Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(3):1342-1350

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

APP-DDOS defense model based on the score strategy

Yulian Hu

Foreign Language Department, Heze University, Heze, China

ABSTRACT

While Distributed Denial of Service Attack in Application layer (APP-DDOS) is popular, it is also becomes more difficult for people to defend it. Based on the current defense technology, this paper presents a new defense model on the basis of "score strategy", which can identify the user through IP, use the score strategy to dynamically adjust the user's scores, and respond to users' service request according to score value. The model not only ensures the normal users' effective access to the maximum, but also shields abnormal requests quickly. The relevant experiment proves that the model has a good defense effect on HTTP protocol attacks.

Keywords: Distributed Denial of Service Attack; application layer; score strategy

INTRODUCTION

With the improvement of science and technology, computer network has become increasingly necessary in people's daily life. Naturally, network safety attracts people's attention, among which the distributed denial of service (DDOS) is one of the hottest topics. In application services, most protocols have potential security problems. Just by a small number of requests can the client greatly consume the server's resources，which is called APP-DDOS. “MyDoom” virus flooded in 2004 was a typical case of APP-DDOS attack. The main reasons include that the tools used to start DDOS attack are very easy to operate and access, internet itself has some major defects in its original design, and also the pursuit of economic profit caused by human greed. Thus the research on APP-DDOS has great theoretical meaning and practical value.

RELATED PRESENTATIONS

2.1 *APP-DDOS attack theory*

DDOS is the full name of distributed denial of service attack, which means that the attackers have the system resources of the host greatly occupied or have its network bandwidth jammed and finally make valid users have no services. Different from the traditional DDOS attack, APP-DDOS usually sends legal request in a way of normal access so as to take up large service resources and finally makes legal users fail to access. Its typical attacks include CC attack, HTTP storm attack and so on. When starting attacks, the attackers constantly demand to calculate the pages with more spending such as query a database and download a file. Those which provide services are usually ordinary servers, which are easily to be made paralyzed if there are enough queries.

2.2. *Commonly used APP-DDOS defense methods*

2.2.1. Methods based on tests

Both reference [5] and [6] adopt Turing test which is very effective because the computer does not have logical calculation ability. Reference [6] compiles a kind of behavior detection procedure and uploads it to the Client. It can distinguish between a normal user and an attacker by detecting the normal user's behavior such as mouse movement; or by analyzing users' request and judging if it is the behavior of normal browsing. But the Turing tests usually interfere with visitors' normal access to the server and impact on their "user experience". Literature [7] puts forward a method called speak-up to resist APP-DDOS. Different from previous filtering methods which reduce the attack traffic or weaken the attacker's behavior ability, speak-up can make all the clients improve sending rate. To get a better attacking result, attackers often start attack with great effort and may neglect the control of sending rate from the attacking target. So those who can improve sending rate are legal users. In this way people can tell the legal access from attacking behavior. But the fault of this proposal is that the server cannot effectively inhibit attacks in a short time, because the server may have already collapsed before recognizing the legal users.

Literature [8] puts forward a defense method of balancing the client and the server. This method can instantly recognize attackers by Turing test when some abnormal behaviors are detected. When the client has not reached the threshold value, it can be allowed to have some limited access. The key of this proposal is that it uses Turing test which has influenced users' experience to some extent.

Literature [9] comes up with a DOW defense mode, which detects the users by reducing the request rate of attack and workload rate, and makes normal users access more service by increasing the session rate of normal clients in the way of flow measurement. The core of this defense method is still similar to speak-up in literature [7], which still cannot quickly inhibit attacks.

2.2.2. Methods based on behavior model

Literature [10] points out the method of the payment channel which is a module that specially records the byte streams. It asks the users who cannot get service to keep request retry and regard the constant retry as a byte- stream. A payment channel is set up, and all clients who have submitted applications are asked to send byte-streams. If malicious nodes submit multiple applications at the same time, several payment channels will be opened up, each of which is independent, and some of which can be collapsed because it unable to send so much flow. This proposal requires the servers to set up so many payment channels that they may bring great burden to the server, which will be collapsed if there are too many payment channels.

Literature[11] analyzes the difference between APP-DDOS attack and normal access behavior, putting forward attributes of abnormal access behavior and abnormal degree model of session. This model can effectively distinguish normal access session and APP-DDOS session so that it can filter attack flow. But such proposal increases calculation amount of the server and can postpone the forwarding requests.

With the reasonable analysis of the proposals above mentioned, this paper proposes a new APP-DDOS defense method---APP-DDOS defense model based on score strategy.

MODEL BUILDING

3.1. *Conditions hypothesis*

Considering the detailed attack situation of APP-DDOS, the paper bases on the following assumed conditions:
(1)The attacker obtains a large number of IP through relevant means to access target server, and each attack IP is real and interactive.

(2)In order to achieve the purpose of denial service, attackers often adopt high frequency attacks, that is, they visit some webpage or use some service with frequencies which are far greater than normal users'.

(3)In this model, the detection algorithm in the intrusion detection host can detect the attack before attack flow paralyzes the target server.

3.2. *Some relevant definitions*

In the DDOS attack of network layer, the more the attack packets, the more energy the server consumes. But in APP-DDOS, a large number of asymmetric requests break the law, so this paper gives some definitions to be measurements.

(1) Service price: The value stands for the score value paid by the users when they use some service or access a page. Each service price Ｘi= Ｃi•Ｑi, Ci stands for gains weight of each service or page, Ｑi stands for the resources costs needed by service. The value integrates CPU computation spending, memory, band width, and I/O speed of each service or page.

(2) Score: The value is the consumables used to obtain some service or access a page. If the service or page is regarded as a commodity, the score is the currency to buy the commodity. The amount of the value is also an important symbol to measure whether a user is normal or not.

3.3 Core concepts of this model

This model is based on the characteristic of application DDOS, whose core concepts are as the following:
(1) Compared with the machine access, normal users can more flexibly access to the website, can effectively handle various emergencies and have logical operations. This is the most striking features. On the contrary, the machine access is just to carry out a series of programming instructions without flexibility in emergencies, logical operations and judgment.

(2)When starting attacks, there is a large mount of IP that participate in the attack. When meeting the unexpected situations, the attackers are not likely to interact with each IP and send every detailed construction.

(3)According to the basic assumptions, once the attackers start attacks, they often use high frequency attacks. They adopt high frequencies that are greater than normal users' to access some webpage or get some service. So the scores of the attackers can be quickly consumed. When their scores are lower than a threshold value, their requests will be frozen and the result of inhibition attack is achieved.

3.4 Model designing

The design diagram of this model is showed in figure 1. The frame of the whole system can be clearly seen from the figure.

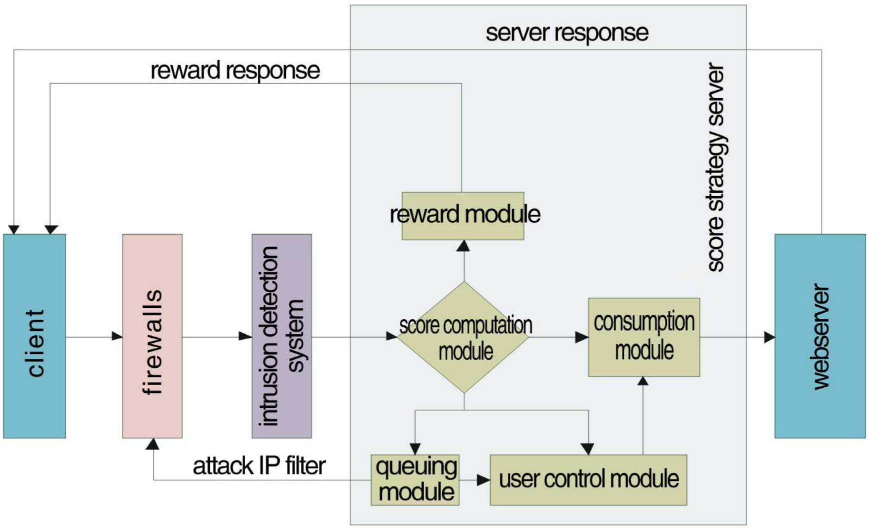

Figure1: Score strategy defense model

3.4.1 Firewalls

A firewall is the first defense line of the model which plays a very important role. Besides defending some existing and common attacks, firewalls can packet filter the blacklist IP maintained in queuing module inside this model. They also have the function of log records which can record the data of network access, which are very helpful for analyzing network attacks. Firewalls can only defend part of attacks, but they firstly filter the flow which enters the defense model, and reduce part of attacks to the server.

3.4.2 Intrusion Detection System
Intrusion Detection System is the second defense line in the whole defense system, the supplement of firewalls and one of the key factors to decide the quality of the system. The rule of this system matches the procedures, which can detect the DDoS attack with existed rules and reduce the burden of the server.

3.4.3 Score calculation module
This module is the core of the whole system, which is responsible for maintaining a user list such as the calculation of the user's score accumulation and deduction, and judges if the IP enters the queuing module or not. The detailed algorithm is as the following:

If the request IP is not in the whitelists of the module controlled by users, this IP will be affirmed as a new user. As for a new user and new IP, the initialized score value of each IP is α (this value is enough for a normal user to finish accessing or enjoying the service in enough long time, but it is not enough for an attack IP with high frequency access); Every time when a service or an access to a page is requested, the corresponding score value will be deducted according to service price list in consumption modules, and be back to the score calculation module for settlement; When users pass the tests of reward module, they will be added the corresponding scores in their IP; When the score accumulation surpasses threshold value *bmin*, the IP will be put in the queuing module for further processing； when the score of some IP in the queuing module is greater again than threshold value *bmin*, it will be taken out and put in the score calculation module to be given consumption qualification.

3.4.4 Reward module
This module is the key to judge whether a user is a normal user or not. But if the IP has been certified as a normal user, the module will not work. It is responsible for sending some tests to the client, collects client's response, and then rewards scores according to the client's performance.

3.4.5 Consumption module
It is responsible for the price list of a page. Each time when receiving some requirements, this module will check the corresponding service prices in the list and then pass the price information to score calculation module. When the requested user is approved to be a normal one, the service price in the score calculation module passed by this module is 0.

3.4.6 Queuing module
This module is to maintain a suspected blacklist and receive the suspected attacking IP from score calculation module. It is also responsible for recording the existing time of an IP in the blacklist. The longer it stays, the more suspicious the IP. Finally the IP whose time accumulative value surpasses the threshold value *Tbad* will be sent to the firewall to be filtered.

3.4.7 User control module
This module maintains a certification user list(the white list). Its control algorithm is as the following:

(1)When getting a requirement, first compare it with the IP fields in the whitelist. If there is any matching record, user control module will notify the consumption module to make the service price become 0.

(2)When consuming modules inform of IP abnormity in the whitelist, the record will be deleted in the whitelist and the score value will be reduced to half of *bmax*. If the IP's corresponding user is in the status of logging in, its log status will be removed and forced to logged in again.

3.5 Flow analysis
The flow of this module is divided into abnormal access and normal access.

3.5.1 The flowchart of abnormal access(see figure 2)
(1)As for each new user or new IP, the initial score value is *a*;

(2)Before users access to webpage or demand service, they will meet different users' behavior test with different probability. When passing the tests of rewarding module, the system will add responding rewarding points to the user's IP;

(3)When the score value is less than the minimum threshold value *bmin*, the IP will be put in the queuing module.

(4)IP in the queuing module cannot get the service, but it can get the hint of log-in/registration, so it can get the certification of normal users.

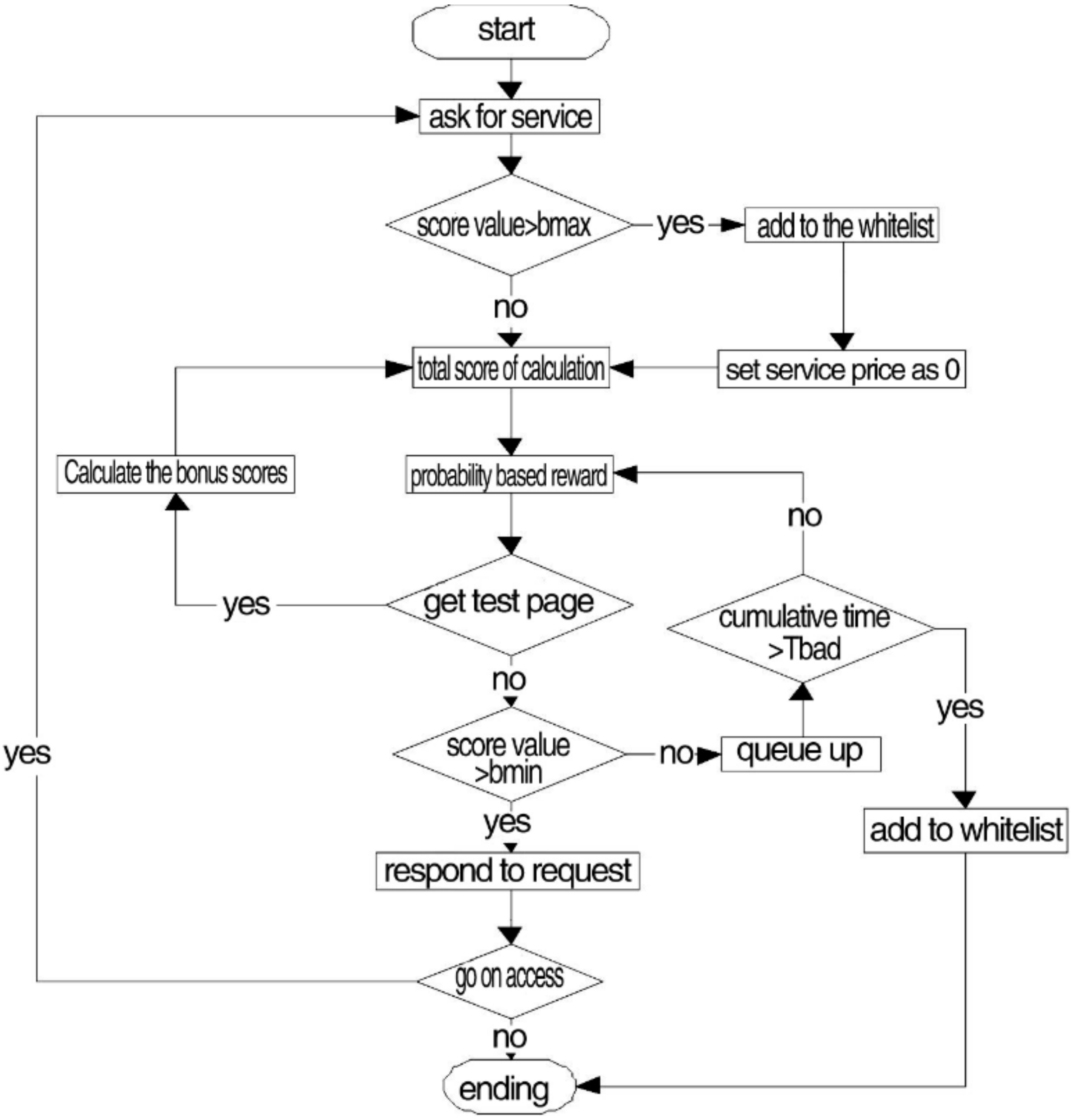

Figure 2: Flowchart of abnormal access

(5)When the score of some IP in the queuing module is again greater than threshold value *bmin*, take it out and put in the score calculation module, it can again get the qualification of consumption.

(6)When the score accumulation is over threshold value *bmax*, users will be regarded as normal, then put this IP in the whitelist in the user control module, the subsequent service will no longer consume any scores.

(7)Each time when demanding a service or accessing a page, the responding service price will be deducted, and be back to score calculation module for settlement.

(8)The queuing module will post the IP whose time accumulative value is greater than *Tbad* to the firewalls to filter.

3.5.2 Normal access flow（see figure3）

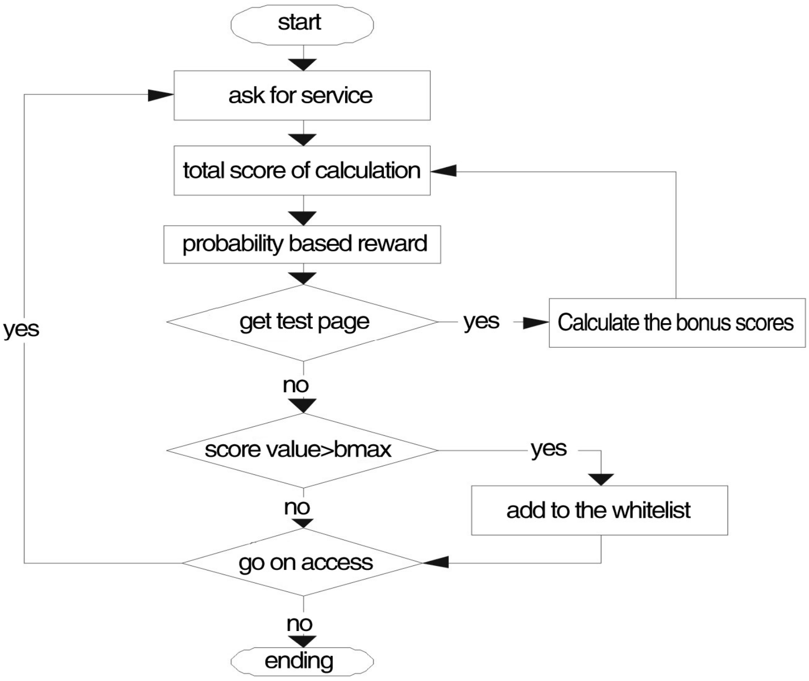

Figure3: Flowchart of normal access

This model has no requirement limitation. By the judgment of score strategy, it is used as the whitelist when being attacked.

THE EXPERIMENT AND ANALYSIS

4.1Experimental environment and steps

To prove the effect of this strategy, this paper sets up the intranet simulation test environment. Due to the limited laboratory conditions, here the author uses 4 attack hosts, a legal user host, a server host, an exchanger and a router. The bandwidth of the intranet is 10Mbps. The score strategy is deployed in the server host. The experimental topology is shown in figure 4, and device configuration is shown in table1:

Table1: Device configuration table

Device name	device configuration	IP address
Attacker1	Intel dual-core processor，over 2.50GHZ，2G RAM， windows 7	192.168.0.100
Attacker2	Intel dual-core processor，over 2.80GHZ ，2G RAM ， windows 7	192.168.0.105
Attacker3	Intel dual-core processor over2.00GHZ，2G RAM，windows7	192.168.0.110
User's computer	Intel dual-core processor，over2.00GHZ，2G RAM， windows XP	192.168.0.106
Server host	Intel dual-core processor，over3.00GHZ，2G RAM， windows XP	192.168.0.101

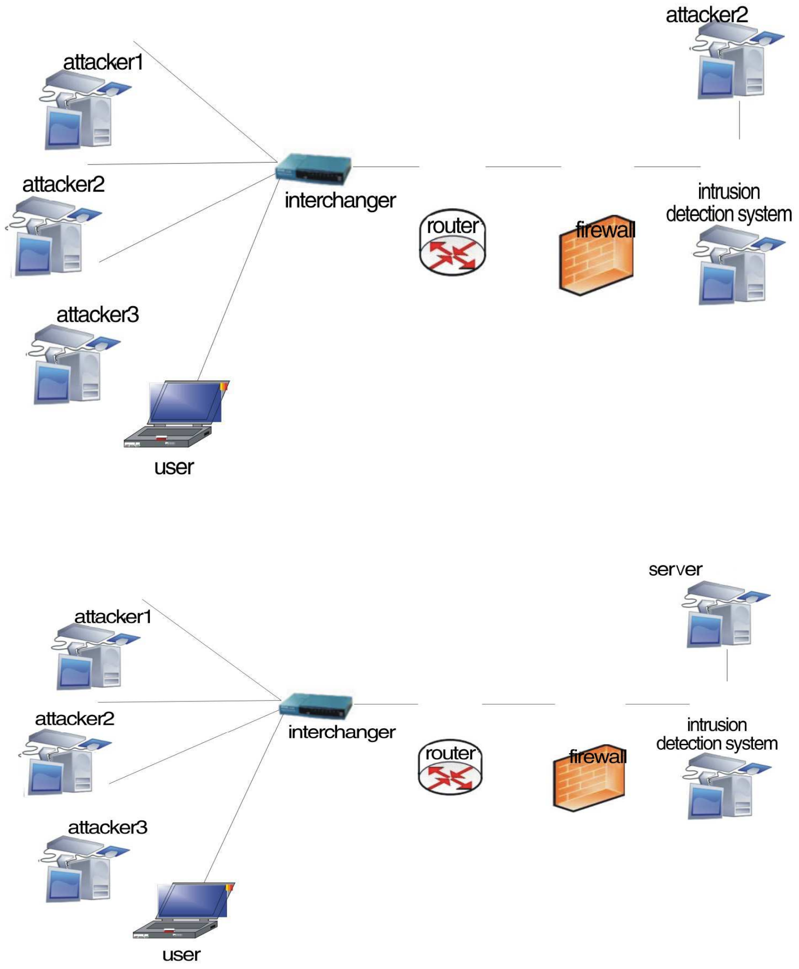

Figure 4: The experimental topology

The server host uses the combination of tomcat and MySQL, and adopts the query page of some website as the testing page. Normal users' routine has access to the testing page every 2 seconds. When attack begins, 3 attackers will simultaneously enable 200 threads to access the testing page of the server host without any interval and with wireless loop.

4.2 Experimental results and analysis

Figure 5 shows the scene without any strategy. With the constant request of the attacker, the server resources are gradually consumed, then the internal storage overflows, and the servers becomes very slow. Whether it is a normal user or attackers, the response time becomes high from the beginning low response and finally has no response.

Figure6 shows that the server has adopted score strategy. Each IP is set to have 200 score value, when the attacker's score value is 0, the server will stop handling its request. As is shown in figure6, when the timestamp is 7 second, the attack happens. With the consumption of the server, the response time of the normal users increases. When the attacker's score value is completely consumed, its request is frozen, the response time of the normal users' returns to normality.

The above experimental results show that the score strategy proposed by this paper has an excellent defense effect on defending HTTP's DDOS attack.

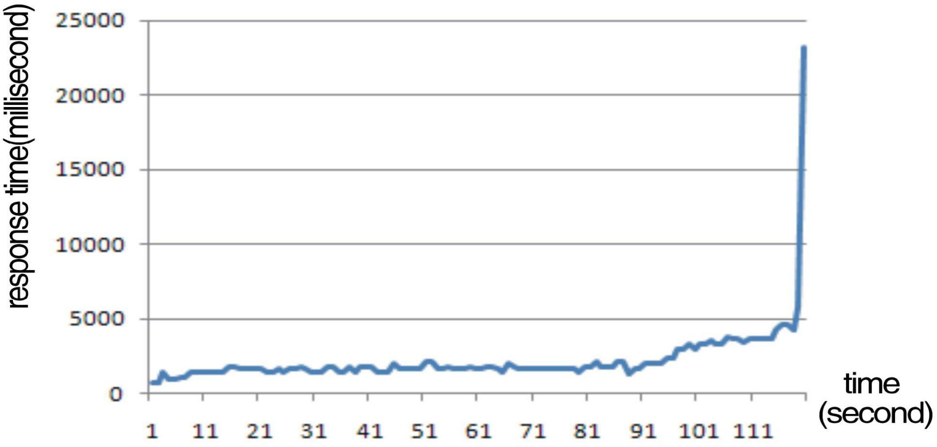

Figure5: Server response time without score strategy

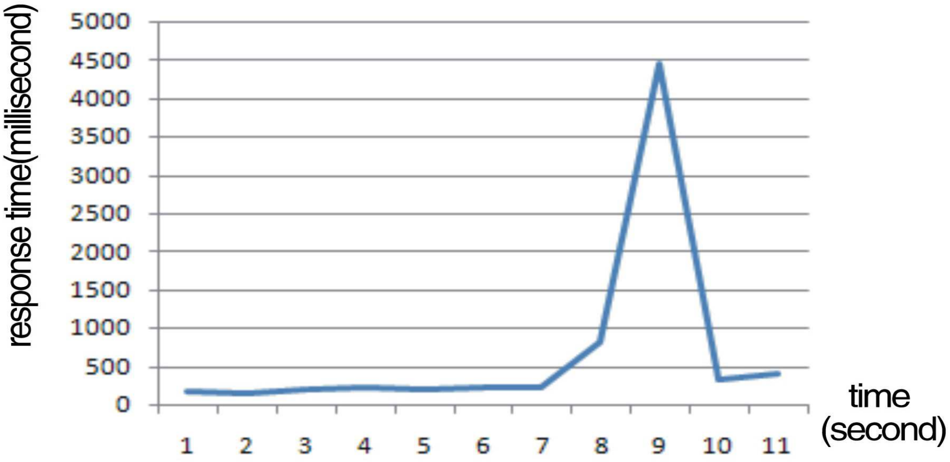

Figure 6: Figure of server response time with score strategy

CONCLUSION

This paper comes up with a score-strategy-based APP-DDOS defense model considering the feature of APP-DDOS, which distinguishes different users according to their IP. It responds to user's service request according to the score value with dynamic adjustment, and rapidly screens unusual requests with the maximum guarantee of normal user's access. The result of the experiment shows that the model has a good defense effect on defending HTTP's DDOS attack

The next focus of the paper is to reasonably control the size of the whitelist in the user's control module, and further optimize every parameter to improve the DDOS defense effect at the application level. The author also hopes that such relevant researches can open a door or guide the learning direction for English learners in China.

REFERENCES

[1] Wu, Yue, Gelan Yang, Huixia Jin, and Joseph P. Noonan. *Journal of Electronic Imaging*. **2012,** 21(1): 013014-1.
[2] Wong K-W, Yuen C-H.. *IEEE Trans Circuits Syst Express Brief*. **2008**, 55(11):1193-1197.
[3] Gao, W., Y. Gao and L. Liang, *Journal of Chemical and Pharmaceutical Research*. **2013a**,5(9):592-598.
[4] Yan, L., W. Gao and J. Li, *Journal of Applied Sciences*, **2013**,13(16): 3257-3262.
[5] Mitchell T M, Hutchinson R, Niculescu R S, et al. *Machine Learning*, **2004**, 57(1-2): 145-175.
[6] Gao,Y. and W. Gao, *International Journal of Machine Learning and Computing*. **2012**,2(2): 107-112.
[7] Mork, P. and P. Bernstein, *IEEE Comput. Soc.* **2004**.0918.

[8] W. Chen, C. Quan, C.J. Tay. *Optics Communications*. **2009** , (282): 3680–3685.

[9] Wong K-W, Yuen C-H. *IEEE Trans Circuits Syst Express Brief.* **2008**, 55(11):1193-1197.

[10] Lu Xingzhou. A DDoS defense plan against the key service of large-scale network [D]. Shanghai: East China Normal University, **2012**.

[11] Xiao Jun, Yun Xiaochun, Zhang Yongzheng. *Chinese Journal of Computers*, **2010**,09:1713-1724.

[12] Xie Ya. Research on detection method of application layer DDoS attacks based on fuzzy comprehensive evaluation [D].Southwest Jiaotong University,**2009**

[13] Ji Haijin,Cai Ming. *Computer Engineering and Design*, **2007**, 19:4619-4621

[14] Doron E, Wool A: *Computer Networks*, **2011**, 55(5):1037-1051.

[15] Ye Xi, Wen Wushao,Ye Yiru. *Telecommunications Science*, **2012**, 10:88-93.