Volume 4, Issue 9, September 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Detection of Application Layer Ddos Attacks for Web Services Using Information Based Metrics

Mr. Nilesh A. Suryawanshi
Department of Computer Science
JSPM'S ICOER, Pune, India

Prof. Mr. S. R. Todmal
Department of Computer Science
JSPM'S ICOER, Pune, India

Abstract*—Distributed Denial of Service attacks is major threats these days over internet applications and web services. These attacks moving forward towards application layer to acquire and waste maximum CPU cycles. By requesting resources from web services in huge amount using rapid fire of requests, attacker automated programs utilizes all the capability of processing of single server application or distributed environment application. The phases of the scheme implementation are user behavior monitoring and detection. In first phase by gathering the information of user behavior and calculating individual user's trust score will take place and Entropy of the same user will be calculated. Based on first phase, in detection phase, variation in entropy will be observed and malicious users will be detected. Rate limiter is also introduced to stop or downgrade serving the malicious users.

Keywords- DDoS, Application Layer , Entropy, Trust Score, Rate Limiter

I. INTRODUCTION

DdoS attack saturates the server machine by over flooding the communication requests. These attacks are purposefully implemented by automating sending the requests for resources which web services provide. This consumes computing cycles, operating system data structures and bandwidth of network which usually leads to a server overload because of which desired services could not be served. The attackers first create the network of computers with automated clients which start requesting resources from servers which generates the huge volume of traffic which deny providing services to attackers and other users as well. The hosts running these attack tools are known as zombies, which are fully controlled by the attackers. Most of the existing techniques cannot discriminate the DDoS attacks from normal users. These attacks were handled and carried out at network layer when they were noticed. But these days it's very easy for attackers to target the application layers with automated clients. Application layer DDoS attacks are focusing to exhaust the server resources such as Sockets, CPU, memory, disk/database bandwidth, and I/O Bandwidth. They are very difficult to detect because they are similar to normal requests. DDos attacks on application layer are mostly of three types:

(1) Request flooding attacks in which huge number of requests appears in sessions.
(2) Session flooding attacks in which huge number of connecting sessions tries to take place.
(3) Session containing high workload appears.

Proposed scheme introduce facade layer between all client and the application servers which monitor the user's behavior (E.g. HTTP request rate, page viewing time and requested sequence of objects and their order) and detect the attacks as well. This facade layer can be another box sitting outside of the main application or facade can be implemented in same box in which the application resides. This facade monitors the users and maintains data about user's behavior, using that data it calculates the trust scores of the users. Based of the previous observed values and current observations the detection is made. Initially the entropy of incoming requests is calculated and compared with the allowable rate. If the deviation exceeds a threshold then that session is considered to be malicious. Otherwise the user or session is allowed to pass through the facade layer.

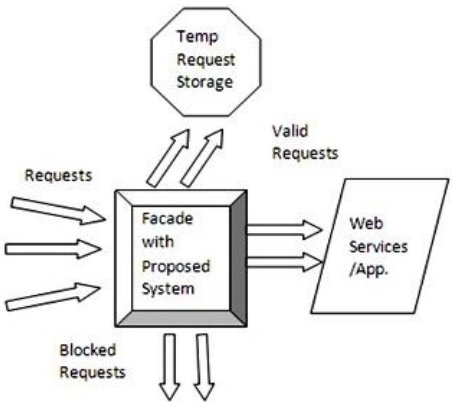

Fig 1: Request redirection

Figure 1 shows the redirection of the incoming requests. Every request for resources which web services serves will first come to the facade layer. In facade, the proposed system is sitting over the main application. Facade will decide what to do with the incoming requests.

II. RELATED WORK

Current system available tries to detect such attacks by analyzing the packet header, packet header arrival rate etc. They treat anomalies as deviations in the IP attributes, e.g., source IP address, TTL, and the combination of multiple attributes. Wang, Jin, and Shin [12], proposed a victim based solution where a received IP packet is discarded if major discrepancies exist between its hop count and the value stored in the previously built table. In StackPi [13], a packet is marked deterministically by routers along its path towards the destination. The victim can associate Stackpi marks with source IP addresses to detect source IP address spoofing. In Differential Packet Filtering against DDoS Flood Attacks [14], author relies on probabilistic means to determine risky packets. This scheme is adaptive to traffic change and attempts to sustain quality of service. Cabrera et al. [16] used the management information base (MIB) data which include parameters that indicate different packet and routing statistics from routers to achieve the early detection. Yuan and Mills [17] used the cross-correlation analysis to capture the traffic patterns and then to decide where and when a DDoS attack possibly arises. Keromytis, Misra and Rubenstein [15] present the conception of Secure Overlay Services (SOS) overlay network through which the legitimate traffic is sent. SOS network is able to change overlay topology dynamically to avoid DDoS and can survive in case that some key nodes are attacked. In [11], DDoS attacks were discovered by analyzing the TCP packet header against the well-defined rules and conditions and distinguished the difference between normal and abnormal traffic.

Liu and Chang [3] proposed a DAT (Defense against Tilt DoS attack) scheme. DAT monitors a user's features throughout a connection session to determine whether he is malicious user or not. For different behaved users, DAT provide differentiated services to them. Jie Yu [6] proposed a Trust Management Helmet scheme in which a user is assigned a license and a trust based on which detection is made.

III. SYSTEM MODEL

Old system was not capable to detect attack on application layer, comparatively new system can detect attack in new sub layer of application layer named "FACADE" layer.Attack detection was not available for web services which are made possible by new system. New developed system is detachable from main application working behind. Developed system can manage user sessions which makes main application completely service oriented.

The key features of the work are:

(1) Multiple checks happen over client.
(2) Main application can be kept safe and separate from attacks because of facade.
(3) False rejection rate is very low.
(4) The facade box is easily attachable and detachable to main application.

Comparing with the normal user requests, In DDoS the request rate increases significantly in very short time. The proposed system in facade layer has two phases, in first analyzing of the available data about the user and its characteristics takes place. Using that analysis a score is assigned to the to each user. Then the entropy of requests per session is calculated. Entropy is an information theoretical concept, which is a measure of randomness. The entropy is employed in this paper to measure changes of randomness of requests in a session for a given time interval. Then, based on the request history the user the trust score is assigned to the user.

In second phase, detection of the DDoS attack takes place. The entropy for the current session is calculated and degree of deviation with the predefined value is estimated. The amount of the deviation decides how suspicious the user is. Greater the degree of the deviation more the user is suspicious. The rate limiter, Scheduler and request blocker is also there in facade layer. Rate limiter sets proper thresholds and limits, based on which filtering is happen. Scheduler schedules the buffered requests on the basis if the server workload.Figure 2 shows the inside architecture of the facade layer. The detection mechanism is also deployed in facade. The authentication request comes first to the facade, and then the facades authenticate that user if valid credentials are provided by returning him the auth key. If user is requesting for authentication again still the same auth key is provided to him, this mechanism keeps him in same session and prevents him from making new session.

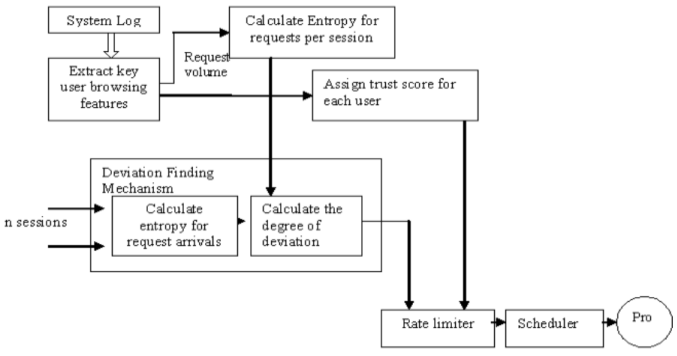

Fig. 2: System Architecture

If the deviation is within limit, then the rate limiter filters the session based on the trust score of the user. The client who behaves better in history will obtain higher degree of trust. If the user is considered legitimate, then the scheduler schedules the request based on the workload of the system. The highest trust score first policy is used to schedule the requests for the server.

A. Entropy Calculation

Let the request in a session be denoted as rij, where i, j I, a set of positive integers. 'i' denotes the request number in session 'j'. Let | (rj, t)| denote the number of requests per session j, at a given time t. Then,

$$|(r_j,t)| = \sum_{i=1}^{\infty} r_{ij} \quad (1)$$

For a given interval Δt, the variation in the number of requests per session j is given as follows;

$$N_j(r_j, t+\Delta t) = |(r_j, t+\Delta t)| - |(r_j, t)| \quad (2)$$

The probability of the requests per session j, is given by

$$P_j(r_j) = N_j(r_j, t+\Delta t) / \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} N_j(r_j, t+\Delta t) \quad (3)$$

Let R be the random variable of the number of requests per session during the interval Δt, therefore, the entropy of requests per session is given as

$$H_{new}(R) = -\sum_{i} P_j(r_j) \log P_j(r_j) \quad (4)$$

Based on the characteristics of entropy function, the upper and lower bound of the entropy H(R) is defined as

$0 \le H(R) \le \log N$ (5)

Where N is the number of the requests.

Under DoS attack, the number of request increases significantly and the following equation holds

$|H(R) - C| > \text{threshold}, t$ (6)

Where C is the maximum capacity of the session.

B. Rate Limiter

To avoid falsely detection, rate-limiter is introduced. Once the entropy is calculated, compute the degree of deviation from the predefined entropy. The system first sets a threshold for acceptable deviation. If the computed deviation exceeds the threshold, then the session is forced to terminate immediately. Otherwise, second level filter is applied by the rate limiter. The system also defines a threshold for validating a user based on the trust score. A user is considered to be legitimate only if the trust score exceeds the threshold. Otherwise, the user is considered malicious and the session is dropped immediately. The legitimate sessions are then passed to the scheduler for getting service from the server.

C. Scheduler

If the user is legitimate, then the scheduler schedules the session based on the lowest suspicion first (user with highest trust score) policy. The well-behaved users will have a little or no deviation. In such case, the legitimate user gets a quicker service. In addition to the scheduling policy, system workload is also considered before scheduling the request for getting service.

D. Monitoring Algorithm

Input: system log

1. Extract the request arrivals for all sessions, page viewing time and the sequence of requested objects for each user from the system log.

$$H_{new}(R) = -\sum_{i} P_j(r_j) \log P_j(r_j)$$

2. Compute the entropy of the requests per session using the formula:
3. Compute the trust score for each and every user based on their viewing time and accessing behavior.

E. Detection Algorithm

Input the predefined entropy of requests per session and the trust score for each user. Define the threshold related with the trust score (Tts) Define the threshold for allowable deviation (Td) for each session waiting for detection Extract the requests arrivals

Compute the entropy for each session using (4)

$$H_{new}(R) = -\sum_{j} P_j(r_j) \log P_j(r_j)$$

Compute the degree of deviation:

$$D = |H_{new}(R)| - |H(R)|$$

If the degree of deviation is less than the allowable threshold (Td), and user's trust score is greater than the threshold (Tts), then allow the session to get service from the web server Else

The session is malicious; drop it.

IV. PERFORMANCE EVALUATION

We present the performance of our system to Attack detection using entropy with trust score and gives result table.

Table 1 Result Analysis

Experiment	Attack exists	Users Involved	Attack Detected	Users Detected / Blocked	Time Taken(Minutes)
1	NO	4	No	0	2
2	Yes	1	Yes	1	1
3	Yes	3	Yes	3	2
4	Yes	12	Yes	10	5

Above table shows how attacker can try to enter system and how it will identify from system. In this system attacker can easily found on basis of trust score and it will block and drop it.

V. CONCLUSION

Developed application introduced efficient way to track DDoS attack over the REST web-services. New way uses pre available information metric for existing users and starts monitoring new users immediately as well. Every request has to pass the multiple checks to reach to its web-service destination. Authentication for the requests is managed by highly encrypted token service which is also part of proposed system. System also has a scheduler and rate limiter to downgrade the service to malicious user requests. Proposed system also has ability to block suspicious or malicious users. System provides workaround to traditional systems of DDoS detection and keeps trust level for individual user.

REFERENCES

[1] Shui Yu, Wanlei Zhou, Robin Doss, & Weijia Jia, (2011) *"Traceback of DDoS Attacks using Entropy Variations"*, IEEE Transactions on Parallel and Distributed Systems.

[2] Supranamaya Ranjan, Ram Swaminathan, Mustafa Uysal, Antonio Nucci, & Edward Knightly, (2009) "*DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer attacks*", IEEE/ACM Transactions on Networking, Vol. 17, No. 1.

[3] Huey-Ing Liu & Kuo-Chao Chang, (2011) "*Defending systems Against Tilt DDoS attacks*", 6th International Conference on Telecommunication Systems, Services, and Applications.

[4] Jin Wang, Xiaolong Yang & Keping Long, (2010) "*A New Relative Entropy Based App-DDoS Detection Method*", IEEE Symposium On Computers And Communications (Iscc).

[5] S. Yu, W. Zhou & R. Doss, (2008) "*Information theory based detection against network behavior mimicking DDoS attack*," IEEE Communications Letters, vol. 12, no. 4, pp. 319–321.

[6] Jie Yu, Chengfang Fang, Liming Lu & Zhoujun Li, (2009) "*A Lightweight Mechanism to Mitigate Application Layer DDoS Attacks*", in Proceedings of Infoscale'2009.

[7] G.Oikonomou & J.Mirkovic, (2009) "*Modeling human behavior for defense against flash-crowd attacks*", ICC2009.

[8] S.Kandula, D.Katabi, MJacob & A.W.Berger, (2005) "*Botz-4-sale: surviving organized DDoS attacks that mimic flash crowds*", in Proc. Second Symp. Networked Systems Design and Implementation (NSDI).

[9] J. Yu, Z. Li, H. Chen & X. Chen, (2007) "*A Detection and Defense Mechanism to Defend Against Application Layer DDoS Attacks*", in Proceedings of ICNS'07.

[10] Yi Xie & Shun-Zheng Yu, (2009) "*Monitoring the Application-Layer DDoS Attacks for Popular Websites"*, IEEE/ACM Transactions on Networking, Vol. 17, No. 1.

[11] L. Limwiwatkul & A. Rungsawangr, (2004) "*Distributed denial of service detection using TCP/IP header and traffi measurement analysis*," in Proc. Int. Symp. Commun. Inf. Technol., Sappoo, Japan,Oct. 26–29, pp. 605–610.

[12] Haining Wang, Cheng Jin & Kang G. Shin, (2007) "*Defense Against Spoofed IP Traffic Using Hop- Count Filtering*", IEEE Transactions on Networking,vol.15.No.1, pp.40-53.

[13] Perrig A., Song D, & Yaar A., (2003) "*StackPi: a new defense mechanism against IP spoofing and DDoS attacks"*, CMU technical report.

[14] Tanachaiwiwat, S. & Hwang, K., (2003) "*Differential packet filtering against DDoS flood attacks.*"ACM Conference on Computer and Communications Security (CCS).

[15] Keromytis, A.D., Misra, V., & Rubenstein, D., (2004) "*SOS: an architecture for mitigating DDoS attacks*", Selected Areas in Communications, IEEE Journal vol. 22, no. 1.

[16] J. B. D. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran & R. K. Mehra, (2001) "*Proactive detection of distributed denial of service attacks using MIB traffic variables a feasibility study*", in Proc. IEEE/IFIP Int. Symp. Integr. Netw. Manag., pp. 609–622.

[17] J. Yuan & K. Mills, (2005) "*Monitoring the macroscopic effect of DDoS flooding attacks,*" IEEE Trans. Dependable and Secure Computing, vol. 2, no. 4, pp. 324–335.

[18] W. Yen & M.-F. Lee, (2005) "*Defending application DDoS with constraint random request attacks,*" in Proc. Asia-Pacific Conf. Commun., Perth, Western Australia, pp. 620–624.

