirwan Ansari, (2005). “Low Rate TCP Denial-Of-Service Attack Detection At Edge Routers,” IEEE Communications Letters, Vol. 9, No. 4

[2] Bin Xiaoa, Wei Chenb, Yanxiang Hec, (2008) “An Autonomous Defense Against SYN Flooding Attacks: Detect And Throttle Attacks At The Victim Side Independently,” Elsevier Journal Of Parallel And Distributed Computing, Volume 68, Pages 456 – 470.

[3] Cert Advisory Ca-1996-21 (1996). “TCP SYN Flooding And IP Spoofing Attacks”, CERT CC, Http://Www.Cert.Org/Advisories/Ca-1996-21.

[4] Dimitris Geneiatakis, Costas Lambrinoudakis, (2007). “A lightweight protection mechanism against signaling attacks in a SIP-based VoIP environment”, Journal of Telecommunication Systems (Springer), Volume 36, Issue 4, pp 153-159.

[5] Dimitris Geneiatakis, Nikos Vrakas, Costas Lambrinoudakis, (2009) “Utilizing Bloom Filters For Detecting Flooding Attacks Against SIP Based Services,” Elsevier Journal Of Computers & Security, Volume 28, Issue 7, Pages 578-591.

[6] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," BCP 38, RFC 2827, May 2000.

[7] Ge Zhang, Simone Fischer-Hübner, Sven Ehlert , (2010). “Blocking attacks on SIP VoIP proxies caused by external processing”, Journal of Telecommunication Systems (Springer), Volume 45, Issue 1, pp 61-76.

[8] Haidar Safa, Mohamad Chouman, Hassan Artail, Marcel Karam, (2008) “A Collaborative Defense Mechanism Against SYN Flooding Attacks In IP Networks,” Elsevier Journal Of Network And Computer Applications, Volume 31 Issue 4.

[9] I.B. Mopari, S.G. Pukale, M.L. Dhore, (2009). "Detection Of DDoS Attack And Defense Against IP Spoofing", In: Proceedings Of The International Conference On Advances In Computing, Communication And Control, ICAC3'09, Mumbai, Maharashtra, India, PP. 489-493.

[10] Issa Khalil, Saurabh Bagchi, Ness B. Shroff, (2008). "Mobiworp: Mitigation Of The Wormhole Attack In Mobile Multihop Wireless Networks," Elsevier Journal On Ad Hoc Networks (6), Pages 344–362.

[11] J.Ioannidis And S. Bellovin, (2002). "Implementing Pushback: Router-Based Defense Against Dos Attacks," In Proc. NDSS.

[12] Jelena Mirkovic, Peter Reiher, (2005) "D-WARD: A Source-End Defense Against Flooding Denial-Of-Service Attacks," IEEE Transactions On Dependable And Secure Computing, Vol. 2, No. 3.

[13] Joseph Chee Ming Teo, Chik How Tan, Jim Mee Ng, (2007). "Denial-of-service attack resilience dynamic group key agreement for heterogeneous networks", Journal of Telecommunication Systems, Volume 35, Issue 3-4, pp 141-160.

[14] Nikhil Saxena, Mieso Denko, Dilip Banerji, (2010). "A Hierarchical Architecture For Detecting Selfish Behaviour In Community Wireless Mesh Networks," Elsevier Journal Of Computer Communications.

[15] P.Ferguson And D.Senie, (2000). "Network Ingress Filtering: Defeating Denial Of Service Attacks That Employ IP Source Address Spoofing," Internet RFC 2827.

[16] Patrick P.C. Lee A, Tian Bu B, Thomas Woob, (2009) "On The Detection Of Signaling Dos Attacks On 3G/Wimax Wireless Networks," Elsevier Journal On Computer Networks (53).

[17] Sungwon Yi, Xidong Deng, George Kesidis, Chita R. Das, (2008). "A dynamic quarantine scheme for controlling unresponsive TCP sessions", Volume 37, Issue 4, pp 169-189.

[18] Sudip Misra, P. Venkata Krishna, Kiran Isaac Abraham, Navin Sasikumar, S. Fredun, (2010) "An Adaptive Learning Routing Protocol For The Prevention Of Distributed Denial Of Service Attacks In Wireless Mesh Networks," ACM Journal Of Computers & Mathematics With Applications, Vol. 60, Issue 2.

[19] Suman Jana And Sneha K. Kasera, (2010). "On Fast And Accurate Detection Of Unauthorized Wireless Access Points Using Clock Skews," IEEE Transactions On Mobile Computing, Vol. 9, No. 3.

[20] Supranamaya Ranjan, Ram Swaminathan, Mustafa Uysal, Antonio Nucci, And Edward Knightly, (2009). "DDoS-Shield: DDoS-Resilient Scheduling To Counter Application Layer Attacks," IEEE/ACM Transactions On Networking, Vol. 17, No. 1.

[21] Wei Chen, Dit-Yan Yeung, (2006). "Throttling spoofed SYN flooding traffic at the source", Journal of Telecommunication Systems, Volume 33, Issue 1-3, pp 47-65.

[22] *Wesley M. Eddy,* "Defenses Against TCP SYN Flooding Attacks", The Internet Protocol Journal, Volume 9, Number 4, December 2006.

[23] Xiaowei Yang, Wetherall, D. Anderson, T., (2008) "TVA: A Traffic Validation Approach", IEEE/ACM Transactions On Networking, Vol. 16 , Issue 6.

BIOGRAPHY OF AUTHORS

I. Diana Jeba Jingle is the Assistant Professor of Loyola Institute of Technology and Sciences, India. She reveived her Bachelor of Engineering degree in Information Technology from Sun College of Engineering and Technology, Anna University, India in 2006. She received her Master of Engineering degree in Computer Science from Francis Xavier Engineering College, Anna University, India in 2008. Her research interests lie in the area of Wireless Networks, Mobile Ad-hoc Networks and network security and specifically focus on denial-of-service characterization, detection and defense, IP spoofing defense. She is a member of CSI.

Elijah Blessing Rajsingh is the Professor and Director for the Department of Computer Science and Engineering of Karunya University, India. He received his Master of Engineering degree with Distinction from the College of Engineering, Anna University, India. He received the Ph. D degree in Information and Communication Engineering from College of Engineering, Anna University, India in 2005, focusing on Security in Wired and Wireless Networks. He is the member of IEEE. He has very strong research background in the areas of Network Security, Mobile Computing, Wireless & Ad hoc Networks, Parallel and Distributed Computing. He is an Associate Editor for International Journal of Computers & Applications, Acta Press, Canada and member of the editorial review board for International Journal of Cases in E Commerce as well as for Information Resources Management Journal, Idea Group Publishers, USA. He is the recognized guide for Ph.D students of Karunya University and is guiding students in their doctoral programme. He has published a number of papers in international journals and conferences.