A Survey On Various Defense Mechanisms Against Application Layer Distributed Denial Of Service Attack

Ms. Anuja R. Zade

Bharati Vidyapeeth University College Of Engineering , Pune.
anujazade@gmail.com

Dr. Suhas .H. Patil

Bharati Vidyapeeth University College Of Engineering , Pune.
shpatil@bvucoep.edu.in

***Abstract*—The severity of application layer Distributed Denial of Service attack has become a major threat to network operators nowadays. Application layer Distributed Denial of Service attack causes unavailability of resources, revenue loss, and customer churns etc and is one of the most difficult problems to defend against in the internet. The goal of paper is to understand various techniques proposed recently to defend against specifically application layer Distributed Denial of Service attack. The next step in the path would be to come up with a better solution to solve this problem. Undoubtedly the following discussed techniques are extremely good. A collaborative approach using few of them can be a better solution.**

Keywords-component; DDoS (Distributed Denial of Service attack), Kalman filter, K-means,

I. INTRODUCTION

A DOS (Denial of Service) attack renders the computer or computer network unavailable to provide its normal service instead of directly damaging the data, or subverting the service itself. A DDoS (Distributed Denial of Service) attack is multiplication of DOS where attacker makes use of multiple computers (secondary victims) to make a coordinated DOS attack on a victim Computer or network. There are two main classes of DDoS:

- **Resource flooding:** Here the attacker consumes victim computers resources (memory, CPU, hard disk) to make it unavailable for legitimate users. Resource flooding can be further classified into two types Protocol exploit attack and Malformed packet attack.
- **Bandwidth flooding:** Here the victim network is flood by unwanted traffic to prevent the legitimate traffic from reaching the victim network. Bandwidth flooding can be further classified into two types flood attack and Amplification attack.

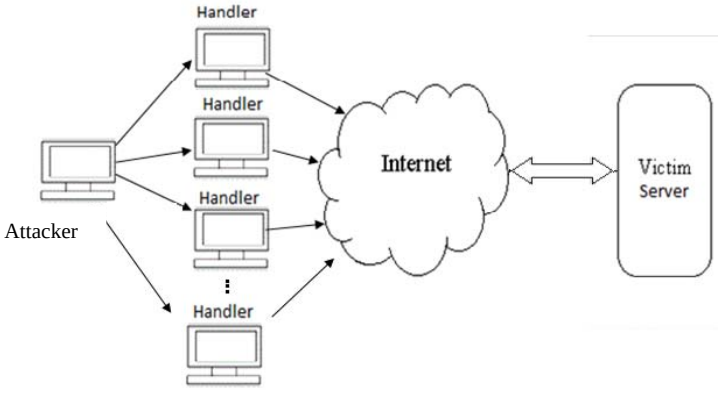

Fig1: A Typical DDoS attack in the internet

DDoS attack has become one of the major threat to the internet nowadays .A survey by Arbor Networks on 1st February 2011 has Revealed that the scale of DDoS attacks have been growing gradually since 2001 [02]. Arbor Networks' Sixth Annual Worldwide Infrastructure Security Report Reveals that DDoS Attack Size Breaks 100 Gbps for the First Time; while application layer attacks are on the top. In year 2010 many high profile attacks were launched against popular Internet services and other well known targets.

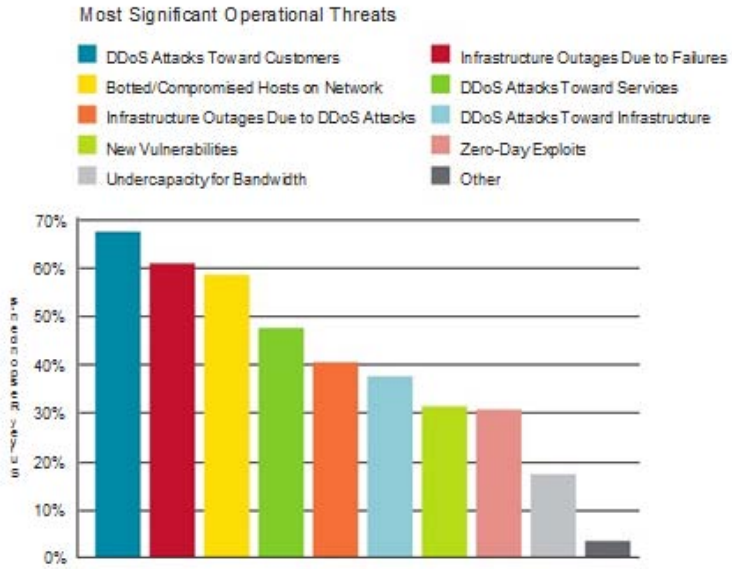

Fig 2: Most Significant Operational Threats observed during 12 months period.

Above Fig 2 illustrates that Sixty-eight percent of respondents indicated that DDoS attacks toward end customers were a significant operational threat encountered during this 12-month survey period. The scale and frequency of DDoS attack activity on the Internet was also high in this year. This represents that DDoS attack bandwidth has been increased by 102% since the previous survey period and which shows 1000 percent increase since Arbor released the first Worldwide Infrastructure Security Report (WISR) in 2005. Figure 1 illustrates the yearly reported maximum attack size of DDoS attack.

Compared to Transport layer DDoS, IDC and mobile/fixed wireless operators are reporting significant outages, increased OPEX, customer churn and revenue loss due to application-layer DDoS attacks. Various attack generation tools are available because of which, Application-Layer DDoS Attacks Are becoming more and more sophisticated with greater operational Impact, so that even a novice user can generate these attacks easily with the help of the available tools on the internet. DDoS attacks are targeting both their customers and their own ancillary supporting services, such as Web portals, DNS etc.

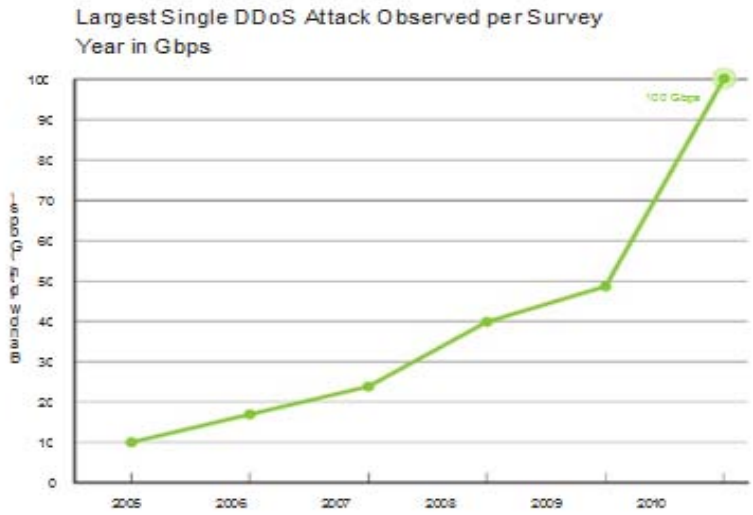

Fig.3: The yearly reported maximum attack size of the DDoS attack.

The report also says that as a low cost, high-profile form of cyber-protest, Botnet-driven DDoS attacks are likely to continue in 2011 and beyond. Major incidents of DDoS attacks in 2010 includes, attacks associated with

the territorial disputes between China and Japan, the ongoing political turmoil in Burma and Sri Lanka and the Wiki Leaks affair.

As new equipment, protocols and services are introduced into networks, the vulnerable attack surface of DDoS continues to expand. Application-layer DDoS attacks are going to be the most significant problems facing network operators in near future. Below is few of the defense mechanism against application layer DDoS attack introduced in technical papers.

II FEW RECENTLY IMPLEMENTED TECHNIQUES TO DEFEND AGAINST DISTRIBUTED DENIAL OF SERVICE ATTACKS

2.1 Protection from Distributed Denial of Service Attacks Using History-based IP Filtering [03]:

HIF technique maintains an IP address database (IAD) to store legitimate IP addresses of frequent users (addresses that appeared in the network in the last two weeks)[04]. HIF filter is activated only when there is a high network or server utilization that leads to dropping of packet. In which case it discards packets whose IP address does not appeared in the IAD.HIF makes use of hash techniques to build an efficient IP lookup table. Under normal traffic conditions Valid IP addresses (IP addresses that completes successful TCP handshake) of incoming traffic are updated in the IAD to learn new IP addresses. Whereas A TCP flow with fewer than 6 packets is considered to be anomalous i.e. result of network noise such a network scan, reflector attack traffic etc and hence source IP address of this type of TCP flow is not stored in the IAD. To remove expired IP addresses IAD makes use of sliding window with length 2 weeks.

2.2 Botz4Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds [05]:

In this paper Author suggests the design and implementation of Kill-Bots, a kernel extension to protect Web servers against DDoS attacks that masquerade as flash crowds. At any instant a Kill-bots web server is in either of two modes when the server is severely overloaded, Kill bots switches to SUSPECTED MODE and continues to operate in this mode until the load goes down to its normal range where it is again switched back to NORMAL MODE .In SUSPECTED MODE each new client connection must authenticate itself by solving a puzzle (CAPTCHA here) [06]. When the Kill-Bots server issues a puzzle, it creates a Token consisting of a 32-bit puzzle ID P, a 96-bit random number R, the 32-bit creation time C of the token, and a 32-bit collision-resistant hash of P, R, and C along with the server secret and embeds this token in the same HTML form as the puzzle. Puzzles in Kill-Bots expire 4 minutes after they have been served. When a user solves the puzzle, the browser reports the answer to the server along with the Kill-Bots token. The server first verifies the token by recomputing the hash, and ensures that the token was created no longer than 4 minutes ago. Next, if the answer to the puzzle is correct then that client is authenticated. Now, Kill-Bots server creates a cookie from the token by updating the token creation time and recording the token in the table of valid Kill-Bots cookies and issues this cookie to that client. Using this cookie the client is allowed to re-enter the system for a specific period of time (here 30 min). Also each correctly answered graphical test allows the client to execute a maximum of 8 simultaneous HTTP requests [07]. Subsequently, when a user issues a new TCP connection with an existing Kill- Bots cookie, the server validates the cookie by recomputing the hash and ensuring that the cookie has not expired, i.e., no more than 30 minutes have passed since cookie creation. The cookie table is used to keep track of the number of simultaneous HTTP requests that belong to each cookie. Connections that began before the server switched to the SUSPECTED ATTACK mode, continue to be served up to certain duration (05 min here) or until they terminate. Kill-Bots tracks how often a particular IP address has failed to solve a puzzle. It maintains a Bloom filter for this purpose [08, 09] and uses webs tone 2.5to distinguish legitimate users from zombies by their reaction to the graphical test rather than their ability to solve it[10]. Also kill bots uses Adaptive Admission Control to make the server function at the optimal admission probability [11, 12, 13, 14, and 15].

2.3 A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors [16] **:**

To model the web user behavior this technique makes the following assumptions : i)The results obtained by web mining shows that about 10 % of the WebPages of a website may draw 90% of the web access .ii)Web user behavior can be abstracted and profiled using the web page request sequence.iii)It is very difficult for an

attacker to build a routine that exactly mimics the legitimate user behavior. Author introduces an extended hidden semi-Markov model to describe the browsing behaviours of web surfers. Also a new on-line algorithm based on M-algorithm [17] is proposed in this scheme for the computation of normality. Here M-algorithm is used to find a path with distortion or likelihood metrics as good as possible. The entropy of an observation sequence can be computed by using forward variables based on [18]. If the deviation of average entropy from mean entropy is larger than a predefined threshold, the user is regarded as an abnormal one, and the request sequence will be discarded by the filter when the resource (e.g. memory and bandwidth) is scarce. When the given slot is timed out, on-line update is done by the self-adaptive algorithm proposed in [19]. to ensure the normality of training data and keep the model updated with changes in the normal users' behavior supervised method is used in building the initial model and the unsupervised method[20] is used to collect new training data to update the model.

2.4 A Deployable Architecture against Application-level DDoS Attacks [21]:

A protected web server has two IP addresses, IP_{SA}: IP anycast address and IP_{SO} : IP used to communicate with the overlay nodes. The architecture has three components: redirection, secure overlay and proxy networks. When a protected web server is under attacks, Redirection redirects the traffic to an overlay via IP anycast. Then the overlay routes the packets to the specific overlay nodes according to its secure routing protocol, at last The overlay nodes provide effective protection to the server by the distributed filter where the overlay nodes can discard all reflection flooding traffic by a simple match rule, such as ICMP and DNS reflection attacks, the distributed traffic control, and also by building a temporary Collaborative edge web cache. A collaborative edge web cache is built to distribute and replicate the servers' contents among the overlay nodes. In this way the whole overlay separates the protected web servers from their clients and acts as a firewall for the servers. And for this purpose it needs ISP's to configure the routers to meet the following: i) any client whose IP is IPC is unreachable to any overlay node. ii) The route between IPC and IPSO is unreachable. iii) The route between IPC and IPSA is reachable, and the packets whose destination is IPSA are routed to some overlay admission node by unicast routing protocols. When there is no attack or congestion, overlay transfers the destination address of packet from IPSA to IPSO and routes this packet directly to the protected web servers.

2.5 A Detection and Offense Mechanism to Defend Against Application Layer DDoS Attacks [22]:

In this paper layer-7 attack model is characterized into three parts: session flooding attacks, request flooding attacks and asymmetric attacks. The goal here is 1) maximize the normal users' service rates; and 2) minimize the normal users' delay for a service request. Author achieves this goal by combining a detecting and blocking method with currency method. Author introduces a DOW (Defense and Offense Wall) model to defend against DDoS Attack. The anomaly detection model is described as a three-phase work. First, normal client behavior profiles are built using K-means clustering method[23]; because clustering method just need normal data, which is easy to collect abundantly in real application this stage involves training, data collection, Normalization of datasets and finally clustering. second, attacks are detected by a cluster distance based method; where for each new request its signature vector is normalized and if the signature is located in the normal cluster then it is considered normal, otherwise, its trust value is inverse proportion to the distance to the nearest cluster *Si*, and direct proportion to *Si*'s session proportion and the number of clusters. a trust value is assigned to each new session , sessions are dropped by a filter based on the trust value of each session. Finally encouragement model, uses dropped sessions as inputs, and immediately asks each input's client to retry using the same session.

2.6 Mitigating application layer DDoS attack via effective trust management [24]:

In the internet Clients are used to represent both legitimate users and malicious attackers, to make the distinction a parameter called as trust is used in this technique. The client who behaves better in history will obtain higher degree of trust. Components of trust are as follows:

***Short-term trust Ts*:** is used to identify those clients who send session connection requests at a high rate when the server is under session flooding attacks.

Long-term trust Tl: is used to distinguish clients with normal visiting history and those with abnormal visiting history.

***Negative trust Tn**:* is used to penalise a client if he is less trustworthy than a new client. Less trustworthy means if the client's overall trust falls below the initial value T0.

Misusing trust Tm: is used to prevent vibrational attacks by repeatedly cheating for high trust.

Trust T*:* represents the overall trustworthiness of a client, which takes into account all of his short-term trust, long-term trust, negative trust and misusing trust. Short-term blacklist is maintained to record the list of clients whose trust T drops below some threshold, with an expiration time.

For each client some information is stored at either server side or client side, which is called as License. Here author suggests dispensing license to clients using cookies or by additional application layer protocols. A license serves two functionalities: user identification and trust computation. It trades off between a server's storage and computation resources. Each license contains: A 64-bit identifier ID, IP address of client, the overall trust T to the client, negative trust Tn, misusing trust Tm, last access time LT, average access interval AT, the total number of accesses AN and a keyed hash H of the concatenation of all the above, with a 128-bit server password SP (private to the server) as the key. Here each client is identified based on its public IP and the server assigned identifier which enables to uniquely identify a client even if he is hidden behind NATs.

The TMH mitigation mechanism is deployed at the server. For each session connection request TMH checks whether the client is blacklisted; if not, it computes the trust to the client and schedule the connection request for the server using trust--based scheduling.

2.7 CALD: surviving various Application layer DDoS attack that mimic flash crowds [25]: When a new HTTP Get request arrives, CALD computes the time difference between the two arrival times and update the average frequencies in the source and target vectors. Each target Webpage and source is identified with special serial number. A front-end sensor is used to detect the abrupt change in the traffic, and in which case it sends an ATTENTION signal to activate the DDoS attack detection component which makes advanced inspection for a decision. The DDoS traffic from the malicious IP addresses will be blocked and flash crowd continues. The traffic the server receives is a stream of successive HTTP Get requests. Traffic intensity measurement is taken at fixed, discrete time intervals from a time series {yt}.If there is intense change in the value of measurement then it means that there exists application-layer DDoS attacks or flash crowds. Therefore, at instant t, the difference between measured value yt and model output yt' represents the abnormal constituent of the traffic. For detecting abnormal traffic Kalman filter is used to calculate the difference between observed traffic value and past p traffic values [26,27]. The sensor counts the HTTP Get requests during the interval of 1 sec.

2.8 DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks [28, 29]: The defense model consists of a *DDoS-Shield integrated* into the reverse-proxy which schedules or drops attack requests before they reach the web-cluster tier. Each request belonging to every session is examined by DDoS- Shield and parsed to obtain the request type and to maintain the workload- and arrival-history of requests in the session. The system architecture for DDoS-Shield consists of: 1) Suspicion assignment mechanism that uses session history to assign a suspicion measure to every client session .and 2) DDoS-resilient scheduler that decides *which* sessions are allowed to forward requests and *when* depending on the scheduling policy and the scheduler service rate[30].

All tiers continuously monitor the resources in the tier and periodically generate resource utilization reports as well as overall system statistics at the application layer such as throughput and response time. The system is said to be under a resource attack when a surge in a resource's usage is accompanied by reduction in throughput and increase in response time without a DDoS attack detected at lower layers.

2.9 Defending Application DDoS with Constraint Random Request Attacks [31, 32]: This paper proposes a technique to defend against application-level DDoS attacks made at search engines. Any host or search engine has *n* users. S is a set of requests made by n users where k out of n users are attackers and Sa is a set of requests made by attackers. Sa is nontrivial and is a strict subset of S. If |S|=m and |Sa|=l, then l<<m.

Author assumes that the first L elements of S and Sa are identitical.ie Sa={e1,e2,...,el} belongs to S={e1,e2,...,el,el+1,...em}. For each user (ui) the host maintains the request history (*Hi*) that contains *r* most recent requests. Two assumptions are made in this paper: i) The request generation and service are considered to be in round-robin. ii) The random drawings of requests are uniformly distributed among the elements in the respective sets. If a request *ei* is drawn by an attacker, then the average number of elements before *ei* is drawn is *l-1*. The expected number becomes *m-1*, if *ei* is drawn by a legitimate user.

The algorithm is divided into three parts. In the first part, the size of the request history is adjusted and users with repeated requests are identified. These users are considered suspects. In the second part, an attacker from

all suspects is identified and all the elements in *Sa* from the attacker's requests are find out. Finally, *Sa* found in the second part is used to figure out all attackers.

CONCLUSION

Application layer DDoS attacks are immense threat to today's internet and the severity is increasing exponentially year by year for which various defense mechanisms have been proposed in the past. Having discussed above techniques we conclude that major challenge of DDoS attack is just to make the distinction between attacker and legitimate user to filter out the attack packets

Undoubtedly the techniques discussed above are extremely useful, a next step in this path would be to compare and evaluate all these various mechanisms by creating sets of data and an experimental testbed or to come up with a collaborative approach to find more efficient solution to defend against application layer DDoS attack.

REFERENCES

[1] 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.

[2] Arbor Networks, "Arbor Networks'' Sixth Annual Worldwide infrastructure security Report, February 2010.

[3] T. Peng, K. R. mohanarao, and C. Leckie, "Protection from distributed denial of service attacks using history-based IP filtering," in *Proc. IEEE Int. Conf. Communications*, May 2003, vol. 1, pp. 482–486.

[4] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites," in *Proceedings of the International World Wide Web Conference*, May 2002, pp. 252–262.

[5] S. Kandula, D. Katabi, M. Jacob, and A. W. Berger, Botz-4-Sale: Surviving organized DDoS attacks that mimic flash crowds Mass. Inst. Technol., Tech. Report TR-969, 2004 [Online]. Available: http://www. usenix.org/events/nsdi05/tech/ kandula/kandula.pdf.

[6] L. von Ahn et al. Captcha: Using Hard AI Problems for Security. In *EUROCRYPT*, 2003.

[7] H. Jamjoom and K. G. Shin. Persistent Dropping: An Efficient Control of Traffic. In *ACM SIGCOMM*, 2003.

[8] A. Broder and M. Mitzenmacher. Network Applications of Bloom Filters: A Survey. In *Allerton*, 2002.

[9] G. Banga et al. Resource Containers: A New Facility for Resource Management in Server Systems. In *OSDI*, 1999.

[10] Mindcraft Inc. Webstone - The Benchmark for Web Servers. http://www.mindcraft.com/webstone/.

[11] Netfilter/Iptables. http://www.netfilter.org.

[12] S. McCanne. The Berkeley Packet Filter Man page, May 1991. BPF distribution available at ftp://ftp.ee.lbl.gov.

[13] J. Mogul and K. K. Ramakrishnan. Eliminating Receive Livelock in an Interrupt-driven Kernel. In *USENIX Tech. Conf.*, 1996.

[14] R. Russell. Linux IP Chains-HOWTO. http://people.netfilter.org/rusty/ipchains/ HOWTO.html.

[15] S. Kandula et al. Botz-4-sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds. Technical Report TR-969, MIT., 2004.

[16] Yi Xie and Shun-Zheng Yu, "A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 17, NO. 1, FEBRUARY 2009

[17] J. B. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Trans. Commun.*, vol. COM-32, pp. 169–176, Feb. 1984.

[18] D. Dhyani, S. S. Bhowmick, and W.-K. Ng, "Modelling and predicting web page accesses using Markov processes," in *Proc. 14th Int. Workshop on the Database and Expert Systems Applications (DEXA'03)*, 2003, pp. 332–336.

[19] X.Yi and Y. Shunzheng, "A dynamic anomaly detection model for web user behavior based on HsMM," in *Proc. 10th Int. Conf. Computer Supported Cooperative Work in Design (CSCWD 2006)*, Nanjing, China, May 2006, vol. 2, pp. 811–816.

[20] M. Kantardzic, *Data Mining Concepts, Models, Methods And Algorithm*. New York: IEEE Press, 2002.

[21] Xiaolin Chen1,2, Hui Deng3, Feng Wang3, Mu Mu4, Sanglu Lu2 "A Deployable Architecture Against Application-level DDoS Attacks", The 9th International Conference for Young Computer Scientists.

[22] Jie Yu , Zhoujun Li, Huowang Chen , Xiaoming Chen "A Detection and Offense Mechanism to Defend Against Application Layer DDoS Attacks", Third International Conference on Networking and Services(ICNS'07) 0-7695-2858-9/07 $20.00 © 2007

[23] H.H. Bock. "Automatic Classification", Vandenhoeck and Ruprechat, 1974

[24] J. Yu C. Fang L. Lu Z. Li "Mitigating application layer DDoS attack via effective trust management" IET Commun., 2010, Vol. 4, Iss. 16, pp. 1952–1962.

[25] Sheng Wen, Weijia Jia, Wei Zhou "CALD: surviving various Application layer DDoS attack that mimic flash crowds" 2010 Fourth International Conference on Network and System Security

[26] B.Krishnamurthy and J.Wang. On Network-Aware Clustering of Web Clients, in Proceedings of the ACM SIGCOMM, Stockholm, Sweden, 2000.

[27] A.Broder and M.Mitzenmacher, Network Applications of Bloom 50Filters: A Survey. Internet Math, Volume 1, Number 4, 2003.

[28] Supranamaya Ranjan,, Ram Swaminathan, Mustafa Uysal, Antonio Nucci, and Edward Knightly,"DDoS-Shield: DDoS-Resilient Scheduling to CounterApplication Layer Attacks" IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 17, NO. 1, FEBRUARY 2009.

[29] TPC-W benchmark. Transaction Processing Council [Online]. Available:

[30] http://www.tpc.org

[31] M. Aron, D. Sanders, P. Druschel, and W. Zwaenepoel, "Scalable content-aware request distribution in cluster-based network servers," presented at the USENIX Annual Technical Conf., San Diego, CA, Jun.2000.

[32] Wei Yen and Ming-Fang Lee "Defending Application DDoS with Constraint Random Request Attacks", 2005 Asia-Pacific Conference on Communications, Perth, Western Australia, 3 - 5 October 2005.

[33] Rocky K. C. Chang, "Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial," *IEEE Communications Magazine*, pp. 42-51, Oct. 2002