DDoS Attacks Detection Model and its Application

[1,2] MUHAI LI, [1] MING LI, [2] XIUYING JIANG
[1] School of Information Science & Technology
East China Normal University
No. 500, Dong-Chuan Road, Shanghai 200241, PR. China
muhaili@126.com, mli@ee.ecnu.edu.cn
[2] Department of Computer Science
Zaozhuang University
Bei-An Road, Shandong 277160, PR. China

Abstract: With the proliferation of Internet applications and network-centric services, network and system security issues are more important than before. In the past few years, cyber attacks, including distributed denial-of-service (DDoS) attacks, have a significant increase on the Internet, resulting in degraded confidence and trusts in the use of Internet. However, the present DDoS attack detection techniques face a problem that they cannot distinguish flooding attacks from abrupt changes of legitimate activity. In this paper, we give a model for detecting DDoS attacks based on network traffic feature to solve the problem above. In order to apply the model conveniently, we design its implementation algorithm. By using actual data to evaluate the algorithm, the evaluation result shows that it can identify DDoS attacks.

Key- words: Algorithm, Attack, Application, DDoS, Detection, Modal

1 Introduction

A DDoS attack is a Denial-of-Service (DOS) attack, it has become one of the major threats and among the hardest security problems in today’s internet. whose impact has been well demonstrated in many computer network literatures.

A DoS attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service [1]. Examples include

- attempts to "flood" a network, in order to prevent legitimate network traffic.
- attempts to disrupt connections between two machines, thereby preventing access to a service.
- attempts to keep a particular individual from accessing a service.
- attempts to stop service to a specific system or person.

The goal of a DoS attack is to prevent a computer or network from providing normal services. The most common DoS attacks will target the computer's network bandwidth or connectivity. Bandwidth attacks flood the network with such a high volume of traffic that all available network resources are consumed and legitimate user requests cannot be responded. Connectivity attacks flood a computer with so many connection requests that they consume all available operating system resources, and result in the computer can no longer process requests of legitimate user.

Distributed Denial of Service (DDoS) is a relatively simple, yet very powerful technique to attack Internet resources. DDoS attack adds the many-to-one dimension to the DoS attack problem, and makes the prevention and mitigation of these attacks more difficult and the impact proportionally severe.

Unlike DoS attacks that rely on a specific network protocol or a system weakness, the DDoS attackers do not require to master high computer technologies, they can attack a site server with simply exploiting the huge resource asymmetry between the Internet and the victim, namely “many to one.” Before attacking, the attackers have controlled a sufficient number of zombies. Then they command these zombies generate so huge amounts of “useless” packets that overwhelm victim. Both DoS and DDoS attacks are have the same goal, this is to say, all of they want to tie up certain network resources completely so that the victim server denies services for legitimate users.

Compared with a DoS attack, A DDoS attack is very difficult to be defended. Because DDoS attack can make use of opening internet feature, which is that a large number of users can be permitted to visit the same site server at the same time. The feature of internet makes the DDoS attack be able to block access to the “thoroughfare” reaching the victim, effectively taking the victim off the Internet so that any victim-level of defense becomes irrelevant. In addition, the DDoS attack’s strategies of hierarchical attack and the technologies of IP spoofing make attackers difficult to be traced. Although great efforts have been involved in attack detection and prevention, there is still a lack of effective and efficient solutions to intercept ongoing attacks in a timely fashion, i.e. short enough to prevent traffic build up from DDOS attack. By now, DDoS attacks have risen to be the Number 1 threat on the Internet [2],

DDoS attacks are comprised of packet streams from disparate attack sources. Attacker can coordinate the power of a vast number of Internet zombies to consume some critical resource of the target and makes the site server deny the service to legitimate clients. Attack traffic is usually so similar to normal traffic that it is difficult to distinguish legitimate attack packets from normal packets. At the same time, the packet streams of DDoS attack have no apparent characteristics that could be directly and wholesalely used for detection and filtering. For keeping from tracing, attackers afford to change attack packet fields (especially *IP* address). With the rapid development of computer technologies, there are more and more extremely sophisticated, “user-friendly” and powerful DDoS toolkits, it makes DDoS attacking programs have very simple logic structures and possess less memory sizes, and makes them relatively easy to implement and hide. Attackers constantly change their tools to bypass inspection of security systems developed by system managers and researchers, who are in a constant alert to modify their approaches to handle new attacks. The DDoS field is evolving quickly, it is becoming increasingly hard to detect the attack. DDoS attacks are getting more sophisticated, spreading faster, and causing more damages [3].

However, there have not been developed fundamental defense solutions of DDoS attacks since these attacks have firstly appeared in June 1998 [4]. Therefore, it is necessary to study a new detection model and keep away DDoS attacks.

the goal of DDoS Attacks is in order to make the site deny the service of legitimate users, it is necessary to send such a large number of “garbage” packets to victim that the victim’s system has not ability to handle them. Therefore, the method recognizing abnormal increase of traffic is the shortcut to detect DDoS attack. In this paper, we just use the ideal to build a modal to solve the detection problem of DDoS attacks. Following this introduction, the paper is organized as follows. Section 2 introduces previous work on DDoS attacks. Section 3 gives the method how to build detecting modal. In this section, we discuss the feature of network traffic, which is the base to build detecting modal, and give an implement algorithm of the detecting modal. Section 4 applies detecting algorithm to verify validity of the modal. Section 5 draws the conclusion of this paper.

2 Previous work

There are a number of DDoS and DoS attack studies [5–7], Most of them address vulnerabilities or possible countermeasures, but few focus on attack detection.

More recent reports [8–14], In [8], Anderson et al. rely on the use of a ‘send-permission-token’ to restrict DoS attacks. Kreibich et al. use a decoy computer, pattern-matching techniques, and protocol conformance checks technologies to create intrusion detection signatures [9]. In [10], Allen et al. use estimates of the Hurst parameter to identify attacks that cause a decrease in the traffic’s self-similarity. This method requires statistics of network traffic self-similarity before the attack.

Yu et al. give a statistical method, namely, Logistic Regression with separate protocols [11]. The method is a theoretical method for finding features in intrusion detection. Using the Support Vector Machine method, the separate protocol model provides better results with high classification accuracy and low false alarm rate. In [12], a general classification of DDoS attacks and methods to deal with them is given. The methods can detect each kind of DDoS attacks and choose an appropriate defense mechanism automatically.

With the great development of wavelet technology, many papers use the technology to build detection DDoS models [18–20], In [18], Carl et al. modify CUSUM approach to detect attacks by wavelet analysis. In the papers [19,20], they find DDoS attack points by wavelet decomposition of signals with singularities.

In the paper [23], Feinstein et al. provide two statistical methods of analyzing network traffic to find DDoS attacks. One monitors the entropy of the source addresses found in packet headers, while the other monitors the average traffic rates of the ‘most’ active addresses. Some papers, e.g., [21, 22, 24] use probabilistic techniques, such as covariance etc, to detect attacks.

All DDoS detection methods define an attack as an abnormal and noticeable deviation of some statistic of the monitored network traffic workload. Clearly, the choice of statistic-based detection techniques is critically important. In [15], Glenn Carl et al. give a conclusion about methods of detecting attacks. At present, there are three kinds of detection technologies such as activity profiling, change point detection, and wavelet-based signal analysis, but all these techniques face the considerable challenge of discriminating network-based flooding attacks from sudden increases in legitimate activity or flash events. In order to meet the challenge, we have done many research works, e.g., [16, 17]. In paper [16], we give a detection model with low false alarm and low miss probability. In paper [16], we apply the Hurst parameter estimate to determine whether the system is under attacks.

Although each detection technique shows promise in limited testing, none completely solves the detection problem [15]. The major shortcoming of classic techniques is that they do not distinguish anomalies from attacks. For example, they cannot be different anomalies from sudden changes at 08:00 a.m., which is the beginning of office hours [19].

In this paper, we try to solve the problem above with using known normal traffic before detecting attacks.

3 Detection model

Let $y(t)$ denote a site total traffic, which is the number of bytes arriving at a site (or server) at time t. Hereby, $y(t)$ can be divided into normal traffic $n(t)$ and attack traffic $a(t)$, where attack traffic is generated by attackers. Then $y(t)$ can be abstractly expressed by

$$y(t) = n(t) + a(t) \qquad (1)$$

Obviously, when a site is not attacked, $a(t) = 0$, this is to say, $y(t) = n(t)$. When the site is under attacks, $a(t)$ will rapidly increase to high level. Therefore, if we can get the value of $a(t)$ during detection, it should be very easy to discover attacks. Unfortunately, we have no way to get $a(t)$ directly during detecting attacks. However, $y(t)$ can be captured with sniffer software conveniently. According to Eq. (1), if we can get the value of $n(t)$, then the aforementioned problem can be solved simply. But $n(t)$ is also unknown in a period of detection yet. Hence, how to get $n(t)$ becomes an essential problem.

3.1 Feature analysis of traffic

In order to solve the problem above, it is necessary to know the features of network traffic.

For achieving the aim of DDoS attacks, the attackers must sent large volume of "garbash" packets to victim. Therefore, the attacks traffic is usually far more than normal traffic. This is a basic feature of traffic.

About traffic feature, there are many literatures to study it, e.g., [23, 25–28].

In [23], Feinstein et al. discusses two kinds of detection methods. They define entropy H, and give a computing formula following as:

$$H = -\sum_{i=1}^{n} p_i \log_2{}^{p_i} ,$$

where p_i is probability of n independent symbols, the symbols can be *IP* addresses. Hence, the entropy can be computed on a sample of consecutive packets.

Through experiments, they have observed that while a network is not under attack, the entropy value of user *IP* addresses falls in narrow range. According to the definition of entropy, the value of entropy is actually the purity of *IP* addresses. This means that the number of new *IP* address is proportional to the one of old *IP* addresses. Actually, the old *IP* addresses represent common users of the site, and new addresses can be regard as new users or random users. At the same time, experiments also show that the number of common users is far more than the one of the new users of the site in normal state [28],

During studying feature of network traffic, we have done many statistic experiments about *IP* addresses, and also discovered the phenomenon above. We call the phenomenon the one of traffic features:

- In normal state, the common users of a site are stable, and the ratio of the number of them to the one of all the site users is approximately a constant. That is to say, if let $n_C(t)$ and $n_A(t)$ denote the number of common users and all users at time t respectively, then $\frac{n_c(t)}{n_a(t)}$ is almost a const.

In [26], Barford et al. give few traffic curves of weeks. These curves can clearly show similarity of traffic. Especially the traffic curves of the same day in different weeks are so. This is another feature of traffic:

- In the normal state, traffic has daily and weekly cycles [26, 27].This is to say, the traffic is similar at the same time of different dates in a certain period. If let C denote time cycle value of a day or a week, then y(t) is almost equal to y(t+C).

Fig. 1 and Fig. 2 can clearly show the feature. For example, the traffic at 8 a.m. on Dec. 18 is similar to the one at same time on Dec. 19.

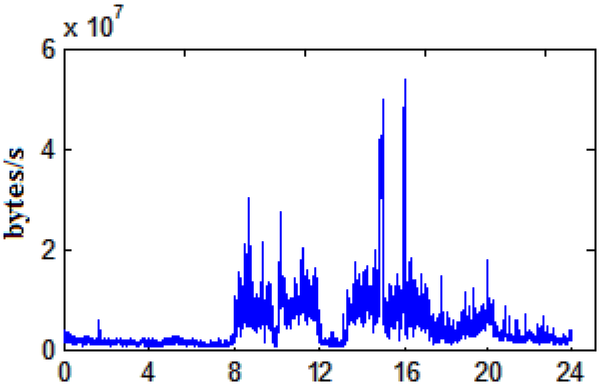

Fig. 1: The curve of traffic on Dec. 18, 2007.

Note: The abrupt changes around 3 p.m. include attack traffic in Fig. 1.

The reason why traffic has these features is mainly that a site provides stable services in a certain period. On the one hand, the stable services certainly constrain the requirement of its users. On the other hand, every user has stable requirement for the server, and steady work habit. The two elements determine a server has stable common users.

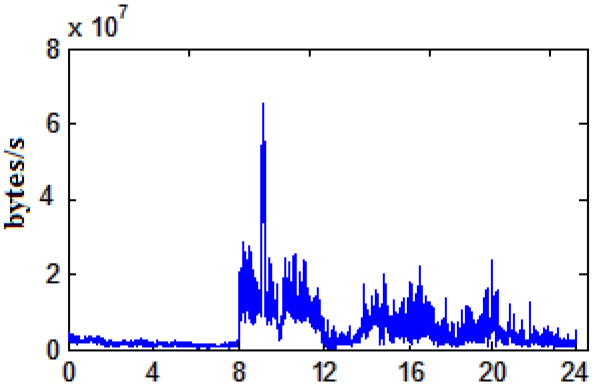

Fig. 2: The curve of traffic on Dec. 19, 2007.

Note: The abrupt changes around 9 a.m. include attack traffic in Fig. 2.

Undoubtedly, in normal state, there are few random users to visit the site, but they only browse the web accidentally. Hence, the traffic, which the random users generate, is far lower than the one of common users. Therefore, common users determinate that traffic of a site has similar feature.

According to the similar feature of network traffic, we can use statistic traffic, which came from a site under no attacks before detection, instead of normal traffic during detection. Let $N(t)$ denote the statistic traffic. So Eq. (1) can be rewritten as

$$a(t) = y(t) - N(t) \qquad (2)$$

Due to having known $y(t)$ and $N(t)$, we can build a detection model based on formula (2). To this purpose, we introduce a lemma as follows:

Lemma 1. x_i (i = 1, 2, ..., n) are n independent random variables, $y = x_1 + x_2 + ... + x_n$, For large n (e.g., n>30), the distribution of y approaches a normal distribution.

This lemma is just the central limit theorem in probability theory [29].

Theorem 1. In normal state, if the number of a site users is invariant, then the distribution of $y(t)$ approaches the normal distribution.

Proof. According to the condition of the theorem, we can assume that the site server has m users. Hereby, $y(t)$ can be expressed by

$$y(t) = y_1(t) + y_2(t) + ... + y_n(t),$$

where i =1, 2, ..., m, $y_i(t)$ is the traffic generated by the ith user.

In normal state, the site users are independent of each other, so their traffic $y_1(t)$, $y_2(t)$,…, $y_m(t)$ are naturally independent. In addition, the number of a site users is generally far greater than 30. Therefore, the traffic $y(t)$ satisfies the condition of lemma 1, the conclusion of the theorem is true. □

It is natural to think that we can build a detecting model based on Theorem 1. Unfortunately, the condition of Theorem 1 is not always satisfied, and sometimes the number of site users changes promptly. For instance, the number of the site users will be abrupt increase at 08:00 a.m.. Because the time is beginning of office hours, there are many users log in the site, and lead to traffic increase rapidly, Fig. 1 and Fig. 2 show it clearly. Obviously, the model relates with the starting time of detection. So if using the model to detect attacks, the result may not be good.

According to traffic similar feature, in normal state, $y(t)$ $-N(t)$, namely $a(t)$, can eliminate the majority of abrupt changes. However, we cannot use $a(t)$ to build the detecting model yet, because the value of $a(t)$ is mainly determined by the random traffic. According to the second feature of traffic, the random traffic is proportional to normal traffic, at the same time, the number of common users relates with the detecting time. Therefore, it is not better that only using the value of $a(t)$ to build detecting modal.

The second traffic feature can be used to solve the problem above, we discover that $\frac{a(t)}{N(t)}$ is a random variable, and is independent of the beginning time of detection in normal state, So, we use it to build detecting model. In the rest of this paper, let $A(t)$ denotes $\frac{a(t)}{N(t)}$.

Theorem 2. In normal state, the distribution of $A(t)$ approaches the normal distribution with mean 0, and it is independent of the number of site users.

Proof. We assume the number of common users is m_t at time t , Let $n_{m_t}(t)$ and $N_{m_t}(t)$ be traffic of the m_t common users respectively. Let r_t, and s_t be the number of the other users of $y(t)$ and $N(t)$ at time t respectively, where the other users are just random users. Let $n_{r_t}(t)$ and $N_{s_t}(t)$ be the random users' traffic. Hence, we have $N(t)=N_{m_t}(t)+N_{s_t}(t)$, $y(t)=n_{m_t}(t)+n_{r_t}(t)$. Then

$$A(t)=\frac{a(t)}{N(t)}=\frac{n_{m_t}(t)-N_{m_t}(t)}{N(t)}+\frac{n_{r_t}(t)-N_{s_t}(t)}{N(t)}.$$

In normal state, $y(t)$ is just n(t), According to the similar feature of traffic, $n(t)\approx N(t)$, $n_{m_t}(t)$ is far greater than $n_{r_t}(t)$, namely, $N(t) \geq n_{m_t}(t) >> n_{r_t}(t)$, where '>>' denotes far more than. Similarly, $N(t) \geq N_{m_t}(t) >> N_{s_t}(t)$. This means that $\frac{n_{r_t}(t)-N_{s_t}(t)}{N(t)}$ is almost zero. Hence, the distribution of $A(t)$ is determined by the one of $\frac{n_{m_t}(t)-N_{m_t}(t)}{N(t)}$. Since $n_{m_t}(t)$ and $N_{m_t}(t)$ come from the same group of common users, hence the distribution of $\frac{n_{m_t}(t)-N_{m_t}(t)}{N(t)}$ has mean zero. Because of the traffic feature, $\frac{n_{m_t}(t)}{N(t)}$ and $\frac{N_{m_t}(t)}{N(t)}$ are almost constants which are independent of m_t, (i.e. the number of common users), According to Theorem 1, the distribution of $\frac{n_{m_t}(t)-N_{m_t}(t)}{N(t)}$ approaches has normal distribution with mean 0.□

3.2 Building detection model

When the site is under attack, $n_{r_t}(t)$ includes attack traffic, this leads to $n_{r_t}(t) \gg N(t)$. Hence, the mean of $A(t)$ is far greater than zero. Using Theorem 2, we can get a detecting method: if $A(t)$ yields normal distribution with mean zero, we can determine the server is secure, otherwise, there are attacks.

We will build a model for detecting attacks with the parameters estimate method of probability theory.

Let T and η be the number of samples and the mean of random variable $A(t)$ respectively, $u(T)$ is the sample mean of $A(t)$ with T samples. For the variance of $A(t)$ is unknown, in order to estimate the mean η, we form the sample variance $S(T)$:

$$S^2(T)=\frac{1}{T-1}\sum_{t=0}^{T-1}[A(t)-u(T)]^2 .$$

In fact, the $S^2(T)$ is an unbiased estimate of the variance of $A(t)$ [29]. Thus, under the assumption that $A(t)$ is normal, the ratio

$\frac{u(T)-\eta}{S(T)/\sqrt{T}}$ has a *Student-t* distribution with T–1 degrees of freedom [29]. Using the distribution, we can estimate the mean η. If we have known the confidence coefficient P, then η yields the approximate confidence interval

$$u(T)+t_{\frac{\delta}{2}}\frac{S(T)}{\sqrt{T}}<\eta<u(T)+t_{1-\frac{\delta}{2}}\frac{S(T)}{\sqrt{T}},$$

where δ=1–P, $t_{\frac{\delta}{2}}$ and $t_{1-\frac{\delta}{2}}$ are the percentiles of the t distribution respectively.

Appling actual data to this model, we discover, if the site is not under attack, the confidence interval of η is included in (–0.5, 0.5). Otherwise, the relation above is not true. Thus, we obtain a model for detecting DDoS attacks.

Using the model above, we give a run-time detecting algorithm as follows

1) Assign *P* and *T* an initial value respectively. the starting time of detection is 0.
2) Open a database, which has stored statistic traffic of the site. Fetch data from the database and load the data into array *N*(*t*); These data correspond with the time from 0 to *T* –1.
3) Set *u*(0) = *N*(0); *S*(0) = *y*(0), where *y*(0) is the traffic datum at starting time 0.
4) Judge whether the relation $t \ge T$ is satisfied. If the answer is true, go to 8).
5) Capture the traffic of the site at time *t*, and load it into *y*(*t*).
6) Compute *u*(*t*) and *S*(*t*).
7) Let *t* = *t* + 1, and go to 4).
8) Compute the confidence interval of η, this is

$$(u(T)+t_{\frac{1-P}{2}}\frac{S(T)}{\sqrt{T}}, u(T)+t_{1-\frac{(1-P)}{2}}\frac{S(T)}{\sqrt{T}}),$$

where *u*(*T*) and *S*(*T*) can be computed with recursive algorithm below.

9) Judge whether the confidence interval of η is included in (–0.5, 0.5), if the result is Yes, then the site is safe; otherwise, gives an attack alarm.
10) End.

For improving efficiency of detection, *u*(*t*) and *S*(*t*) can be computed with recursive algorithm. The recursive algorithm of *u*(*t*) is as follows:

$$u(t)=\frac{1}{t}\sum_{s=0}^{t-1}A(s)=\frac{1}{t}\sum_{s=0}^{t-2}A(s)+\frac{1}{t}\frac{a(t)}{N(t)}$$
$$=\frac{t-1}{t}u(t-1)+\frac{1}{t}[\frac{y(t)}{N(t)}-1].$$

The recursive algorithm of *S*(*t*) is

$$S^2(t)=\frac{1}{t-1}\sum_{s=0}^{t-1}[A(s)-u(t)]^2$$
$$=\frac{1}{t-1}[\sum_{s=0}^{t-1}A(s)^2-2u(t)\sum_{s=0}^{t-1}A(s)+\sum_{s=0}^{t-1}u(t)^2]$$
$$=\frac{1}{t-1}\sum_{s=0}^{t-1}A(s)^2-\frac{2t}{t-1}u(t)^2+\frac{t}{t-1}u(t)^2$$
$$=\frac{t}{t-1}[\overline{A(t)^2}-u(t)^2],$$

where

$$\overline{A(t)^2}=\frac{1}{t}\sum_{s=0}^{t-1}A(s)^2=\frac{t-1}{t}\overline{A(t-1)^2}+\frac{1}{t}[\frac{y(t)}{N(t)}-1]^2.$$

Obviously, the algorithms time complexity is *O*(*T*). Hence, the recursive algorithms make the detecting modal has more high efficiency of execution, and can help the modal finish run-time detection of DDoS attacks.

4 Model Application

For verifying the algorithm above, we sample a large of data from a central server in Zaozhuang University with Sniffer software. The sample time interval is 10s. Fig. 3 shows the curve of the data.

Fig. 3 is component of three sections; the first section is statistic traffic, which was sampled before detection without attacks. The other two sections represent the data sampled on Dec. 12 and Dec. 19 in 2007 respectively.

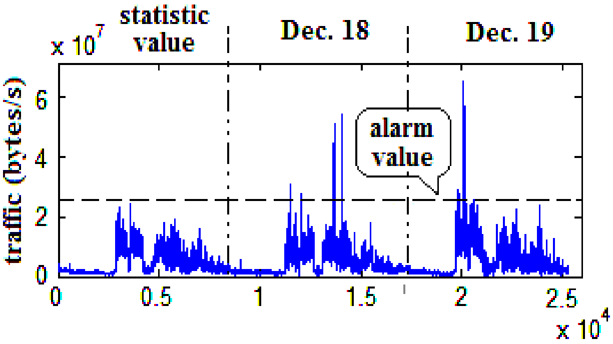

Fig. 3: Statistic and detection traffic

From the Fig. 3, it is obvious to see that the traffic has a similar characteristic. We can also discover some abnormal traffic in the figure. In fact, some of them are generated with attack software. We apply attack software to attack the server three times. Two of them occurred on Dec. 18, the first attack was at 2:33 p.m., and the time length of the attack is 10 minutes. The second attack is at 3:36 p.m., the time length of attack is 8 minutes.. There was one attack to the site on Dec.19, and the attack lasted 11 minutes. From Fig. 3, we can see the abrupt changes of traffic at corresponding time.

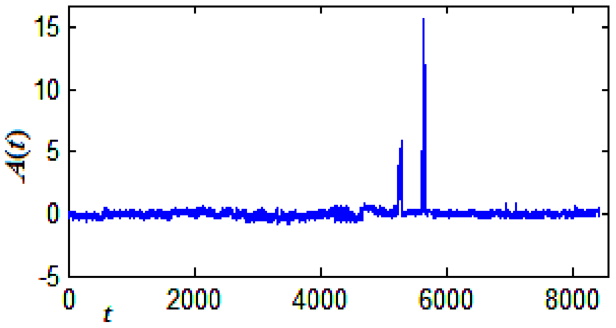

Fig. 4: The curve of $A(t)$ on Dec. 18, 2007

Fig. 4 is the curve of $A(t)$ on Dec. 18, 2007. From the figure, it can be easy to see that the curve is independent of traffic scale. Tow abrupt changes represent the site is being under attacks at that time.

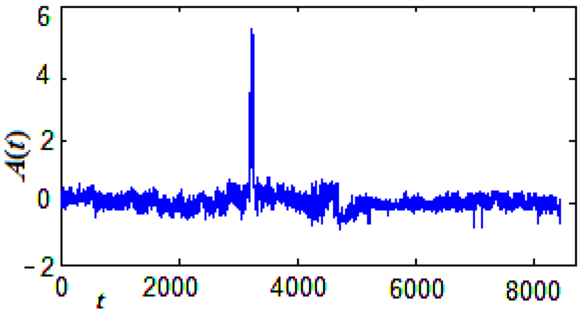

Fig. 5: The curve of $A(t)$ on Dec. 19, 2007

Fig. 5 represents the curve of $A(t)$ on Dec. 19, 2007. We can clear see a abrupt change in the figure, the change is caused by attack traffic, and shows that the site is being under attacks at that time. In addition, we can also see that the curve is independent of network traffic scale.

For improving the efficiency of detection, we set an alarm value. Once the traffic of the server reaches it, detection program will start automatically. In this paper, the alarm value is 2.5×10^{6}; the length of detection time is 10 minutes; confidence coefficient P is 0.95; the sample time interval is 10s. On Dec. 18, 2007, the detection program was executed four times; two of them gave attack alarm. On another day, the server was detected five times automatically, we got two attack alarms. Table 1 and Table 2 show the results of detection.

Table 1: The detection results on Dec. 18, 2007

No.	Starting time	Confidence interval	state
1	8:30	（−0.0151, 0.1001）	no
2	10:01	（−0.1409, 0.0883）	no
3	14:33	**（2.3799,3.0245）**	**yes**
4	15:36	**（3.6149,5.0761）**	**yes**

Note: In Table 1,"yes" represents the site is under attacks, "no" means not.

The table 1 shows that two Confidence intervals are not include in (–0.5,05), this means that the site was under attacks at 2:33 p. m. and 3:36 p. m. on Dec. 18, 2007 respectively.

Table2: The detection results on Dec. 19, 2007

No.	Starting time	Confidence interval	state
1	8:05	（0.2973,0.3697）	no
2	8:22	（0.0449,0.1684）	no
3	8:55	**（3.2534,3.7432）**	**yes**
4	9:05	**（0.2954,0.7259）**	**yes**
5	10:29	（0.1439,0.2060）	no

Note: The meaning of "yes" and "no" in Table2 is the same as the one in Table1.

Table 2 shows as if that the site server was under attacks two times on Dec. 19, 2007. However, we actually attacked the site one time on that day. This is because the length of attack time is longer than the one of detection time. Thus, the fourth detection used 1-minute attack data. Therefore, we received two alarms.

The example shows that our detection algorithm can identify whether the server is under attacks.

5 Conclusion

In this paper, by studying the basic feature of traffic, we give a model of detecting DDoS attacks. The model cannot be influenced by abrupt changes of normal traffic, and is independent of the starting time of detection. Hence the modal do it’s better in detecting DDoS attack. During detection, the modal do not used the signatures of DDoS attacks, so it can detect unknown DDoS attacks. This is to say the detecting modal is more robust. In order to realize run-time detection, we give an implementation algorithm of the model with simple structure, low complexity, and low memory possession. With actual data to test the algorithm, the results show the algorithm can rapidly identify whether the server is under attacks. However, the detecting modal is dependent on statistic traffic before detection, the quality of the statistic traffic directly affect on the result of detecting. Thus, it is very important to know the normal state of the site, and capture network traffic in time.

During the detection of DDoS attacks, we use the confidence interval (–0.5,05), In fact, the confidence interval is not invariant, it may vary with the difference of site, and relate with the precision of detection. If we require the modal can recognize the DDoS attacks that have slight attack traffic, the interval should be set up more small. Usually, the confidence interval (–0.5,05) is good choice for detection.

In future, we will study the control function of firewalls and routers about traffic, and try to build a management system, which can automatically detect, control, and manage the server.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under the project grant number 60573125, and by the Research and Development Project of Shandong Provincial Education Department under the project number J07WJ29.

Reference

[1] Denial of Service Attacks *http://www.cert.org/tech_tips/denial_of_service.html*, 2008.

[2] Background on DDoS *http://www.ddos.com/index.php?content=products/background.html*, 2008.

[3] Editorial, Distributed denial-of-service and intrusion detection, *Journal of Network and Computer Applications*, Vol. 30, 2007, pp.819 – 822.

[4] K. Lee, K. Kim, et al., DDoS attack detection method using cluster analysis, *Expert Systems with Applications*, 2007, doi:10.1016/j.eswa. 2007.01.040.

[5] V. Paxson, Bro: a system for detecting network intruders in realtime, *Computer Networks* Vol. 31, 1999, pp. 2435 – 63.

[6] L. Ricciuli, P. Lincoln, P. Kakkar, TCP SYN flooding defense, *Communication Networks and Distributed Systems Modeling and Simulation (CNDS '99)*, 1999, pp. 17 – 20.

[7] E. Strother, Denial of service protection the Nozzle, *In: Proceedings of the 16th annual computer security applications conference (ASAC’00)* , 2000, pp. 32 – 41.

[8] T. Anderson, T. Roscoe, D.Wetherall, Preventing internet denial-of-service with capabilities, *Computer Communications Review*, Vol. 34, No. 1, 2004, pp. 39 – 44.

[9] C. Kreibich, J. Crowcroft, Honeycomb – creating intrusion detection signatures using honeypots, *Computer Communication Review (ACM SIGCOMM)*, Vol. 34, No. 1, 2004, pp. 51 –56.

[10] W. Allen, G. Marin, The LoSS technique for detecting new denial of service attacks, *SoutheastCon, 2004. Proceedings. IEEE,* 2004, pp. 302-309.

[11] K. M. Yu, M. F. Wu, Protocol-Based With Feature Selection in Intrusion Detection, *WSEAS Transactions on Computer*, Vol. 3, No. 3, 2008, pp. 135 –146.

[12] A. Asosheh, N. Ramezania, A comprehensive faxonomy of DDoS attacks and defense mechanism applying in a smart classification, *WSEAS Transactions on Communications*, Vol 7, No. 4, 2008, pp. 281–290

[13] H. Sun, B. Fang, H. Zhang , A new intrusion

Detection Approach based on Network Tomography, *WSEAS Transactions on Information Science & Applications*, Vol.3, No. 2, 2006, pp. 211– 217.

[14] D. H. Kang, B. K. Kim, J. T. Oh , Protocol anomaly and pattern matching based intrusion detection system, *WSEAS Transactions on Communication*, Vol.4, No. 10, 2005, pp. 994 – 1101.

[15] G. Carl, et al., Denial–of–Service Attack Detection Techniques, *IEEE Internet Computing*, Vol. 10, No. 1, 2006, pp. 82 – 89.

[16] M. Li, An approach to reliably identifying signs of DDoS flood attacks based on LRD traffic pattern recognition, *Computers & Security*, Vol. 23, 2004, pp. 549 – 558.

[17] M. Li, Change trend of averaged Hurst parameter of traffic under DDoS flood attacks, *Computers & Security*, Vol. 25, No. 3, 2006, pp. 213–220.

[18] G. Carl, R. R. Brook, S. Rai, Wavelet based denial-of-Service detection, *Computers & Security*, Vol. 25, 2006, pp. 600 – 615.

[19] M. Hamdi, N. Boudriga, Detecting denial- of-service attacks using the wavelet transform, *Computer Communications*. Vol. 30, 2007, pp. 3203 – 3213.

[20] A. Antoniadis, I. Gijbels, Detecting abrupt changes by wavelet methods, *Technical Report, Laboratoire LMC-IMAG. France: Universite Joseph Fourier*, 1997. 9.

[21] N. Ye, X. Li, et al., Probabilistic techniques for intrusion detection based on computer audit data, *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, Vol. 31, No.4, 2001, pp. 266 – 274.

[22] S.Y. Jina, et al., Network intrusion detection in covariance feature space, *Pattern Recognition*, Vol. 40, 2007, pp. 2185 –2197.

[23] L. Feinstein, et al., Statistical approaches to DDoS attack detection and response, *In: DARPA information survivability conference and exposition proceedings*, Vol. 1, 2003, pp. 303 – 314.

[24] C.Krugel, T.Toth, E.Kirda, Service specific anomaly detection for network intrusion detection, *ACM*, 2002.

[25] P. Abry, R. Baraniuk, et al., Multiscale nature of network traffic, *IEEE Signal Processing*, Vol. 19, No. 3, 2002, pp. 28 – 46.

[26] P. Barford, J. Kline, D. Plonka, A.Ron, A signal analysis of network traffic anomalies, *In: Proceedings of ACM SIGCOMM Internet measurement workshop, Marseilles, France*, 2002.

[27] V. Paxson, Measurements and analysis of end-to-end internet dynamics, *Ph.D. thesis, University of California Berkeley*, 1997.

[28] T. Peng, C. Leckie and R. Kotagiri. Detecting reflector attacks by sharing beliefs, *In: Proceedings of IEEE Global Conference, Globecom*, 2003.

[29] A. Papuilis, S. U. Pillai, *Probability, Random Variables, stochastic Processes*, McGraw-Hill Inc., 2002.