Detection Architecture of Application Layer DDoS Attack for Internet

Sanjay B Ankali
Department of Information Science & Engg, SJBIT, Bangalore, India
Email: sanjay.ankali@yahoo.com

Dr. D V Ashoka,
Professor & Head, Department of Information Science & Engg, SJBIT, Bangalore, India
Email: dr.ashok_research@hotmail.com

---**ABSTRACT**---

Internet was intended with functionality and not Security in mind. For this reason, its architecture has some intrinsic weaknesses and bugs called vulnerability which results in successful origin of DDOS attacks. Over the time, researchers proposed many solutions to prevent the DDOS attack from different OSI layers, on the other hand none have seen proper deployment and there were very a small number of researches on layer Seven. This paper designs two independent architectures for HTTP and FTP which uses an extended hidden semi-Markov model is proposed to describe the browsing habits of web searchers. A forward algorithm is resulting for the online implementation of the model based on the M-algorithm in order to reduce the computational amount introduced by the model's large state space.

Keywords - ***Application-layer, distributed denial of service (DDoS), Denial of Service (DOS), FTP and HTTP.***

Date of Submission : April 28, 2011 Date of Acceptance : June 01, 2011

I. INTRODUCTION

The name "availability" means that the information, the computing systems and the security controls are all accessible and operable in committed state at some random point of time. Threat to the Internet availability is a big question which is hampering the growth and continued existence of e-business and other Internet based applications. The Internet like any other product is also prone to failures. Internet failures can be accidental or intentional. The Internet design concentrates mainly on providing functionality though a little concentration has been given on designing strategies for controlling accidental failures. On the other hand, intentional attacks by malicious users/hackers/crackers have no answer in the original Internet design. A Denial of Service (DoS) is such an intentional attempt by malicious users/attackers to completely disrupt or degrade availability of service/resource to genuine/authorized users [1]. Some well-known DoS attacks are SYN Flood, teardrop, smurf, ping of death, land, finger bom, black holes, octopus, snork, ARP Cache poisoning and the misdirection. DoS attacks exploit weaknesses in Internet protocols, applications, operating systems and protocol implementation in operating systems. Distributed Denial of Service (DDoS) attacks degrade or completely disrupt services to genuine users by expending communication and/or computational resources of the target. Mirkovic et al. [2] described DDoS attacks as amplified form of DoS attacks, where attackers direct hundreds or even thousands of compromised hosts called zombies against a single target. These zombie hosts are innocently recruited from the millions of unprotected computers accessing the Internet through high-bandwidth and always available connections.

DDoS attack has caused severe damage to servers and will cause even greater threats to the development of new Internet services. Conventionally, DDoS attacks are carried out at the network layer, such as ICMP flooding, SYN flooding and UDP flooding, which are called Net DDoS attack. This paper proposed different schemes (e.g., network measure or anomaly detection) to protect the network and equipment from bandwidth attacks, it is not as easy as in the past for attackers to launch the DDoS attacks based on network layer. When the simple Net-DDoS attacks fail, attackers shift their distasteful strategies to application-layer attacks and establish a more sophisticated type of DDoS attacks. To circumvent detection, they attack the victim Web servers by HTTP GET requests (e.g., HTTP Flooding) and pulling large image files from the victim server in overwhelming numbers. In another case, attackers run a massive number of queries through the victim's search engine or database query to bring the server down [1]. Such attacks are called application-layer DDoS (App-DDoS) attacks. The MyDoom worm [3] and the CyberSlam [4] are all instances of this type attack Surfers.

II. LITERATURE SURVEY

Long-ago the research has been done to detect the DDOS attack from three different layers of OSI namely **network layer, Transmission layer and Application layer** but the work done on layer 7 is very fewer because the attacks where very little in past, the techniques are highlighted below.

1) Client Puzzle Protocol

Client Puzzle Protocol (CPP) is an algorithm for use in Internet communication, whose goal is to make abuse of server resources infeasible. The idea of the CPP is to necessitate all clients connecting to a server to correctly solve a mathematical puzzle before establishing a connection, if the server is under attack. After solving the puzzle, the client would return the solution to the server, which the server would quickly confirm, or reject and drop the connection. The puzzle is made simple and easily solvable but requires at least a minimal amount of computation on the client side. Genuine users would experience just a negligible computational cost but abuse would be deterred: those clients that try to simultaneously establish a large numbers of connections would be unable to do so because of the computational cost (time delay). This method holds promise in fighting some types of spam as well as other attacks like Denial of Service.

2) Intrusion Detection System

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of probable incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies or standard security practices. Incidents have many causes, such as malware (e.g., worms, spyware), attackers gaining unauthorized access to systems from the Internet and authorized users of systems who misuse their privileges or attempt to add additional privileges for which they are not authorized. An intrusion detection system (IDS) is software that automates the intrusion detection process. Primarily, IDS is concerned with the detection of hostile actions.

3) Factoring problem

Factoring is the act of splitting an integer into a set of smaller integers (factors) which, when multiplied together, form the original integer. For example, the factors of 403 are 13 and 31; the factoring problem is to find 13 and 31 when given 403. Prime factorization requires splitting an integer into factors that are prime numbers; every integer has a unique prime factorization. Multiplying two prime integers together is easy but factoring the product is much more difficult. No high-quality algorithms exist to solve this problem in polynomial time and the best algorithm which solves this problem in less complexity is general number field sieve in $O(\exp((64/9b)1/3.(\log b)2/3))$ for a b-bit integer. For a quantum computer it takes $O(b3)$ by using Shor‟s algorithm.

4) Ingress filtering

In computer networks, ingress filtering is a technique used to make sure that incoming packets are actually from the networks that they claim to be from. Generally networks receive packets from other networks. Normally a packet will contain the IP address of the computer that originally sent it. This allows other computers in the network to know where it came from, which is needed for things like sending a packet back to the sending computer. In certain cases, the sending IP address will be spoofed. This is typically done as part of an attack, so that the attacked computer does not know where the attack is really coming from. Filtering a packet is when the packet is not processed normally but is denied in some way. The computer processing the packet might simply pay no attention to the packet completely or where it is possible it might send a packet back to the sender saying the packet is denied. In ingress filtering, packets coming into the network are filtered if the network sending it should not send packets from IP address of the originating computer.

In order to do ingress filtering, the network wants to know which IP addresses each of the networks it is connected to may send. This is not always potential. For instance, a network that has a single connection to the Internet has no way to know if a packet coming from that connection is spoofed or not. Edge networks, whether multi-homed or not, usually have a limited number of address blocks in use. Such edge networks should filter packets leaving their networks, verifying the source IP address in all packets is within the address blocks allocated. Enterprises, universities and others who run edge networks should be doing this. The idea is to prevent computers on your network from spoofing (acting as another). Implementation for edge networks of egress packets in this way is very simple and should be done with access lists.

5) Threshold Value

The threshold value is the number of requests that a server can handle without straining its resources. It is defined as a predetermined percentage of the maximum number of requests that a server can handle.

III. DDOS OVERVIEW

The operating systems and network protocols are developed without applying security engineering which results in providing hackers a lot of insecure machines on Internet. These insecure and unmatched machines are used by DDoS attackers as their army to launch attack. An attacker gradually implants attack programs on these insecure machines. Depending upon complexity in logic of implanted programs these compromised machines are called Masters/Handlers or Zombies and are collectively called bots and the attack network is called botnet in hacker’s community. Hackers send control instructions to masters, which in turn communicate it to zombies for launching attack. The zombie machines under control of masters/handlers (running control mechanism) as shown in Figure1 transmit attack packets, which converge at victim or its network to exhaust either its communication or computational resources. This paper classified DDOS attacks into two broad categories: flooding attacks and vulnerability attacks. Flooding DDoS attacks consume resources such as network bandwidth by overwhelming bottleneck link with a high volume of packets. Vulnerability attacks use the expected behavior of protocols such as TCP and HTTP to the attacker's advantage. The computational resources of the server are tied up by seemingly legitimate requests of the attackers and thus prevent the server from processing transactions or requests from authorized users. Flooding DDoS is basically a resource overloading problem. The resource can be bandwidth, memory, CPU cycles, file descriptors and buffers etc., the attackers bombard the scarce resource(s) by sheer flood of packets. In Figure 2 a flood of packets is shown, which congests the link between ISP's edge router and border router of victim domain. Attack packets keep coming as per distribution fixed by attacker, whereas legitimate clients cut short their packet sending rates as per flow control and congestion signals. A situation comes when whole of bottleneck bandwidth is seized by attack packets.

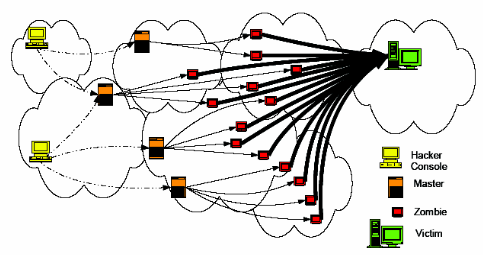

Fig 1. Attack modus operandi.

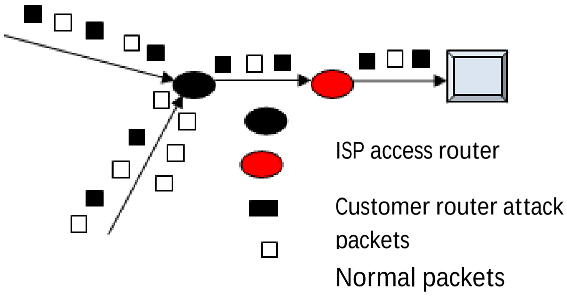

Fig 2. Packet drops

Thus, service is denied to legitimate users due to limited bottleneck bandwidth. However, resources of connecting network are not a problem in case of commercial servers as these are hosted by the ISPs, quite close to their backbone network with high bandwidth access links. But server resources such as processing capacity, buffer limit etc., are put under stress by flood of seemingly legitimate requests generated by DDoS attack zombies. Each request consume some CPU cycles. Once the total request rate is more than the service rate of server, as shown in Figure 2, the requests start getting buffered in the server and after some time due to buffer over run, incoming requests are dropped. The congestion and flow control signals force legitimate clients to decrease their rate of sending requests, whereas attack packets keep coming. Finally, a stage comes when only attack traffic is reaching at the server. Thus, service is denied to legitimate clients. Moreover, Robinson et al. [8] highlights that as attack strength grows by using multiple sources, the computational requirements of even filtering traffic of malicious flows become a burden at the target.

Even though DoS attacking strategies differ in time, studies show that attackers mainly target the following resources to cause damage on victim [9].

Network bandwidth resources: This is related with the capacity of the network links connecting servers to the wider Internet or connectivity between the clients and their Internet Service Providers (ISP). Usually, the bandwidth of client's internal network is less than its connectivity with the external network. Thus the traffic that comes from the Internet to the client may consume the entire bandwidth of the client's network. Thus, a legitimate request will not be able to get service from the targeted network. In a DoS attack, the vast majority of traffic directed at the target network is malicious; generated either directly or indirectly by an attacker. These attacks prevented 13,000 Bank of America ATM from providing withdrawn services and paralyzed such large ISPs as Freetel, SK Telecom, and Korea Telecom on January 25, 2003.

1) System memory resources: An attack targeting system memory resources typically aims to crash its network handling software rather than consuming bandwidth with large volume of traffic. Specific packets are sent to confuse the operating system or other resources of the victim's machine. These include temporary buffer used to store arriving packets, tables of open connections and similar memory data structures. Another system resource attack uses packets whose structures trigger a bug in the network software, overloading the target machine or disabling its communication mechanism or making a host crash, freeze or reboot which means the system can no longer communicate over the network until the software is reloaded.

2) System CPU resources/ Computational Capacity: An attack targeting system's CPU resources typically aims to employ a sequence of queries to execute complex commands and then overwhelmed the CPU. The Internet key Exchange protocol (IKE) is the current IETF standard for key establishment and SA parameter negotiation of IPsec. However, IKE's aggregate mode is still very susceptible to DoS attacks against both computational and memory resources because the server has to create states for SA and compute Diffie-Hellman exponential generation [14].

IV. PROBLEMS WITH DDOS DETECTION

The main aim of a DDoS defense system is to relieve victim’s resources from high volume of counterfeit packets sent by attackers from distributed locations, so that these resources could be used to serve legitimate users. There are four approaches to combat with DDoS attack as proposed by Douligeris et al. [10]: Prevention, Detection and Characterization, Trace back, and Tolerance and Mitigation. Attack prevention aims to fix security holes, such as insecure protocols, weak authentication schemes and vulnerable computer systems, which can be used as stepping stones to launch a DoS attack. This approach aims to improve the global security level and is the best solution to DoS attacks in theory. Attack detection aims to detect DDoS attacks in the process of an attack and characterization helps to distinguish attack traffic from legitimate traffic.

Trace back aims to locate the attack sources regardless of the spoofed source IP addresses in either process of attack (active) or after the attack (passive). Tolerance and mitigation aims to eliminate or curtail the effects of an attack and try to maximize the Quality of Services (QoS) under attack. Carl et al. Douligeris et al. and Mirkovic et al. have reviewed a lot of research schemes based on these approaches but still no comprehensive solution to tackle DDoS attacks exist. One of the main reasons behind it is lack of comprehensive knowledge about DDoS incidents. Furthermore the design and implementation of a comprehensive solution which can defend Internet from variety of DDoS attacks is hindered by following challenges [11]:

- Large number of unwitting participants.
- No common characteristics of DDoS streams.
- Use of legitimate traffic models by attackers.
- No administrative domain cooperation.
- Automated DDoS attack tools.
- Hidden identity of participants because of source addresse spoofing.
- Persistent security holes on the Internet.
- Lack of attack information.
- Lack of standardized evaluation and testing approaches.

In order to build a comprehensive DDoS defense solution in light of these challenges, Robinson et al. recommended following DDoS defense principles:

- As DDoS is a distributed attack and because of high volume and rate of attack packets distributed instead of centralized defense is the first principle of DDoS defense.
- High Normal Packet Survival Ratio (NPSR) (ratio of number of normal packets received to total number of packets reaching at the server), i.e., less collateral damage is the prime requirement for a DDoS defense.
- A DDoS defense method should provide secure communication for control messages in terms of confidentiality, authentication of sources, integrity and freshness of exchanged messages between defense nodes.
- A partially and incrementally deployable defense model is successful as there is no centralized control for Autonomous Systems (AS) in Internet.
- A defense system must take into account future compatibility issues such as interfacing with other systems and negotiating different defense policies.

Technique used or algorithm used:

The M-algorithm is being widely adopted in decoding digital communications because it requires far fewer computations than the Viterbi algorithm. The aim of the M-algorithm is to find a path with distortion or likelihood metrics as good as possible (i.e., minimize the distortion criterion between the symbols associated to the path and the input sequence).

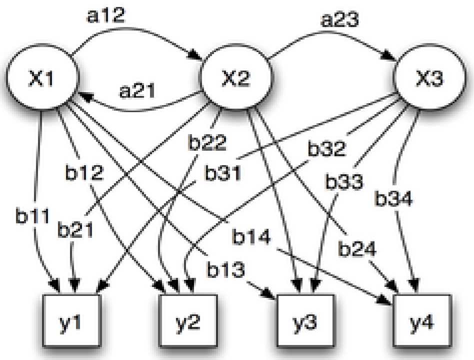

Fig.3 Markov chain

Probabilistic parameters of a hidden Markov model (example)
x—states
y—possible observations
a — state transition probabilities
b — output probabilities

A **hidden Markov model** (**HMM**) is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved state. An HMM can be considered as the simplest dynamic Bayesian network.

In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a *hidden* Markov model, the state is not directly visible, but output dependent on the state is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by a HMM gives some information about the sequence of states. Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still 'hidden'.

Hidden Markov models are particularly known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bioinformatics.

A **hidden semi-Markov model** (HSMM) is a statistical model with the similar structure as a hidden Markov model except that the unobservable process is semi-Markov rather than Markov. This means that the probability of there being a change in the hidden state depends on the amount of time that has elapsed since entry into the current state. This is in contrast to hidden Markov models where there is a constan**1.** probability of changing state given survival in the state up to that time. For instance Sansom *et al.* modelled daily rainfall using a hidden semi-Markov model. If the underlying process (e.g. weather system) does not have a geometrically distributed duration, an HSMM may be more appropriate. Statistical inference for hidden semi-Markov models is more difficult than in hidden Markov models, since algorithms like the Baum-Welch algorithm are not directly applicable, and must be adapted requiring more resources.

V. SYSTEM ANALYSIS

Existing System:

Internet DDoS attack is real threat on websites such as Yahoo, CNN, Amazon, eBay, etc i.e. services were unavailable for several hours due to Lack of defense mechanism on current Internet and also for individual Systems. The on hand feature for user behaviors can be summarized as the following ways. The first is based on probabilistic model, a double Pareto distribution for the link-choice, and a log-normal distribution for the revisiting, etc. The second is based on click-streams and web content, e.g., data mining to capture a web user's usage patterns from the click-streams dataset and page content. The third is based on the Markov model, e.g. Markov chains to model the URL access patterns that are observed in navigation logs based on the previous state.

Disadvantages:

1. This Systems do not take into account the user's series of operations information (e.g., which page will be requested in the next step)

And methods need intensive computation for page content processing and data mining and hence they are not very suitable for on-line detection.

2. The methods omit the dwell time that the user stays on a page while reading and they do not consider the cases that a user may not follow the hyperlinks provided by the current page

3. It is very hard to identify DDoS attack flows at sources since the traffic is not so aggregate.

4. From a network's perspective, protecting is considered ineffective. Attack flows can still incur congestion along the attack path. So it leads to network congestion.

Proposed System

In the proposed System it can able to detect DDos attack based on TCP connection and web user browsing behavior can be abstracted and profiled by users' request sequences. As a result, one can use a universal model to profile the short-term web browsing behavior and we only need the logs of web server to build the model without any additional support from outside of the web server. Browsing behavior can be described by three elements: HTTP request rate, page viewing time and requested sequence (i.e., the requested objects and their order).

Advantages:

One can make these systems to take into account the user's series of operations information. There is an intensive computation for page content processing and data mining, and hence they are very suitable for on-line detection. The dwell time that the user stays on a page while reading and we can find cases that a user may follow the hyperlinks provided by the current page.

FOR DDOS ATTACK:

1. *Distribution*: the number of hosts sending packets to the destination in each observation period

2. *Continuity*: reflect to the observation that a DDoS attack always lasts for an extended period of time.

3. The effectiveness of packet filter is the best

VI DETECTION ARCHITECTURE

The overall procedure of this detection architecture is illustrated in Fig. 3. The scheme is divided into three phases: **login, anomaly detection, prevention**

Database Design: SQL server 2005

In this design we are using total 5 tables:

Login: Login table containing username and password

Access: contains information (like username, password, and IP address) about all users who have accessed the particular site for some period of time.

Adminlog: This table containing username and password

Browselog: This table containing full user browsing details like user who browsed, countlog, start time, end time, website address, system name and date.

Service: This table containing server IP address, user name, file size and fcount.

Srm: This table containing ID and name to be displayed .

Front End: C#.NET

Login/Registration:

The Valid user enter into login to send data to available network systems, if the user doesn't register it will move to new user creation from. In this Module Collecting the general user details and store database for future references. It is having Name, Password, Confirm Password, and Email address.

Anomaly detection:

Anomaly detection relies on detecting behaviors that are abnormal with respect to some normal standard. Many anomaly detection systems and approaches have been developed to detect the faint signs of DDoS attacks.

Browsing behavior:

Website can be characterized by the hyperlinks among the WebPages and the number of in-line objects in each page.

When users click a hyperlink pointing to a page, the browser will send out a number of requests for the page and its several in-line objects. The above details help to easily detect the browsing behavior.

Prevent the attack:

By the use of a DDoS tool the source IP address of the attacking packets can be spoofed and this way the true identity of the secondary victims is prevented from exposure and the return packets from the victim system. Then deny the access of the users.

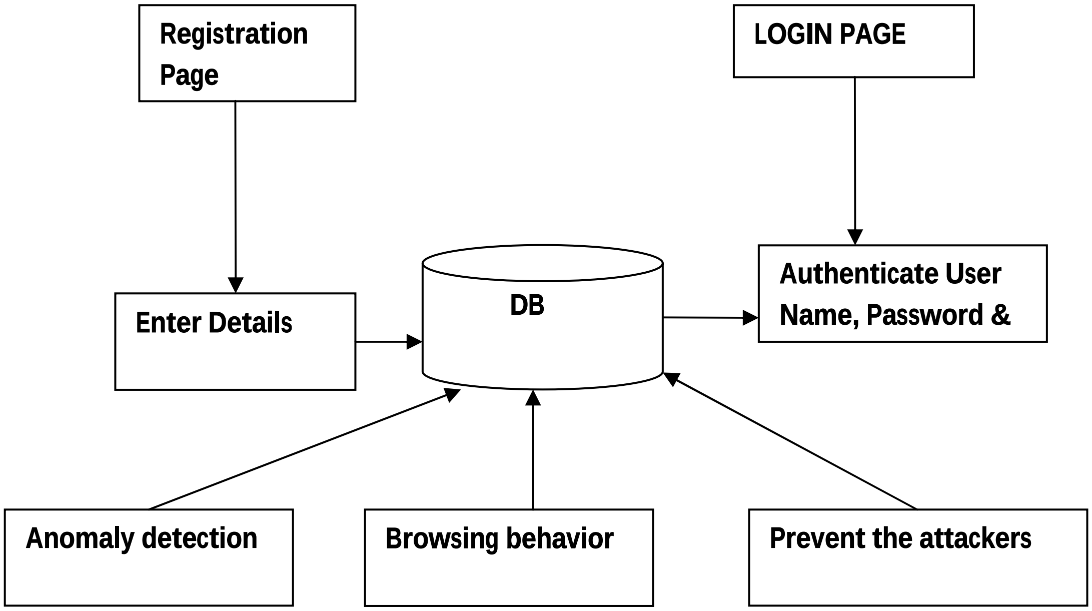

Fig. 4. Detection Architecture.

VII. DISCUSSIONS

The conventional security technologies such as firewalls [16] Intrusion Detection Systems (IDSs) [17] and access control lists in routers are unable to defend networks from these attacks. The stumbling barrier against these attacks is that it is almost impossible to differentiate between genuine and attack packets. Since the potency of flooding DDoS attacks does not depend upon exploitation of software bugs or protocol vulnerabilities, it only depends on the volume of attack traffic. Consequently, flooding DDoS packets do not need to be malformed, such as invalid fragmentation field or a malicious packet payload. As a result, the flooding DDoS traffic looks very comparable to legitimate traffic [18]. Also IP spoofing and stateless routing reduces the chances of attacker being caught. Moreover, flooding DDoS attacks are very dynamic to elude existing defense systems. Therefore, it has become a real challenge to defend against these attacks. The seriousness of DDoS problem and growing sophistication of attackers have led to development of numerous defense mechanisms. But still, the growing number of DDoS attacks and their financial implications press the need of a comprehensive solution. Moreover, as attackers share their attack codes similarly to fight against these attacks, Internet community needs to devise better ways to accumulate details of attack. Only then a comprehensive solution against DDoS attacks can be devised.

CONCLUSION

Creating defenses for attacks requires monitoring dynamic network activities in order to obtain timely and signification information. As most current effort focuses on detecting Net-DDoS attacks with stable background traffic. This paper highlights detection architecture aiming at monitoring Web traffic in order to reveal dynamic shifts in normal burst traffic, which might signal onset of App-DDoS attacks during the flash crowd event. This method reveals early attacks merely depending on the threshold specified and gives all the privilege for administrator who can effectively identify and **block** the connections for specified attacking host. This architecture is expected to be practical in monitoring App-DDoS attacks and in triggering more dedicated detection on victim network.

REFERENCES:

[1] George Coulouris, Jean Dollimore, DISTRIBUTED SYSTEMS, 4th EDITION, pearson education 2005.

[2] Chang C., “Defending Against Flooding-Based Distributed Denial of Service Attacks: A

Tutorial,"*Computer Journal of IEEECommunication Magazine*, vol. 40, no. 10, pp. 42-51, 2002.

[3] "Incident Note IN-2004-01 W32/Novarg. A Virus," CERT, 2004. [Online].Available: http://www.cert.org/incident_notes/ IN-2004-01.html

[4] S. Kandula, D. Katabi, M. Jacob, and A. W. Berger, "Botz-4-Sale: Surviving Organized DDoS Attacks that Mimic Flash Crowds,"MIT, Tech. Rep. TR-969, 2004 [Online].Available:http://www.usenix.org/events/nsdi05/tech/kandula/kandula.pdf

[5] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites," in *Proc. 11th IEEE Int. World Wide Web Conf.*, May 2002, pp. 252–262.

[6] S.-Z. Yu and H. Kobayashi, "An efficient forward-backward algorithm for an explicit duration hidden Markov model," *IEEE Signal Process Lett.*, vol. 10, no. 1, pp. 11–14, Jan. 2003.

[7] L. I. Smith, A Tutorial on Principal Components Analysis [EB/OL], 2003 [Online]. Available: http://www.snl.salk.edu/~shlens/pub/ notes/pca.pdf

[8] A. Hyvärinen, "Survey on independent component analysis," *Neural Comput. Surveys*, vol. 2, pp. 94–128, 1999.

[9] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 626–634, Jun. 1999.

[10] Douligeris C. and Mitrokotsa A., "DDoS Attacks and Defense Mechanisms: Classification and State of the Art," *Computer Journal of Networks*, vol. 44, no. 5, pp. 643-666, 2004

[11] Mirkovic J. and Reiher P., "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *Computer Journal of ACM SIGCOMM*, vol. 34, no. 2, pp. 39-53, 2004.

[12] [Online]. Available: http://ita.ee.lbl.gov/html/traces.htm

[13] A. M. G. Cooper, R. Tsui, and M. Wagner, Summary of Biosurveillance- Relevant Technologies. [Online]. Available: http://www.cs.cmu. edu/~awm/biosurv-methods.pdf

[14] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the selfsimilar nature of ethernet traffic (extended version)," *IEEE/ACM Trans Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[15] Bai Y. and Kobayash H., "Intrusion Detection Systems: Technology and Development," in Proceedings of the 17th International Conference on Advanced Information Networking and Applications, USA, pp. 710-715, 2003.

[16] Mirkovic J. and Reiher P., "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," Computer Journal of ACM SIGCOMM, vol. 34, no. 2, pp. 39-53, 2004.

[17] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: Classification and state-of-the-art," *Computer Networks: The Int. J. Computer and Telecommunications Networking*, vol. 44, no. 5, pp. 643–666, Apr. 2004.

[18] J. Mirkovic, G. Prier, and P. L. Reiher, "Attacking DDoS at the source," in *Proc. 10th IEEE Int. Conf. Network Protocols*, Sep. 2002, pp. 312–321.

Authors Biography

Dr. D.V Ashoka, presently working as a Professor and Head, Department of Information Science and Engineering, S.J.B. Institute of Technology, Bangalore. He received his M.Tech from VTU and Ph.D degree in Computer Science and Engineering from Dr. MGR, University, Chennai. He has more than 16 years of academic, administrative and research experience. His fields of interest are Requirement Engineering, Operating System, Computer Organization, Software Architecture and Cloud Computing. He was published more than 25 papers in national, international conferences and journals.

Mr. Sanjay B Ankali, studying in 4th sem M.Tech, in computer networking at SJBIT, Bangalore **(**VTU Belgaum). His fields of interest are Computer networking, Operating System, Resource Virtualization and Knowledge Management.