

An Application Layer DDoS Real-Time Detection Method in Flash Crowd

Chuan Xu, Cheng Du⁺ and Xiangyue Kong

School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, 400065 Chongqing, China

Abstract. In recent years, real-time and effective DDoS attack detection system is desired since new application-layer-based DDoS attacks are more undetectable and frequent and the imperceptibility of the attack is increasing. There is therefore a need for a new effective detection for such new DDoS attacks. For the detection of application-layer-based DDoS attacks in Flash Crowd, this paper sets up user activity degree index and proposes a detection method based on Pearson correlation coefficient. The performance of the proposed real-time detection method is tested by the real data from network and has less misjudgment rate.

Keywords: application layer DDoS attack, attack detection, Pearson correlation coefficient, real-time.

1. Introduction

One of the difficulties in the application layer DDoS attack detection is the attack's concealment. Recent researches [1][2] show that attackers choose the timing to launch attacks. As a large number of normal users pay attention to certain information, the web request rate may increase dramatically (called flash crowd). When the flash crowd happens, the server would burden a large workload. If the attackers select this moment to launch the attacks, they may easily reach the attack purposes. In this case, attack detection algorithm may encounter great challenges: 1) the attack request rate is viewed the same as flash crowd's request rate, and it is hard to discriminate between attack and flash crowd; 2) the high traffic property of flash crowd makes web server have to receive many users' requests, so the detection algorithm must be able to detect the DDoS attack in time.

In view of the above problems, Jaeyeon J. et al.[1] had tried to distinguish Flash Crowd and DDoS attack through features of the traffic, users and file access. However Sheng W. [3] point out that this features were not the essence of application layer DDoS attack, and attackers could still escape detection through the small change. Oikonomou G. et al. [4] build the user behaviour model through investigation of user requests time interval and requests transition probability, but it is hard to finish for a large number of dynamic pages in a web site. Yi Xie et al. [5][6] researched the HTTP attack during flash crowd event. They proposed a detector utilizing HsMM to describe the dynamics of Access Matrix abstracted by PCA and ICA. But, it is high complexity for HsMM being multiple parameters resulting difficult training. These methods are mainly based on the user behaviour for improving the attack detection accuracy, less regarding the real-time performance of the algorithm.

In this paper, we propose an application layer detection algorithm based on Pearson correlation coefficient (PCC). This proposed algorithm is an effective and real-time method to discriminate requests among DDoS attack sources and normal users by using user activity degree and Pearson correlation coefficients. The contributions of the paper are listed as follows:

- We proposed the user activity degree index to analyze the characteristics of user browsing behavior, and established the browsing behavior model by using the Pearson correlation coefficient. Through

⁺ Corresponding author. Tel.: +86 236246 0934; fax: +86 236246 0934.
E-mail address: duchengcc@163.com.

observing real dataset, we found the active degree of normal users, flash crowd and attackers.

- We pointed out a method to estimate the user activity degree utilizing the number of user requests. The experiment shows that correlation coefficient of user activity degree could evidently discriminate between attackers and normal users in flash crowd.
- In order to realize the real-time detection in the high traffic environment, we pointed a Pearson correlation coefficient based application layer DDoS real-time detection method in flash crowd.

The rest of the paper is organized as follows. Section 2 presents the details of user activity degree model. In Section 3, a state detection algorithm based on Pearson correlation coefficient is proposed. The performance evaluation of our method is shown in Section 4 by conducting experiments with the real datasets. In the final Section, we provide the summary of our works.

2. User Activity Degree Model

2.1. Analyzing characteristics of user activity degree

The related researches [6][7][8] of DDoS attack have shown that there are several features of application-layer DDoS attack targeting web server:

- (1) The attack is launched by many real or fake IPs;
- (2) These IPs would send many HTTP-GET requests to the web servers;
- (3) Servers would encounter a large amount of requests in the short time.

In order to distinguish attacker from normal users, we propose User activity degree and Pearson correlation coefficient to describe user browsing situation for the duration. We define User activity degree as:

$$UA_i(t) = UF(IPC(t, ip_i), r) \quad (1)$$

Where IP is the ID of a user, UF is the evaluation to $UA_i(t)$ and $NO.i$ user's number of HTTP requests is $IPC(t, ip_i)$ which ranked r in t th duration Δt . It is obvious to know $UA_i(t) \propto IPC(t, ip_i)$, $UA_i(t) \propto 1/r$. So, the User activity degree of a user is influenced by not only the number of requests of this user, but also others'.

Because of the difference between behaviour patterns of the normal user and the attacker (as shown in Fig. 1), User activity degree can discriminate the request rates between them [9], particularly, under the flash crowd background. After sending a series of requests for a web page, a normal user would stop for browsing the page for a while time when he is 'quiet' to server until click other page. So it is obvious to understand that normal user's User activity degree is quite different. UA value of this user would turn up at duration t_k for the increasing of $IPC(t, ip_i)$ and $1/r$. Yet the User activity degree would turn down for user's lacking of action. On the contrary, when attackers triggered an attack, they would not stop sending requests for the purpose of denying service of servers. $IPC(t, ip_i)$ and $1/r$ would stay at high level, UA value would not turn down during the attacking time. In Fig. 2, it is shown the different features of UA curve of normal user and attacker. UA curve of a normal user shows volatile while attacker's UA curve is stable.

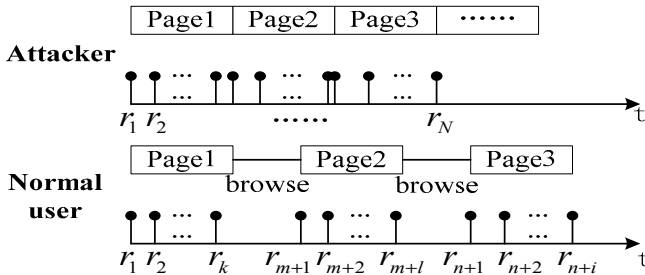


Fig. 1: Behaviour patterns of normal user and attacker

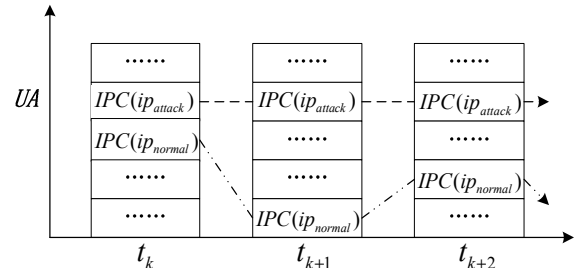


Fig. 2: UA curve of normal user and attacker

2.2. Modeling of User activity degree with Pearson correlation coefficient

Based on the above analysis, the difference between the normal user and the attacker is the similarity of UA value at times which we propose Pearson correlation coefficient to measure defined as:

$$PCC_{XY} = \frac{\text{cov}(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}} \quad (2)$$

Where $\text{cov}(X, Y)$ is the covariance of X and Y . $D(X)$, $D(Y)$ is the variance of X and Y respectively.

In our case, PCC value is used to measure the similarity of UA values at duration t_k and t_{k+1} , can be rewritten as:

$$PCC_{UA}(t_k, t_{k+1}) = \frac{\text{cov}(UA(t_k), UA(t_{k+1}))}{\sqrt{D(UA(t_k))}\sqrt{D(UA(t_{k+1}))}} \quad (3)$$

According to equation (1), we know that $IPC(t)$ is the main factor of UA value. So it's reasonable to take $IPC(t)$ as the evaluation of UA value for which equation (3) can be rewritten as:

$$PCC_{IPC}(t_k, t_{k+1}) = \frac{\text{cov}(IPC(t_k), IPC(t_{k+1}))}{\sqrt{D(IPC(t_k))}\sqrt{D(IPC(t_{k+1}))}} \quad (4)$$

It could be easily inferred from equation (1) and (2) that $PCC \in [-1, 1]$ and $|PCC|$ value to be the closer to 1, the stronger relativity of X and Y be; the closer to 0, the weaker relativity of X and Y to be. In this paper, we use PCC of a random process $X(t)$ to reflect its change upon duration t_k and t_{k+1} . We can make good use of this mathematical feature in attack detection by calculating PCC value. When PCC value is higher than preset threshold δ , the attack happens, or else there is no attack. As mentioned above, PCC is the key factor for us to construct our attack detection algorithm.

3. Pearson Correlation Coefficient Based Detection Algorithm

The proposed algorithm employs the feature of User activity degree of attackers and normal users. To achieve the better detection efficiency, our algorithm will make a pre-treatment to the statistical data in durations to describe which we have such definitions,

- (1) X and Y , the vector that presents the status of a web site in t_k and t_{k+1} respectively.
- (2) $IP(t)_r$, the ID of the user whose number of HTTP requests ranks r in t among users.
- (3) $IPC(t)_r$, the number of HTTP requests ranks r among numbers.
- (4) $IPC(t, ip_i)$, the number of HTTP requests of the user whose ID is ip_i in duration t .
- (5) $IPC'(t_{k+1})_{r,s}$, the number of HTTP requests of the $IP(t_k)_r$ user which ranks r in t_{k+1} , while that of this user ranks s in t_k .

Our framework is using $PCC_{IPC}(t_k, t_{k+1})$ to evaluate $PCC_{UA}(t_k, t_{k+1})$. As shown in Fig. 3, we propose a method to construct X and Y for this task.

Construction Method: As collecting the workloads of a web site in Δt periods, the descending ranking of the numbers of HTTP requests of users $IPC(t)_r$ can be easily obtained. For Mark Meiss et al. point out those numbers' Zipfs distribution, which states that the small part of users causes most workload of web server. We extract the set $\{IPC(t_k)_r, |1 < r < R\}$ from the sorted numbers $IPC(t_k)_r$ as vector X . When we obtain $IPC'(t_{k+1})_{r,s}$ at t_{k+1} , Y vector would be constructed with the set $\{IPC'(t_{k+1})_{r,s}, |1 < r < R\}$ from sorted $IPC'(t_{k+1})_{r,s}$ where $IPC'(t_{k+1})_{r,s} = 0$ if $k > R$ for the purpose reflecting the request number's influence User Activity Degree.

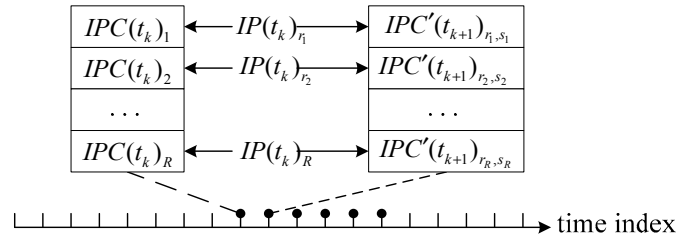


Fig. 3: X and Y construction method

With **Construction Method** for X and Y , the description to our algorithm as below:

- (1) Initiation: set the value of threshold δ obtained by pre-approval, Δt determining how long is the period for statistics; initial variable t flagging the beginning of the period, the hash table HT which indexes every user's ID to his record for request number.

- (2) Recording to request number of every user: it is the process that handles every request arrived whose time stamp $request.t$ is used to judge if exceeds period Δt associating variable t and source IP $request.source_ip$ is the index to hash table HT in which this user's record pluses 1.
- (3) Sorting the request number of users: when $request.t - t$ exceeds period t , the request number of users would be sorted whose result is necessary for construction to X and Y .
- (4) Calculating PCC value: obtaining the request number of users in period Δt stored in HT, PCC value can be calculated after constructing X and Y with **Construction Method** if which higher than threshold the alert would be pulled.
- (5) Repeating step (1), (2), (3) and (4).

4. Experiment Results and Performance Analyses

The experiments in this paper use two data sets. The first data was acquired by monitoring the users who visited www.sina.com in June 1, 2010 in Chengdu telecom IDC room with the technologies of router mirroring and network packet capturing. The other data set is the official 98 World Cup website access log. This data set recognized as a Flash Crowd phenomenon is authoritative in the application layer network security. This paper uses the website visit records on 8 July 1998, when a 1/2 final was held. We demonstrate the universal applicability of the algorithm in this paper with these data sets, one for determining threshold value, the other for testing the algorithm efficiency.

4.1. Threshold determining

The first data set is used to determine the suitable threshold value. In this statistics process, the factors: time interval Δt , X and Y length R , will all affect the PCC value. Synthesizing factors of physical properties reflected by PCC value and the algorithm real time, the time interval is set as 5 minutes, and R is 50. The PCC value of the data set can be calculated with algorithm whose probability histogram shown in figure 4.

In Fig. 4, $PCC_{IPC}(t_k, t_{k+1})$ mainly concentrate in area of $[-0.2, 0.85]$, which illustrates the activity degree is limited. Further, by analyzing Fig. 4, this paper set the $PCC_{IPC}(t_k, t_{k+1})$ abnormal threshold as 0.9. If the $PCC_{IPC}(t_k, t_{k+1})$ exceeds 0.9, the attack is judged as happened.

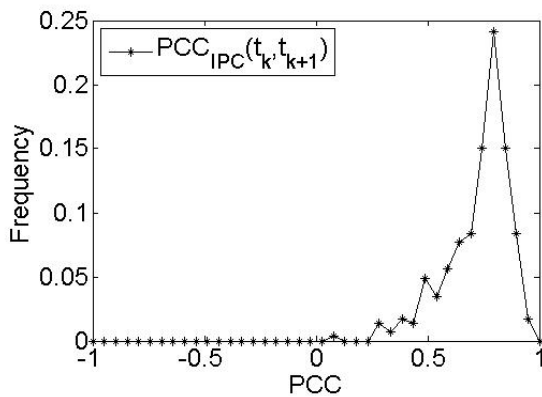


Fig. 4: Probability histogram of PCC value

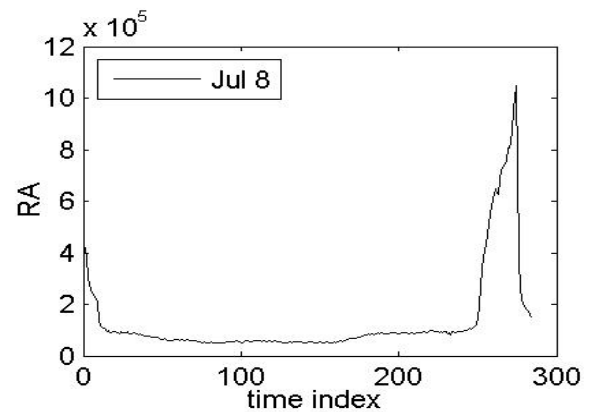


Fig. 5: RA curve on July 8

4.2. PCC performance testing

This paper utilized the second data set to test the PCC performance. This RA (Request Arrival) curve in the data set is shown in Fig. 5. The time interval Δt is set as 5 minutes. In this figure it can be seen that the site flow in most time of the whole day is steady, between the 250th and 270th Δt periods, web flow continued to rise to the peak in this day. However, all the flow is generated by the normal users. The phenomenon like this that most flow is concentrated in a period of the whole day is called Flash Crowd.

We further add attack data in this data set to test the performance of the proposed method. The experiment is realized by Perl script, which concludes powerful string processing functions. We utilizes mathematics library contained by Perl to simulate attack. The 800 attack IPs are randomly generated in the attack simulation. The IP attack requests number in a time interval Δt obeys normal distribution, whose mean value is 2000 and variance is 5. In order to investigate the performance of suggested detection method, the experiments add application layer DDoS attack in Flash Crowd.

In this data set, Flash Crowd phenomenon occurs between 250th and 280th Δt . This paper adds the attack data in flash crowd, and from 255th Δt , the time lasted about 10 Δt .

After attack data added, $PCC_{IPC}(t_k, t_{k+1})$ curve is showed in Fig. 6, the dotted line is the threshold. It can be seen that, during the attack period, PCC value exceeded the threshold set in advance. This illustrates that, aiming at attacks added in normal data set, the suggested method can detect attacks efficiently by setting the right threshold value. This figure proves the feasibility of the proposed method, which can detect attacks hidden in Flash Crowd.

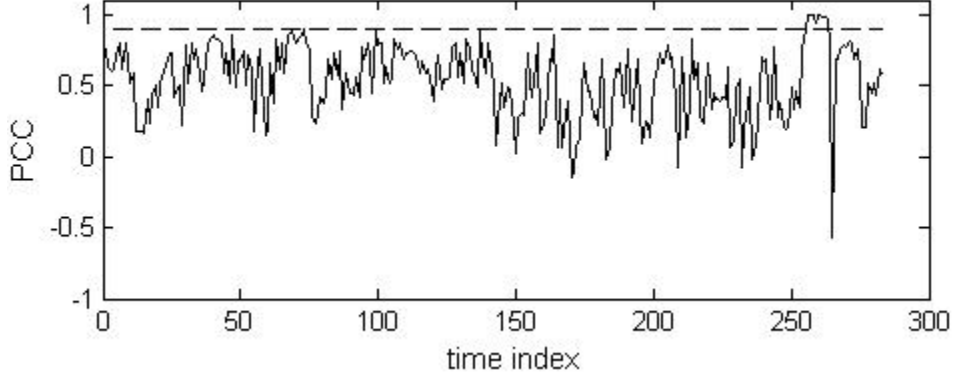


Fig. 6: $PCC_{IPC}(t_k, t_{k+1})$ curve with attack data added in this experiment

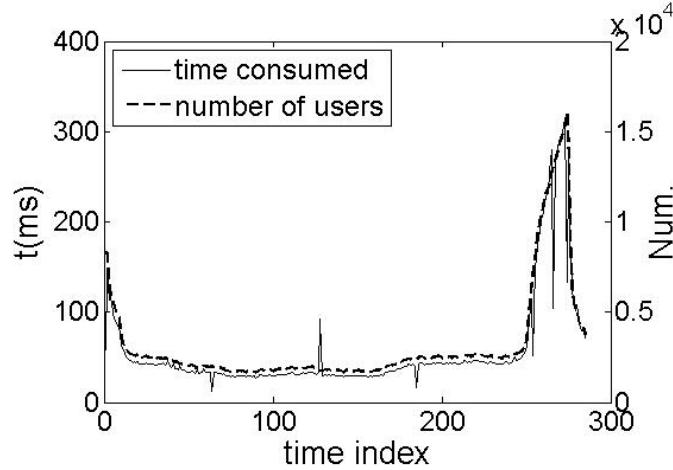


Fig. 7: Detection time consumed and the number of users

4.3. Algorithm efficiency compare

Basing on the time interval statistics, the proposed method is can quickly detect abnormal behaviour by judging PCC whether exceeds the threshold. In order to manifest the low complexity and real-time property in the proposed method, this paper performed experiments to contrast the proposed method with the one detection algorithm of literature [5] application layer DDoS attack under Flash Crowd background.

For Literature [5]'s using the same data set as this paper, it is convenient to contrast the two methods. Literature [5] has to experience the algorithm train period in which the algorithm complexity is about $O((MD + M^2)TI)$. In the simulation experiment, $M = 9, D = 60, T = 1440, I = 10$. The run time algorithm complexity is about $O(NKT + K^2T + (MD + M^2)T)$ where $N = 3000$ with 3×10^5 multiplications. The algorithm needs 10 minutes at the train stage and 500 ms in once detection.

Analyzing the proposed method, we can find the detection algorithm complexity mainly focuses on the ranking of user request, namely ranking algorithm. Generally the complexity in the ranking algorithm is related to the size of proceed set, namely $O(N) \sim O(N^2)$, while N represents users number in Δt period.

To investigate time consuming, this paper and had similar hardware configuration with the literature [5] for experiments. The Fig. 7 shows each time consumed of this algorithm processing data in literature [10]. Here this paper put the number of subscribers each Δt period in this figure. It can be seen that each time

consumed in detection algorithm is related to the number of handled users: the more subscribers, the more spending time. Each running time is about 56ms.

Table. 1: Performance Comparison Result

	Time complexity	Time Consumed
Algorithm of literature [5]	$O(NKT + K^2T + (MD + M^2)T)$	500ms
Algorithm of this paper	$O(N) \sim O(N^2)$	56ms

As is shown in table 1, the time complexity and time consumed in the proposed algorithm are both superior than the method proposed in the literature [5].

5. Conclusion

The characteristics of application layer DDoS attack is described in this paper with the User active degree and Pearson correlation coefficient. First, we estimate User active degree with the number of user requests; then, use Pearson correlation coefficient to measure their variation in times. In the experiment, this proposed algorithm is applied to detect the attack under Flash Crowd as proved being higher efficiency of real-time in contrast experiment.

6. Acknowledgements

This work has been partially supported by the National Natural Science Foundation of China under the grants (61040044 and 60873079), and the Natural Science Foundation of Chongqing under the grant (CSTC.2009BA2089).

7. References

- [1] Jaeyeon J, Balachander K, Michael R. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites[C]. WWW 2002, pp. 293-304.
- [2] Chu-Hsing Lin, Chen-Yu Lee, Jung-Chun Liu. A detection scheme for flooding attack on application layer based on semantic concept[J]. Computer Symposium (ICS), 2010, pp. 385-389.
- [3] Sheng Wen, Weijia Jia, Wei Zhou. CALD: Surviving Various Application-Layer DDoS Attacks That Mimic Flash Crowd[C]. Network and System Security (NSS), 2010, pp. 247-254.
- [4] Oikonomou, G., Mirkovic, J. Modeling Human Behavior for Defense Against Flash-Crowd Attacks[C]. IEEE International Conference, 2009, pp. 1-6.
- [5] Yi Xie, Key technology of HTTP application-layer attack detection research [M]. Zhongshan University, 2008.
- [6] Yi Xie, Shun-Zheng Yu. Monitoring the Application-Layer DDoS Attacks for Popular Websites[C]. IEEE/ACM Transactions, 2009, pp. 15-25.
- [7] Yi Xie, Shun-Zheng Yu. A Novel Model for Detecting Application Layer DDoS Attacks[J]. Computer and Computational Sciences, 2006, pp. 56-63.
- [8] Thapngam, T., Shui Yu, Wanlei Zhou, et al, Discriminating DDoS Attack Traffic from Flash Crowd through Packet Arrival Patterns. INFOCOM WKSHP, 2011, pp. 952-957.
- [9] Yi Xie, Shun-Zheng Yu. A Large-Scale Hidden Semi-Markov model for Anomaly Detection on User Browsing Behaviors[C]. Networking, IEEE/ACM Transactions on, 2009, pp. 54-65.
- [10] <http://ita.ee.lbl.gov/html/contrib/WorldCup.html>.