New Mitigating Technique to Overcome DDOS Attack

V. Praveena, and N. Kiruthika

***Abstract*—**In this paper, we explore a new scheme for filtering spoofed packets (DDOS attack) which is a combination of path fingerprint and client puzzle concepts. In this each IP packet has a unique fingerprint is embedded that represents, the route a packet has traversed. The server maintains a mapping table which contains the client IP address and its corresponding fingerprint. In ingress router, client puzzle is placed. For each request, the puzzle issuer provides a puzzle which the source has to solve. Our design has the following advantages over prior approaches, 1) Reduce the network traffic, as we place a client puzzle at the ingress router. 2) Mapping table at the server is lightweight and moderate.

***Keywords*—**Client puzzle, DDOS attack, Egress, Ingress, IP Spoofing, Spoofed Packet.

I. Introduction

DISTRIBUTED denial of service attack pose a major threat to the availability of internet services. CERT defined the term DOS as follows [1],

- Occupancy of limited resources of difficult to renew such as network bandwidth, data structure or memory of a system.
- Changeable or damage network data, for instance delete system configuration, shutdown web services.
- Changeable or damage physical information.

DDOS attack can be organized from the following factors.

- Lack of security in the whole internet
- Launching attack tools has more capability to launch sophisticated attack.
- Network bandwidth or resource attack can inevitably be avoided.
- Any host on the internet can be a victim of attack.

DDOS means there are more than one object which is DOS attacker (either automated tools or human). A DDOS attacker can greatly reduce the quality of a target internet service or even can completely break the network connectivity of a server generally to achieve resource overloading; a DDOS attacker will first compromise a large number of hosts and subsequently instruct this compromised host to attack the service by exhausting a target resource.

V. Praveena is with Dr. N.G.P. Institute of Technology, Coimbatore, Tamilnadu, India (phone: 919894512112; e-mail: praveenavenkatesan@gmail.com).

N. Kirithuka is with Dr. N.G.P. Institute of Technology, Coimbatore, Tamilnadu, India (phone: 919894742361; e-mail: kiruthika.me@gmail.com).

Due to lack of built in security mechanism in the current internet infrastructure an attacker can easily get access to a large number of insecure computers with exploit/attack programs such as trinoo, TFN etc.

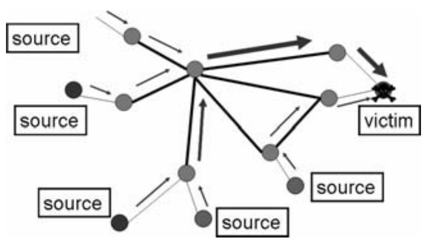

Fig. 1 DDOS Attack

In Feb. 2000, a string of DDOS attacks crippled popular wed sites including CNN.com, yahoo.com, eBay.com for several hours. In 2003, for example, one honey pot research project saw 15,164 unique zombies from a large botnet within days. In 2004, the witty worm created 12,000 zombies within 45min. IP spoofing has often been exploited by DDOS attack to 1) conceal flooding sources and dilute localities in flooding traffic 2) coax legitimate host into becoming reflectors redirecting and amplifying flooding traffic.

IP spoofing is commonly associated with malicious network activities, such as DDOS attacks, which block legitimate access by either exhausting victims severs resources [2] or saturating stud networks access links to the internet [3]. On the other hand, defending against DDOS attack is extremely difficult because there is usually no explicit attack pattern to distinguish legitimate packets from malicious ones. Moreover to hide the source of attack programs generally fill IP header fields, especially the 32-bit source IP address, with randomized values. This IP spoofing technique has made the detection and filtering of DDOS traffic extremely difficult and it has become a common feature of the many DDOS attack tools.

To design an effective and feasible DDOS countermeasure, there are several requirements a DDOS defense mechanism should meet [4].

II. Related Work

Currently there are several mechanisms to counter DOS and DDOS attack. These schemes can be roughly categorized into four classes: attacker-end based, network-based, victim-end based, and hybrid. The attacker-end based approaches [8, 9] attempt to identify DDOS attack traffic or spoofed IP packets at attack sources. Once DDOS attack traffic or spoofed packets are detected, proactive filtering mechanisms are activated to stop attack traffic from entering the Internet. The network-based approaches count on Internet routers to defend against DDOS attacks in a cooperative manner. Schemes in this category perform either the trace back of the attack traffic or complex filtering operations on routers. IP traceback schemes [10, 11,12,13,14, 15, 16, 17, 18] focus on identifying the origins of spoofed DDOS attacks, rather than stopping these attacks. The victim-end approaches [19, 20, 21] try to enhance the resilience of Internet servers against DDOS attacks. The advantages of the victim-end approaches are that they do not require support from the Internet routing infrastructure and that they strongly motivate the victim to deploy these schemes owing to the direct benefit to the victim itself.

Schemes in the fourth category can be considered a hybrid of network-based and victim-based approaches. These schemes require support from the Internet routing infrastructure and from the victim or victim network. In these schemes, routers mark each incoming IP packet in a deterministic or probabilistic manner. Then, in victims or victim networks, attack packets are identified and discarded on a per packet basis according to marks left by internet routers [22, 23, 24]. An IP traceback method is employed to construct the attack graph, and subsequently IP packets marked with one of network edges in the attack graph are discarded [22,23]. In another scheme [24], each participating router marks some bits (one or two bits) in the Identification field of an IP packet according to the router's IP address and the TTL value in the IP header. In this way, an IP packet will arrive at its destination along with a unique identifier representing the path it has traversed.

The Internet currently carries an enormous amount of undesirable network communication. This is evidenced by the growing infestation of worms and viruses such as Nimda, Code Red, and SQL Slammer [25, 26, 27], reconnaissance attacks such as port scans, targeted distributed denial-of-service attacks, and spam. Client puzzles [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39] have been proposed as a mechanism for controlling such communication. Being protected generates a cryptographic puzzle that a client must answer correctly before it is given service. Such a mechanism gives devices the ability to selectively push back load to the source of an attack when overloaded. While the standard defense for preventing undesirable communication is to apply a binary filter to traffic, such a defense is difficult to use due to the impact of false positives and the inability to completely differentiate good traffic from bad. Client puzzles provide a complementary weapon to filtering in that they provide an analog control against traffic that may potentially be deleterious. In contrast to filtering, client puzzles also limit an attacker's ability to send bad traffic to multiple victims concurrently by consuming their computational resources.

In our approach we are combining the attacker-end based and victim-end based approach together, in which the DDOS attack is almost completely removed. In the attackers end hint based hash reversal puzzle is used and in the victim end a path fingerprint approach is used.

III. Client Puzzle

Client puzzle is a technique that strives to improve the DDOS attack: the client is required to commit computing resources before receiving resources.

A. Puzzle Protocol

The client attaches a cookie consisting of its nonce and a timestamp. A server requiring puzzles generates a puzzle and answers along with a hash of the answer, server nonce, puzzle expiration time, puzzle maturity time and flow identifier. The server then sends back to client, the client cookie, puzzles and its parameters, flow identifier and a server cookie consisting of the above hash, server timestamp, puzzle maturity and expiration time. The client upon receiving the puzzle calculates the solutions and sends back the answer along with the server cookie. Upon receipt of this message, the servers uses the server timestamp to index into the server nonce table to obtain the server nonce, checks that the nonce has not expired and verifies the answer by regenerating the hash and comparing it against what the client sent. If it does, the correct answer has been given and the server accepts the packet.

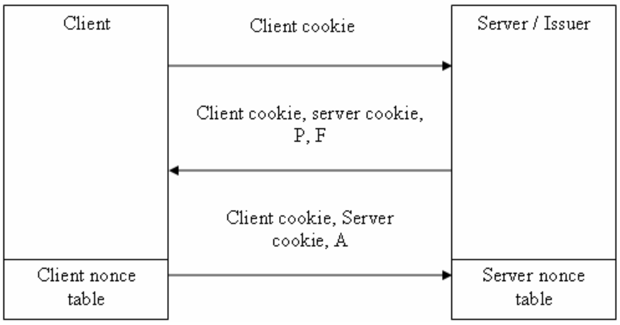

Fig. 2 Client Puzzle Protocol

A good puzzle should have the following properties,

- Creating a puzzle and verifying the solution is inexpensive for the server.
- The cost of solving the puzzle is easy to adjust from zero to impossible.
- The puzzle can be solved on most types of client network.
- It is not possible to precompute solutions to the puzzles
- While the client is solving the puzzle the server does not need to store the solution or other client specific data.
- The same puzzle may be given to several clients. Knowing the solution of one or more clients does not help a new client in solving the puzzle.
- A client can reuse a puzzle by creating several instances of it.

Some of the puzzles are, Time-lock puzzle [5], Hash-reversal puzzle [6], multiple hash reversal puzzle [6], Hint based hash reversal puzzle [7]. We here present a new way to use puzzle to mitigate DDOS attack. In this proposed scheme the client puzzle is placed in the ingress router. Client puzzle is placed in the ingress side in-order that the network traffic will be reduced as the spoofed packets are filtered in the beginning it self. Any of the above mentioned puzzles can be used. In this paper we are using the "Hint based hash reversal puzzle" for delivering fine grained puzzles in which a single hash reversal puzzle is given to the client along with a hint that gives the client an idea of where the answer lies. The hint is a single value that is near the answer and solves the coarseness problem to hash-reversal puzzles.

To adjust the difficulty of the puzzle, the accuracy of the hint is increased or decreased. The creation of puzzle is outsourced to a secure entity we call a bastion. For example, suppose a randomly generated number '*x*' is used as an input to the hash *h(x).* To generate a puzzle with *O(D)* difficulty, the issuer passes the client the hash and a hint *x-u(0,D).* Where *u(0,D),* is the randomly chosen number uniformly distributed between *0* and *D.* The client then starts at the hint and searches the range linearly for the answer. The number of hashes done by the client to find *x* varies probabilistically but the expected value is *d/2.* An arbitrary number of servers or routers can use the same bastion, and can safely share the same set of puzzles. Once constructed, the puzzles will be digitally signed by the bastion so that they can be redistributed by anyone. The client can solve the puzzles off-line, so that users don't have to wait for puzzles to be solved. Solving a puzzle gives a client access, for a time interval, to a channel on the server (i.e.,) to a small slice of the servers' resources and the server ensures no virtual channel uses more than its fair share of available resources.

The client must present their solution with the server cookie which was attached to the puzzle. To verify correctness, the server uses the timestamp to index into the nonce table and obtains the corresponding nonce, performs a hash of the client solution with the nonce and checks to see if it matches the echoed server cookie. Across 1000 puzzle verifications on our evaluation systems, the average time was 1.24µs.

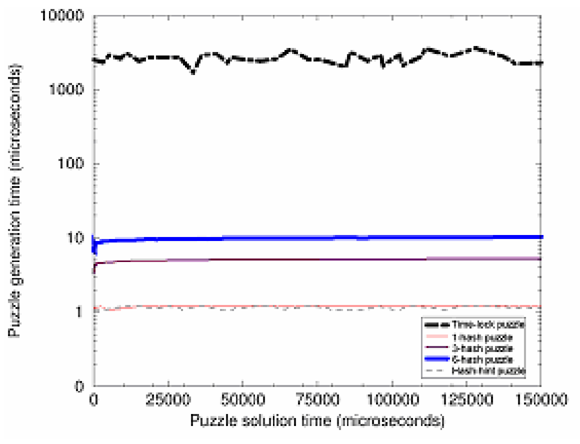

Fig. 3 Puzzle generation versus solution time

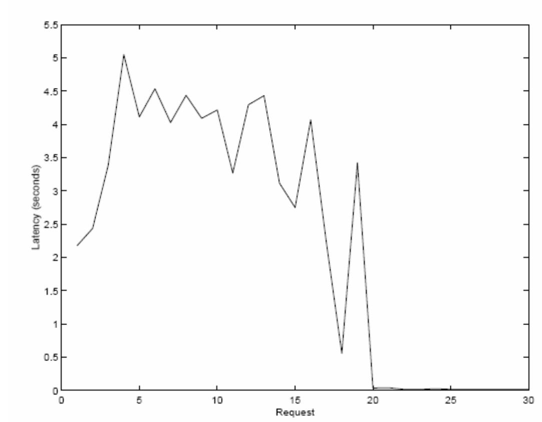

Fig. 4 Latency for a legitimate client without puzzles (During and after attack)

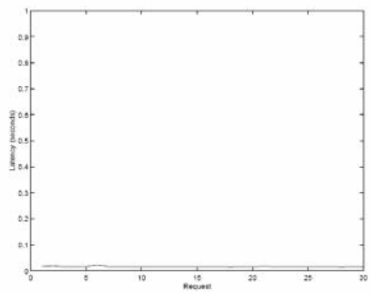

Fig. 5 Latency for a legitimate client without puzzles (During and after attack)

At the end the client who solves the puzzle will be allowed to pass through the path to reach the server which it wants to access. In the graph shown above with and without using the puzzle clearly shows that only a small amount of spoofed packets will be passing through the ingress router. The clients which don't solve the puzzle will be considered spoofed and the IP packet will be discarded.

IV. Path Fingerprint

In this scheme, each IP packet is embedded with a unique path fingerprint representing the route an IP packet is traversed, and IP packets with incorrect path fingerprint is considered spoofed. The proposed scheme eliminates some weakness of conventional schemes and is designed specifically for defending against spoofed DDOS stack. The path fingerprint scheme is placed at the server which the client want s to access the servers resources. To generate a pth fingerprint representing the route an IP packet traversed, it is assumed that each participating router assigns each of its network interface a n-bit random number, and these random numbers are kept securely. A path fingerprint of an IP packet is composed of two fields: a d-bit distance field and an n-bit path identification field where the former represent the number of intermediate routers traversed, and the latter denotes an identifier derived from the random numbers associated with the traversed network information in the route. The path fingerprint of an IP packet is stored in the IP packet header and thus it is delivered to the destination host along with the packet. The path fingerprint procedure is presented as follows, whenever a participating router receives an IP packet; it first examines the distance field. If its value is 0, the receiving router is then aware that it is the first participating router the packet encountered in the path. In this case, the receiving router sets the distance field to 1 and sets the path identification field to the random number associated with the incoming interface of the packet. On the other hand, if the distance field is already a non-zero value its value is just incremented by one and updates the path identification field with $H(PID,N_i)$, where PID represents the current value of path identification field in the packet, N_i denotes the random number of the incoming weak collision resistance.

Algorithm 1: Computation of path fingerprint on a participating router,

1. Let P denote an incoming IP packet
2. P.dist and P.pid denote the distance and path identification fields in packet P respectively.
3. Let Ni denote the random number associated with the incoming interface of P.
4. For the first time the value of P.dist is initialized to 0.
5. P.dist$\leftarrow$P.dist + 1
6. P.pid$\leftarrow$ H(P.pid,Ni)
7. End

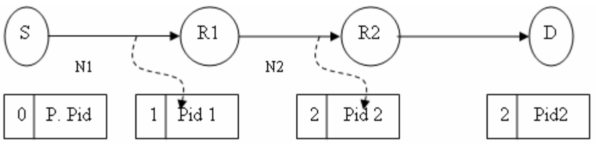

Fig. 6 Path fingerprint

In the example depicted, a packet traverses from the source S to the destination D across routers R1 to R2. The first router in the path R1 sets the distance field to 1 and sets the initial Pid value to the random number of the incoming interface, i.e., N1. Afterwards, each router increases the distance field and updates the Pid field according to the previous Pid value and the random number of the current incoming interface. H denotes the hash function. The 16-bit identification field in the IP header is chosen to be overloaded, the space for storing a path fingerprint. This 16-bit is divided into two fields; one is 5- bit long used to store distance value and the remaining bits are used to store the Pid value.

In the server we maintain a queue for holding the incoming IP packets. We currently recommended that several scheduling techniques such as FCFS (first come first Served), priority, round robin, multilevel queue, multilevel feed back queue can be used for scheduling the incoming IP packets. We don't make any claim that which scheduling algorithm is the best, this may warrants further research. In our scheme the service requests are serviced in FCFS fashion. In this way, filtering of spoofed IP packets will be quite straight forward if the table that contains the mapping of IP addresses and their path fingerprint is present.

A. Construction and Updating of SIPF Table

SIPF table is the table which will be used for discarding the spoofed packets. The table consists of 3 fields namely, IP, Pid, Counter where, IP represents the various source IP address, Pid represents the path fingerprint and counter represents the no. of times a particular source has visited the server. The SIPF table is constructed and its entries will be updated if there are changes in the topology of internet or in the internet paths due to dynamic routing. Here SIPF table have entries for IP addresses that ever connected to the destination in the past communications. So the SIPF takes only a small set of values.

A new entry will be added to the table if the destination host receives IP packet from a new IP address. Thus we can say that SIPF needs only a moderate amount of storage and at the same time, reducing the time for searching. The path fingerprint of a specific source IP address is explicitly explored by the use of ICMP echo-request message. Before an entry of a new source IP address can be made, the destination host sends an ICMP echo-request message to the source IP address. Then the path fingerprint in the returned ICMP echo-reply message is treated as the most upto date path fingerprint of that source IP address.

There are 2 main reasons for invoking exploration of the path fingerprint of a specific IP address. The first refers to the arrival of an IP packet with a new IP address. The second directs to the necessity of updating SIPF table entry. Consider the first case when the packet from a new client arrives, a SIPF table cannot accommodate the mapping of all possible IP addresses replacing an old SIPF table entry with a new one is also an important issue that needs to be addressed. So we need a replacement algorithm. In our scheme the replacement algorithm we use is MFU (most frequently used) algorithm. This algorithm is based on the arguments that the entry with the smallest count was probably just brought in and yet to be used. There can be almost 2^{32} entries. So, when the table is full then the IP address of the source which has frequently visited will be replaced by the new incoming IP packet. The algorithm below shows the updation of SIPF table.

Algorithm 2: Updation of the SIPF table.

1. Let P denote the incoming packet and Q represents the scheduling queue
2. Let P.IP and P.pf denote the source IP address and the path fingerprint stored in the packet header of P.
3. Let ICMP.addr denote the value of ICMP echo request of the new incoming packet
4. If P.pf = SIPF.pf then
5. If SIPF.counter >35 then
6. Delete the entry from the SIPF table, move the IP Packet to the end of the queue Q.
7. Else SIPF.counter $\leftarrow$ SIPF.counter+1
8. Endif
9. Else
10. Send an ICMP echo request to the source
11. If ICMP.pf = P.pf then
12. Update the new entry P.ID, and P.pf to the SIPF table
13. SIPF.Counter $\leftarrow$ 1
14. Else
15. Spoofed, drop the packet P.

16. Endif
17. Endif

The no. of entries that are allowed in the SIPF table is upto the administrator of the internet servers. In our proposed scheme we have assumed that a maximum number of times a particular source can request the server as 35. Once the maximum value is reached the corresponding entry will be deleted. This mechanism is used because the entire request will be responded without much delay and all the clients will get its turn without much delay.

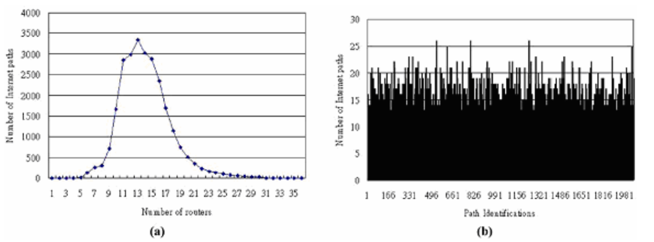

Fig. 7 The distribution of: (a) number of intermediate routers, and (b) the value of path identifications

V. Conclusion and Future Work

In this paper, we presented a defending technique against spoofed DDOS traffic. This technique intends to complement, rather than replace existing schemes. For instance, the proposed scheme helps to discard spoofed packets in the ingress routers by using client puzzles and at the egress side we use the path fingerprint scheme. Furthermore, by weeding out a majority of spoofed attack packets, our approach allows some resource management systems, that share resource fair amount many participants, to work better. In our approach, at the ingress side we place the client puzzle mechanism by which most of the spoofed packets are discarded and the packet which crosses the ingress router is embedded with a unique path fingerprint that represents the Internet path it has traversed. By learning path fingerprints from past traffic, the victim can efficiently establish the SIPF table which contains the mappings of source IP addresses, corresponding path fingerprints, and the frequency by which the a particular client has contacted the server.

A spoofed packet can be easily identified by consulting the SIPF table since it is very unlikely that a spoofed packet can have a path fingerprint identical to that of the spoofed IP address. Thus, by identifying and filtering spoofed packets, a spoofed DDoS attack can be identified and prevented. This makes the proposed scheme an effective and efficient approach for defending against spoofed DDoS attacks.

References

[1] CERT coordination center, “DOS attack”, http://www.cert.org/tech_tips/denial-of-service.html.
[2] TCP SYN flooding and IP Spoofing, CERT advisory CA-96.21,2000(online. Available: http://www.cert.org/advisories/CA-96.21.html.
[3] S.Gibson, “distributed reflection denial of service”, Gibson research Corp., Tech, Rep., Feb 2002(online) Available: http://grc.com/dos/drdos.htm
[4] “Defending against spoofed DDOS attack with path fingerprint”- Fu-Yuan Lee, Shiuhpyng shieh. www.elsevier.com/locate/cose
[5] R.L.Rivest,A.Shamir amd D.A. Wagner, “Time-lock Puzzle and Times-release crypto”, MIT/LCS/TR-684,1996.
[6] A.Juels and J.Brainard, “Client Puzzle: A Cryptographics defense against connection depletion”, in NDSS, 1999, PP. 151-165.
[7] WU.Chang Feng, ED Kaiser, WC-chifeng, Antoine Luu, “ The design and implementation of network Puzzles”
[8] Li J, Mirkovic J, Wang M, Reiher P, Zhang L. Save: source address validity enforcement protocol. In: Proceedings of IEEE INFOCOM, vol. 3; June 2001. p. 1157e566.
[9] Mirkovic J, Prier G, Reiher P. Attacking DDoS at the source. In: Proceedings of international conference on network protocols;Nov. 2002. p. 312e21.
[10] Belenky A, Ansari N. IP traceback with deterministic packet marking. IEEE communications Letters April 2003;7(2): 162e4.
[11] Bellovin S, Leech M, Taylor T. ICMP traceback messages [Online]. Available from: http://www.ietf.org/internet-drafts/draftietf - itrace-04.txt; Feb. 2003.
[12] Dean D, Franklin M, Stubblefield, A. ‘‘An algebraic approach to IP traceback’’. ACM Transactions on Information and System Security May 2002;5(2):119e37.
[13] Sanchez LA, Milliken WC, Snoeren AC, Tchakountio F, Jones CE, Kent ST, et al. Hardware support for a hash-based IP traceback. In: Proceedings of the second DARPA information survivability conference; June 2001. p. 146e52.
[14] Savage S, Wetherall D, Karlin AR, Anderson T. Practical network support for IP traceback. In: Proceedings of SIGCOMM conference; Aug. 2000. p. 295e306.
[15] Savage S, Wetherall D, Karlin AR, Anderson T. Network support for IP traceback. IEEE/ACM Transactions on Networking June 2001;3:226e37.
[16] Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F, Kent ST, et al. Hash-based IP traceback. In: Proceedings of the ACM SIGCOMM conference; Aug. 2001. p. 3e14.
[17] Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F, Schwartz B, et al. ‘‘Single-packet IP traceback’’. IEEE/ACM Transactions on Networking 2002; 10(6) :721e34.
[18] Song D, Perrig A. Advanced and authenticated marking schemes for IP traceback. In: Proceedings of IEEE INFOCOM conference; Apr. 2001. p. 878e86.
[19] Jin C, Wang H, Shin KG. Hop-count filtering: an effective defense against spoofed ddos traffic. In: Proceedings of ACM conference on computer and communications security; Oct. 2003. p. 30e41.
[20] Peng T, Leckie C, Ramamohanarao K. Detecting distributed denial of service attacks using source IP address monitoring. Australia: The University of Melboume; 2002. Tech.http://www.ee.mu.oz.au/pgrad/taop/research/detection.pdf
[21] Peng T, Leckie C, Ramamohanarao K. Protection from distributed denial of service attacks using history-based IP filtering. In: Proceedings of IEEE international conference on communications, vol. 1; May 2003. p. 482e6.
[22] Sung M, Xu J. IP traceback-based intelligent packet filtering: a novel technique for defending against internet DDoS attacks. In: Proceedings of international conference on network protocols; Nov. 2002. p. 302e11.
[23] Sung M, Xu J. IP traceback-based intelligent packet filtering: a novel technique for defending against internet DDoS attacks. IEEE Transactions on Parallel and Distributed Systems Sep. 2003;14(9):861e72.
[24] Yaar A, Perrig A, Song D. Pi: a path identification mechanism to defend against DDos attacks. In: Proceedings of the IEEE symposium on security and privacy; May 2003. p. 93e109.
[25] D. Moore, C. Shannon, and J. Brown, “Code-Red: A Case Study on the Spread and Victims of an InternetWorm,” in Internet Measurement Workshop, November 2002.
[26] S. Staniford, V. Paxson, and N. Weaver, “How to 0wn the Internet in Your Spare Time,” in 11th USENIX Security Symposium (Security ’02), 2002.
[27] CERT, “CERT Advisory CA-2004-02 Email-borne Viruses,” http://www.cert.org/advisories/CA-2004-02.html, 2004.

[28] C. Dwork and M. Naor, “Pricing via Processing or Combatting Junk Mail,” in Crypto, 1992.
[29] R. Merkle, “Secure Communications Over Insecure Channels,” Communications of the ACM, vol. 21, no. 4, April 1978.
[30] L. von Ahn, M. Blum, N. Hopper, and J. Langford, “CAPTCHA: Using Hard AI Problems for Security,” in Eurocrypt 2003., 2003.
[31] A. Juels and J. Brainard, “Client Puzzles: A Cryptographic Defense against Connection Depletion,” in NDSS, 1999, pp. 151–165.
[32] D. Dean and A. Stubblefield, “Using Client Puzzles to Protect TLS,” in 10th Annual USENIX Security Symposium, 2001.
[33] T. Aura, P. Nikander, and J. Leiwo, “DOS-Resistant Authentication with Client Puzzles,” Lecture Notes in Computer Science, vol. 2133, 2001.
[34] J. Leiwo, T. Aura, and P. Nikander, “Towards Network Denial of Service Resistant Protocols,” in SEC, 2000, pp. 301–310.
[35] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D.Wallach, “Security for Peer-to-Peer Routing Overlays,” in Proceedings of OSDI, December 2002.
[36] M. Abadi, M. Burrows, M. Manasse, and T.Wobber, “Moderately Hard, Memory-bound Functions,” 2003.
[37] I. Clarke, O. Sandberg, B. Wiley, and T. Hong, “Freenet: A Distributed anonymous Information Storage and Retrieval System,” Lecture Notes in Computer Science, vol. 2009, pp. 46+, 2001.
[38] X. Wang and M. Reiter, “Defending Against Denial-of-Service Attacks with Puzzle Auctions,” in IEEE Symposium on Security and Privacy, 2003.
[39] W. Feng, “The Case for TCP/IP Puzzles,” in ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA-03), Karlsruhe, Germany, August 2003.
[40] CERT Coordination Center. CERTR incident note IN-99-07 distributed denial of service tools [Online]. Available from: http://www.cert.org/incident_notes/IN-99-07.html; Jan. 1999a.
[41] CERT Coordination Center. Results of the distributed-systems intruder tools workshop [Online]. Available from: http://www.cert.org/reports/dsit-workshop-final.html, http:// www.cert.org/reports/dsit-workshop.pdf ; Nov. 1999.
[42] CERT Coordination Center. CERTR advisory CA-1999-17 denialof-service tools [Online]. Available from: http://www.cert.org/advisories/CA-1999-17.html; Dec. 1999.
[43] CERT Coordination Center. CERTR advisory CA-2000-01 denial-of-service developments [Online]. Available from: http://www.cert.org/advisories/CA-2000-01.html Jan. 2000.
[44] Cheswick B, Burch H, Branigan S. Mapping and visualizing the internet. In: Proceedings of USENIX annual technical conference [Online]. Available from: http://www.usenix.org/publications/library/proceedings/usenix2000/general/ cheswick.html; June 2000.
[45] Darmohray T, Oliver R. ‘‘Hot spares’’ for DoS attacks;login [Online]. Available from: http://www.usenix.org/publications/login/2000-7/apropos.html ; July 2000.
[46] Dean D, Franklin M, Stubblefield, A. ‘‘An algebraic approach to IP traceback’’. ACM Transactions on Information and System Security May 2002;5(2):119e37.
[47] Dittrich D. The DoS project’s trinoo distributed denial of service attack tool [Online]. Available from: http://staff.washington.edu/dittrich/misc/trinoo.analysis; Oct. 1999a.
[48] Dittrich D. The tribe flood network distributed denial of service attack tool [Online]. Available from: http://staff.washington.edu/dittrich/misc/tfn.analysis; Oct. 1999b.
[49] Ferguson P, Senie D. Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing. Internet engineering task force, RFC 2827 [Online]. Available from: http://www.rfc-editor.org/rfc/rfc2827.txt; May 2000.

V. Praveena, Coimbatore, 15.07.1981. Completed the B.E., degree in Computer Science and Engineering in Maharaja Engineering College, Coimbatore, Tamilnadu, India in 2002.

She worked as the software trainee in M/s Kalpatharu software solutions, Coimbatore, India, from Sep 2002 to Nov 2002. She worked as a Lecturer in Maharaja Engineering College, Coimbatore, India, from Nov 2002 to Jan 2004 in the Department of Information Technology. She worked as a software programmer in G-Net solutions, Coimbatore, India, from Dec 2002 to March 2004 in the Department of Information Technology. She worked as a Lecturer in Park college of Engineering and Technology, Coimbatore, India, from Jan 2004 to August 2007 in the Department of Computer Science and Engineering. She is working as a Lecturer in Dr. N.G.P. Institute of Technology, Coimbatore, Tamilnadu, India, from August 2007 to till date in the Department of Computer Science and Engineering. She has published a book: Fundamentals of Computing and Computer Programming (Coimbatore, Tamilnadu, India: Pratheeba Publishers, 2008).

Ms. Praveena is a member of ISTE.

N. Kiruthika, Coimbatore, Completed B.E., degree in Information Technology in Bannari amman Institute of Technology, Sathyamangalam, Tamilnadu, India in 2006. Completed M.E., degree in Tamilnadu Engineering College, Coimbatore, Tamilnadu, India in 2008.

She is working as a Lecturer in Dr. N.G.P. Institute of Technology, Coimbatore, Tamilnadu, India, from June 2008 to till date in the Department of Information Technology.

Ms. Kiruthika is a member of ISTE.