$$\mu = \frac{\sum x}{n}$$

Where,
μ=Mean
x=samples
n=Average of samples

$$\sigma^2 = \frac{\sum (x-\mu)^2}{n}$$

σ i.e. Standard deviation is determined by square root of variance.

Global survival rate is determined by

$$SR_E = \frac{\varphi(BI_w, \mu BI, \sigma BI)}{\varphi(\mu BI + 2\sigma BI, \mu BI, \sigma BI)}$$

$\varphi(x, \mu BI, \sigma BI)$ is nothing but of Probability density function of N (μBI, σ^2BI).

Reshaping algorithm finds the discard number and randomly marks the requests to reject them. It discards the marked requests and survives the remaining requests to user.

C] Mathematical Model

1] Behavior index

$$BI_w \underline{def} P[\hat{X}^w_{1|T_w}, \vec{O}^w_{1|T_w} \mid \lambda]$$

Where,

T_w = Total number of requests

$\vec{O}^w_{1|T_w}$=Observation sequence of the w_{th} time window.

$\hat{X}^w_{1|T_w}$= the optimal hidden state sequence corresponding to $\vec{O}^w_{1|T_w}$

$$L_w = P[\vec{O}^w_{1|T_w} \mid \lambda]$$

denote the likelihood of $\vec{O}^w_{1|T_w}$ Fitting to the parameter set λ.

2] Structure Factor

$$SF^i_w = \frac{Num(i;w)}{\sum_{i=1}^{M} Num(i;w)}$$

Where,

Num (i, w) denotes the number of appeals generated by state i of the w^{th} time window $\sum_{i=1}^{M} Num(i;w) = T_w$ and i € IM.

$$\overline{SF^i} = \frac{1}{W}\sum_{w=1}^{w} SF^i_w, i \in IM$$

D] Expected Experimental Setup

The system is built using Java framework (version JDK 8) on Windows platform. The Netbeans (version 8.0.2) is used as a progress tool. The system doesn't require any specific hardware to run; any standard machine with operating system windows XP or above is capable of running the application.

4. Result and Discussion

Result Analysis

The proposed scheme is implemented on single machine and experiments are performed on dataset of Oracle log. Dataset of Oracle log is taken to generate the number of requests. The experiment includes following aspects:

1) The performance comparison between existing method and previous method.
2) The performance comparison between short term detection with Pre-filtering and short term detection without Pre-filtering.
3) The performance comparison between Long term detection with Pre-filtering and long term detection without Pre-filtering.

Analysis

Comparative analysis between existing system and previous system:-

As we discussed earlier; on detection of DDoS attack; existing explorations rejects the entire incoming requests (even if incoming traffic is an aggregation of normal as well as abnormal requests).Here it affects on the Quality of Service of legal user. So to solve this issue soft control mechanism was given in [1]. Proposed work implemented this soft control scheme in order to preserve the Quality of Service of legal user.

In this work there is need to be provide number of HTTP and HTTPs request as an input data to the system. Table 1 shows the record of denied requests by hard control system and by soft control system

Table 1: Number rejected request by Hard control scheme and Soft Control Scheme

Sr. No.	*Number of Requests*	*Number of Rejected Requests*	
		Existing System (Hard Control)	*Previous system and Proposed system (Soft Control)*
1	37	37	15
2	36	36	21
3	37	37	19
4	37	37	17

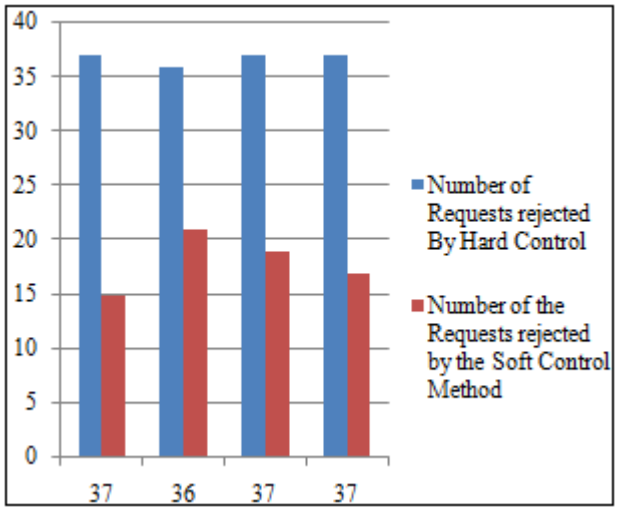

Figure 2: Comparison between Hard Control Mechanism and Soft control mechanism

Y-axis: Number of the request rejected by System
X-axis: Number of the requests

Figure2 depicts the comparison between Hard Control Mechanism and Soft Control mechanism. Blue bar shows number of requests rejected by hard control method and red bar shows number of requests rejected by soft control method. By graph we analyzed that soft control method has less rejected request than the hard control method which helps to improve the Quality of Service.

Evolution

We analyzed our proposed system performance for following cases

1) Processing time of short-term detection without pre-filtering
2) Processing time of short-term detection with pre-filtering.
3) Processing time of Long-term detection without pre-filtering
4) Processing time of Long-term with pre-filtering.

We have performed testing experiment for short term detection on 36 requests of 1sec firstly. At the time of initial execution there was no any previous record of blocking request. So for all new requests; system will do its task of finding normality or abnormality. During this diagnosis; system found abnormal requests on the basis of threshold value. After that it filters the requests by means of reshaping algorithm. System keeps the record of rejected requests and puts the number of blocked request into BlockIp list for future use.

Next time when requests arrive at system it makes the use of previous record of blocked list to check that whether newly arrived requests contains block list entries. In this checking system founds that among 36new requests 8 are matched with Blocked URL with IP. System has deleted these matched requests from new requests. So, now 28 requests are remaining into system for DDoS Detection. Obviously time require to process 37 requests will more than the time require to process 28 requests. From fig.3 we concluded that processing time of 36 is more than 28.

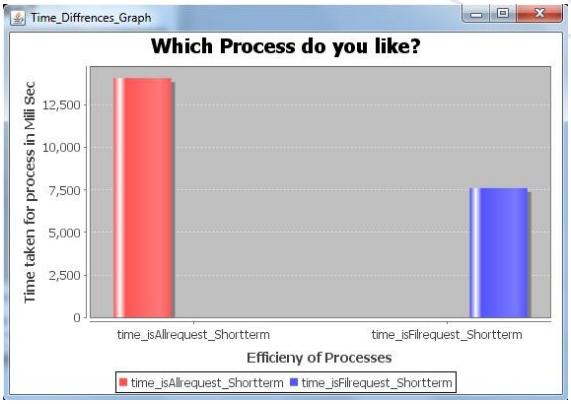

Figure 3: Efficiency of Process in Short term

Same experiment is performed for Long-term diagnosis "without pre-filtering" And "with pre-filtering" As it filters the request initially (if matches found) the time require to process whole incoming request is more than pre-filtered request. These results are depicted in fig 4.

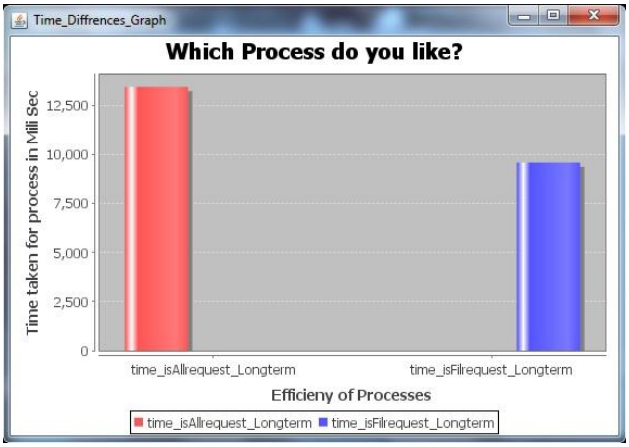

Figure 4: Efficiency of Process in Long-term

Considering the issue of Quality of Service, while implementing this system in real time there is need to delete the record of Block list after Sometime to achieve the basic requirement i.e. quality of service of legal user. Otherwise it may block the legal user requests for longer while. In our work we have assumed count when entries of block list exceeds than that count all records are deleted automatically. So system will take decision on the basis of new records.

Effectiveness of System

Fig.5 depicts the count of allowed requests and blocked request. X-axis denotes the total number of requests (Allowed requests as well as blocked requests) and Y-axis indicates the count of number of requests. From that we can concluded that proposed system is more effective than the system which was explored in [1].

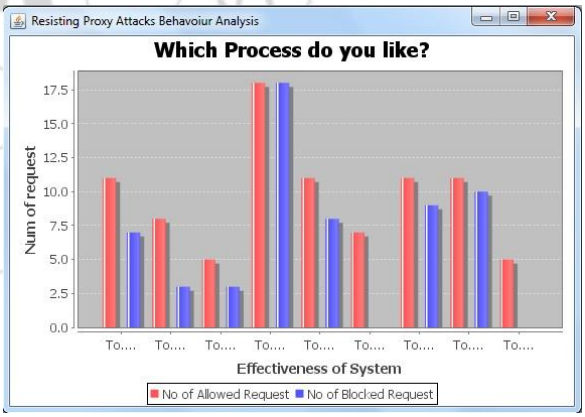

Figure 5: Effectiveness of System

5. Conclusion and Future Scope

Presented work is focused around recognition of DDoS attack and it ensures the origin server. Spatial Locality used to mine the access behavior of users which results into dynamic evolution of proxy-to-server traffic rather than static statistic. Temporal locality is unrelated to the frequency changing of documents. So, system is independent on the frequently varying web contents. Proposed system solves the aggregated traffic (i/e. mixed traffic of attack requests and normal requests) problem of DDoS attack by reshaping the

attack requests from the entire incoming traffic from proxy-to-server. To provide the Quality of Service to the typical client it utilizes soft control mechanism to handle the user demand. On detection of abnormal behavior; it reshapes the requests. On the basis of earlier record if discarded requests are found in newly arrived request at server then those requests are blocked by system at entry point. As it blocks the abnormal request initially; it requires less time to check normality or abnormality of remaining requests.

Finally we have concluded that scheme achieves its goals of providing the quality of Service to normal user and minimizing the time to process the requests (In case of early blocking). Proposed work is well suited for protecting the web server from DDoS attacks.

Proposed work focuses on attack detection, filtration and blocking (Pre-filtering). The reshaping algorithm which is used in our work for filtering discards the requests randomly. In worst case it may reject the requests of real user. Consequently this random rejection of request must be done on the basis of proper criteria. So it motivates us to do such exploration that will reject the request on the basis of particular analysis.

For to achieve better performance even more than proposed system, we have to prevent the happening of such attack instead of providing protection to the system after its happening. In future there is need to develop such mechanism that will prevent the happening of such attack.

References

[1] Yi Xie, Resisting Web Proxy-Based HTTP Attacks by Temporal and Spatial Locality Performance, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 7, JULY 2013.
[2] X. Jin et al., ZSBT: A novel algorithm for tracing DoS attackers in MANETs, EURASIP J. Wireless Commun. Netw., vol. 2006, no. 2, pp. 19, 2006.
[3] A. Shevtekar, K. Anantharam, and N. Ansari, Low rate TCP Denial-of- Amenity attack detection at edge routers, IEEE Commun. Lett., vol. 9, no. 4, pp. 363365, Apr. 2005.
[4] G. Carl et al., Denial-of-amenity attack-detection techniques, IEEE Internet Comput., vol. 10, no. 1, pp. 8289, Jan./Feb. 2006.
[5] P. Du and S. Abe, IP packet size entropy-based scheme for detection of DoS/DDoS attacks, IEICE Trans. Inf. Syst., vol. E91-D, no. 5, pp. 12741281, 2008.
[6] X Chen, J Heidemann, Flash crowd mitigation via adaptive admission rheostat based on application level observations. ACM Trans Internet Technol. 5(3), 532569 (2005). doi:10.1145/1084772.1084776.
[7] S Ranjan, R Swaminathan, M Uysal, A Nucci, E Knightly, DDoSshield: DDoSresilient scheduling to counter application layer attacks. IEEE/ACM Trans Netw. 17(1), 2639 (2009).
[8] J. Yuan and K. Mills, Monitoring the macroscopic effect of DDoS flooding attacks, IEEE Trans. Dependable and Secure Computing, vol. 2, no. 4, pp. 324335, Oct.-Dec. 2005.
[9] T. Peng and K. R. M. C. Leckie, Protection from distributed denial of amenity attacks using history based IP filtering, in Proc. IEEE Int. Conf. Commun., May 2003, vol. 1, pp. 482486.
[10] B. Xiao, W. Chen, Y. He, and E. H.-M. Sha, An active detecting method against SYN flooding attack, in Proc. 11th Int. Conf. Parallel Distrib. Syst., Jul. 2022, 2005, vol. 1, pp. 709715.
[11] J. Yu, C. Fang, L. Lu, and Z. Li, Mitigating Application Layer Distributed Denial of Amenity Attacks via Effective Trust Management, IET Comm., 2010.
[12] S. Lee, G. Kim, and S. Kim, Taxonomy-Order-Independent Net-effort Profiling for Detecting Application Layer DDoS Attacks, EURASIP J. Wireless Comm. and Networking, 2011.
[13] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang, Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient, IEEE Trans. Parallel and Distributed Systems, 2012.
[14] S. Yu, W. Zhou, R. Doss, and W. Jia, Traceback of DDoS Attacks Using Entropy Variations, IEEE Trans. Parallel and Distributed Systems, 2011.
[15] Y. Xie and S. Yu, A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Performances, IEEE/ACM Trans. Networking, 2009.
[16] Y. Xie and S. Yu, Monitoring the Application-Layer DDoS Attacks for Popular Websites, IEEE/ACM Trans. Networking, 2009.
[17] A. Chonka et al., Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks, J. Netw. Comput. Applicat.Jun. 23, 2010 [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2010.06.004