TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.1, January 2014, pp. 753 ~ 761
DOI: http://dx.doi.org/10.11591/telkomnika.v12i1.3355

Detecting DDoS Attacks Against DNS Servers Using Time Series Analysis

Tong Guang NI, Xiao Qing GU*, Hong Yuan WANG
School of Information Science and Engineering, Changzhou University, Changzhou 213164, China, Telp 86-0519-86330558, Fax 86-0519-86330284
e-mail: nitongguang@gmail.com, tiddyddd@163.com*, niyifeimomo@gmail.com

Abstract

Domain Name System (DNS) Service is the basic support of Internet, which security plays a vital role in the entire Internet. Because DNS requests and responses are mostly UDP-based, and the existing large numbers of open recursive DNS servers, it is vulnerable to distributed denial of services (DDoS) attacks. Through the analysis of several aspects of these attacks, a novel approach to detect DDoS attacks is proposed based on characteristics of attack traffics (CAT) time series. Then CAT time series are transformed into a multidimensional vector series and a support vector machine (SVM) classifier is applied to identity the attacks. The experiment results show that our approach can identify the state features of the abnormal flow due to the DDoS attacking flows, and detect DDoS attacks accurately.

Key words: *DNS server, distributed denial of service, time series, support vector machine*

1. Introduction

Current distributed denial of services (DDoS) attacks remains a high threat to Internet security. The attacks can be carried out by attack tools, worms, and botnets with attack variants of packet transmission [1]. These sources of DDoS attacks are powerful and can overwhelm any online host and server. Moreover, one of the biggest catastrophic DDoS attacks outcomes when this class of attacks triggered against core component of the Internet infrastructure like Domain Name System (DNS) services. DNS is Internet's address book, which mainly translates domain names to IP address for sending the packets out. Consequently, even a small part of the DNS infrastructures being unavailable for a short period of time could have a significant rippling effect on the rest of the Internet. In recent years, a number of DDoS attacks against the availability of DNS have occurred. As reported in [2], in May 2009 several root DNS services in China were suffered a massive DDoS attack, as a result, the network in southern six provinces are interrupted and the economic loss is about 23 millions. There are many other similar attacks launched against DNS services from 2010 to 2012.

There is a little research work towards the DDoS attacks against DNS servers. In [3], DNS-Guard is proposed to detect the spoofing DNS requests. It involves several policies that generate some form of cookies for a DNS server to implement origin authentication. However, it cannot verify the requests from general DNS resolver clients, which are the main attack tools in attacks. A new DNS transport protocol is proposed in [4], but this protocol cannot hold its promise when it is attacked. In [5], [6], the reserving TTL-expired records are used to detect the unavailability of DNS servers with the keep-alive scheme, but this method is not available when the TTL value is spoofed by attack. The incoming and outgoing IP addresses matching protocol is used to in [7], but it needs large memory size and unsalable for practical use.

In this paper, an efficient and real time detection approach is proposed based on the characteristics of attack traffics (CAT) time series. By approximating the adaptive autoregressive (AAR) model, CAT time series is transformed into a multidimensional vector series to depict the state features of DNS traffics. Furthermore, a support vector machine (SVM) classifier is applied to classify vector series and identity DDoS attacks.

2. Analysis of DDoS Attacks Against DNS Servers

DNS is a protocol which is not encrypted during the transmission process. DNS mainly uses UDP protocol, so it is easy to spoof source IP address as UDP is connectionless and does

Received May 27, 2013; Revised July 15, 2013; Accepted August 27, 2013

not use three-way handshake to start a connection like TCP. In the other hand, small DNS requests can general large DNS response messages in length. Normally, in the initial DNS specification the DNS response was restricted up to 512 bytes length, but in the condition of an Extended DNS (EDNS), the response message is much bigger [8]. As a result, the combination of the simplicity of the DNS protocol and it uses UDP makes DNS servers extremely vulnerable to DDoS attacks.

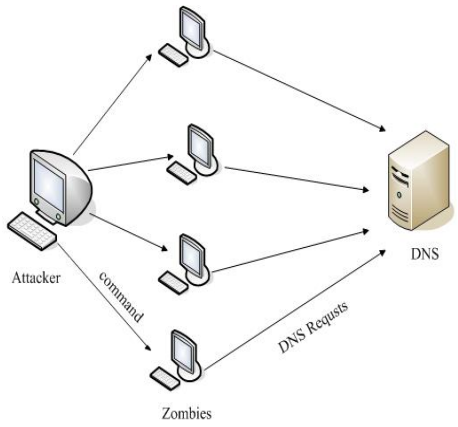


Figure 1. Architecture of a flooding DDoS attack

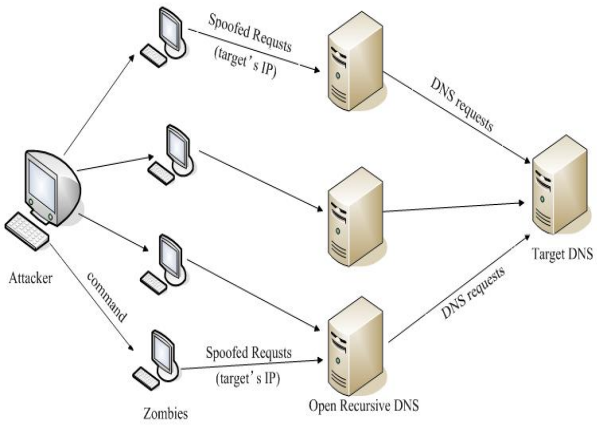


Figure 2. Architecture of a DNS amplification attack

DDoS attacks against DNS servers are classified into two classes [9]. One type of attacks directly floods DNS servers by sending a large number of DNS requests, called flooding DDoS attack. The goal of flooding attack is the expeditious consumption of a DNS server and making it unavailable to legitimate uses. As a standard DNS server cannot distinguish the spoofed requests from the non-spoofed ones, it would simply accept all requests and send the responses. When it becomes overloaded, DNS server starts to drop requests indiscriminately. An example of multiple sources flooding attack against a DNS server is shown in Figure 1. The attacker orchestrates several zombies to simultaneously generate spoofed DNS requests aiming at disrupting the normal DNS operation by consuming its resources; mainly memory and CPU.

Another type of attacks is to exploit open recursive DNS servers to amplify attack traffic, called DNS amplification attack. The open recursive servers are those that accept arbitrary DNS queries from any source and send the final response messages to the request [10]. The attacker spoofs the IP address of the victim to reflect the network traffic using open recursive DNS servers by initiating relatively small DNS requests. The attackers employs a distributed architecture similar to that presented in Figure 2, it is obvious that the bandwidth and resources at the victim server are consumed rapidly with the increase rate of the response message. As stated in [11], by combining different response types, the amplification effect can reach up to a factor higher than 60. It is not difficulty to lunch a DNS amplification attack, because 75% name server in world is an open resolver. Therefore, DNS amplification attacks may be stealthier and more dangerous for the DNS servers, owing to its amplification in attack effect and its difficulty to trace the attacker.

3. Detection of Attacks Based on CAT Time Series

3.1. Definition of CAT

To enhance the effect of the flooding attack, attackers usually have numerous zombie machines to set up a DDoS attack. At the same time, attackers fabricate spoofed requests, so DNS servers cannot resolver these domain names to IP address successfully. As a result, the resolutions of these spoofed requests are failed. In a flooding DDoS attack, the proportion of failed resolutions to successful resolutions is extremely high.

Based on the one-to-one mapping of the DNS request and response, under a normal operation when a client queries a name resolution sends a request towards local recursive sever, which is responsible to create the corresponding response. Nevertheless, in a DNS amplification attack, the targeted DNS server receives responses without having previously sent out the corresponding requests. Therefore by comparing the numbers of requests and responses in a certain time could be a key character of DNS amplification attack.

The traffic on a DNS server is complex network flows with strong outburst and instability. The traffic targeted is a stream of successive UDP requests and UDP responses. In this paper, the characteristics of attack traffics (CAT) series are sampled from these UDP packets. CAT is defined as follows, which reflects the change of the traffics distribution caused by DDoS attacks.

Definition 1: In a certain time interval Δt, the proportion of failed resolutions to successful resolutions (FSR) is defined as

$$FSR = \frac{\sum_{\mathrm{T}} false}{\sum_{\mathrm{T}} success} \tag{1}$$

where $\sum_{\mathrm{T}} false$ is the number of UDP responses that are resolved falsely, and $\sum_{\mathrm{T}} success$ is the number of UDP responses that are resolved successfully.

Definition 2: In a certain time interval Δt, the proportion of UDP responses to UDP requests (USQ) is defined as

$$USQ = \frac{\sum_{\mathrm{T}} responses}{\sum_{\mathrm{T}} requests} \tag{2}$$

where $\sum_{\mathrm{T}} responses$ is the number of UDP responses, and $\sum_{\mathrm{T}} requests$ is the number of UDP requests.

Definition 3: The characteristics of attack traffics (CAT) of DNS traffics is defined as

$$\mathrm{CAT} = \theta \times USQ + (1-\theta) \times FSR \qquad (0 < \theta < 1) \tag{3}$$

From the definitions, we can see that a flooding DDoS attack will increase the value of FSR, and a DNS amplification attack will increase the value of USQ. So a DDoS attack will result in an abnormal increase in CAT volume dramatically in a short time, thus CAT will form a new traffic state different from the normal one. Therefore, CAT can reflect the characters of DDoS attack against DNS server including the burst in the traffic volume, asymmetry of the flow.

3.2. Generation of CAT Time Series

DNS traffic is sampled with sampling period Δ*t* and calculate CAT of every interval. Therefore, the UDP packets arrival are modeled by CAT time series: $Z(N, \Delta t) = \{CAT_i, i=1, 2, ...,N\}$, where N is the length of the series. A stationary model is unable to adapt to changes in normal system behavior and has to be re-estimate if the normal system behavior evolved. In this paper, the adaptive autoregressive AAR (*p*) model [12], [13] is used to represented CAT time series. The model is defined as

$$a_i = \sum_{j=1}^{p} \phi_j(i) a_{i-j} + \varepsilon_i \tag{4}$$

where a_i is the observation at instant t, and weight vector $\phi(i)$ is the time-varying model parameters. As the traffic collecting device may cause measurement errors, stochastic variable ε_i is used to capture this error, assumed to be zero mean Gaussian white noise sequences. Suppose t called current time, and i called serial number, so $i = t/\Delta t$. The model uses a weighted sum of p previous values to estimate the current observation value. $\phi(i)$ (i=1,…, p), the parameters vector of AAR model , are time dependent, and the current value can be predicted as a linear combination of p past values, so weight vector $\phi(i)$ can be described as

$$\phi(i) = [\phi_1(i), \phi_2(i), ..., \phi_p(i)]^T \quad (5)$$

and its corresponding observation vector can be described as

$$A(i\text{-}1)=[a_{i-1}, a_{i-2}, \ldots, a_{i-p}]^T \quad (6)$$

Weight vector $\phi(i)$ cannot be observed directly, here Recursive Least Square (RLS) algorithm is used to estimate them. RLS algorithm is a kind of Kalman filter in nature, which exactly meets least square criterion [14], [15]. RLS is an adaptive and recursive data processing algorithm that is suited for on-line estimation. It can process traffic matrix as a whole and all traffic can be estimated simultaneously. Suppose $\hat{\phi}(i)$ called estimate of $\phi(i)$ at instant t-1, $\hat{\phi}(i)$ can be predicted by

$$\hat{\phi}(i) = \hat{\phi}(i-1) + k(i)^T \times \hat{\varepsilon}_i \quad (7)$$

In which $\hat{\varepsilon}_i$ is the error vector and is given by

$$\hat{\varepsilon}_i = a_i - \hat{\phi}(i-1)^T A(i-1) \quad (8)$$

The information gain matrix $k(i)$ is defined as

$$k(i) = T(i-1) \times A(i-1) / Q(i) \quad (9)$$

$$Q(i) = A(i-1)^T \times T(i-1) \times A(i-1) + \lambda \quad (10)$$

where $T(i\text{-}1)$ is covariance matrix, and is the N-by-N identity matrix.

$$T(i) = \frac{1}{\lambda}[T(i-1) - k(i) \times A(i-1)^T \times T(i-1)] \quad (11)$$

λ is forgetting factor. The contribution of previous samples is smaller when λ is smaller. It makes the filter more sensitive to recent samples, which means more fluctuations in the filter coefficients. Based on the equation (7)-(11), $\hat{\phi}(i)$, the estimate of parameters vector of AAR model, can be calculated. The initial Weight vector $\phi(0)$ can be set as any smaller values, such as 0, because the filter recursion process can update weight vector iteratively.

3.3. SVM Classifier

By sampling the DNS traffic with time interval Δt and calculating CAT of every sample, we get CAT time series. After transforming CAT time series into a multidimensional vector of degree p by estimating the AAR model using RLS algorithm, multidimensional vector $\hat{\phi}(i)$ of

degree *p* can be used to describe the state features of DNS traffic. As a result, detecting DDoS attacks equates to classifying $\hat{\phi}(i)$ series virtually.

The support vector machine (SVM) is a well-known machine learning method proposed by Vapnik et al [16], which is based on a limited sample of information to find the best compromise between model complexities and is ability to learn in order to obtain the best generalization. It can establish a mapping of a non-linear separable data sample in higher dimensional characteristic space by selecting the non-linear mapping function, structures the optimal hyperplane and the problem can be converted into a linearly separable one in the high-dimensional feature space. Furthermore, it solves the dimension problem and its computational complexity is independent of the sample's dimension. Since DNS traffic is detected as legitimate or malicious, attack detection can be viewed as a binary classification problem. Before the SVM can classify traffics, it should undergo a training process to develop a classification model. Here the LibSVM library [17] is used to implement SVM.

4. Experiments and Analysis

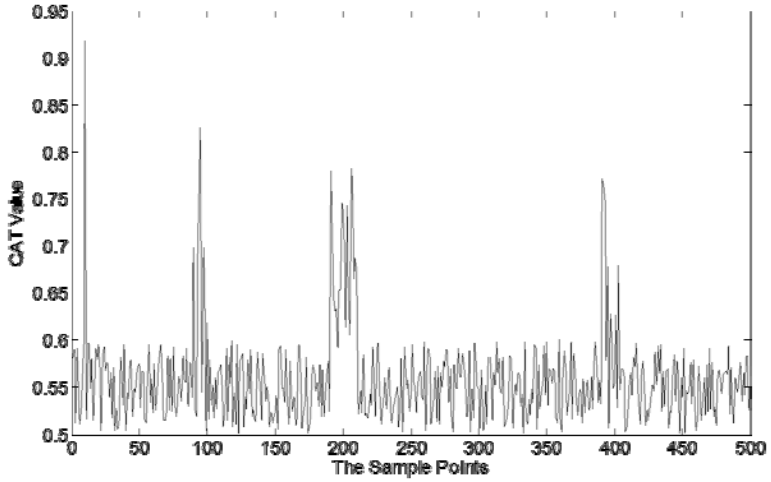
We conducted the simulation experiments to evaluate the efficiency of our detection approach. The experiments were divided into two groups: to detect flooding DDoS attacks and to detect DNS amplification attacks.

4.1. Experimental Environment and Model Parameters

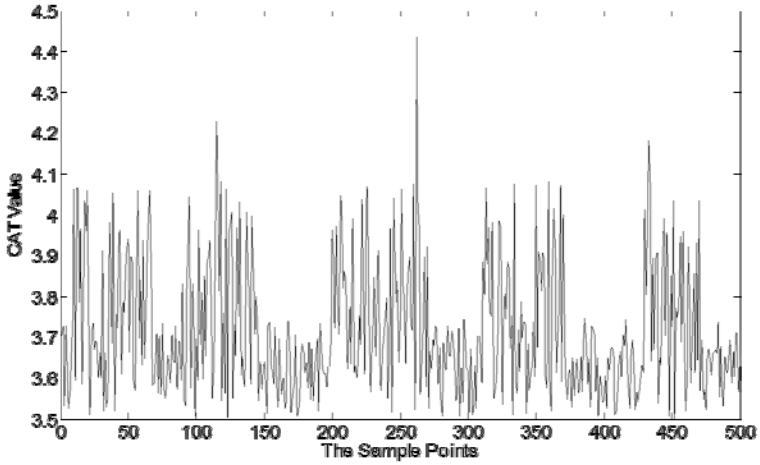
The experiment environment was established according to DDoS attacks as shown in Figure 1 and Figure 2. Five computers and a sub-network of our university involving 10 zombie machines were used to simulate the environment. One computer was used as an attacker to generate DNS requests. The software, NEMESIS-DNS, was installed on attacker computer to generate DNS attack, and it was able to perform 2000 times per second at most. Three computers are used as open recursive DNS servers. The last computer was used as the victim DNS server. Windows XP professional, VMware and Ubuntu Linux are installed on these computers to simulate DNS service. The hardware are with 2.6GHz intel core2 CPU, 8G RAM. The architecture for generating flooding DDoS attack is same as above, except for three open recursive DNS servers. In the experiments, the DNS normal traffic is added to simulate the real world network environment, which is collected from the DNS server of Changzhou University over a period of two months in 2012.

CAT time series is obtained by multiple sampling and calculation while the sampling interval Δt is 0.01s. As shown in Figure 3(a), CAT of normal traffic varies with the time and its mathematical expectation is 0.56. Figure 3(b) shows CAT of flooding DDoS attack and its mathematical expectation is 3.73. CAT of DNS amplification attack is shown in Figure 3(c) with mathematical expectation 2.71. Consequently, CAT time series are sensitive to two types of DDoS attacks; it can distinguish attack traffic from normal attack distinctly.

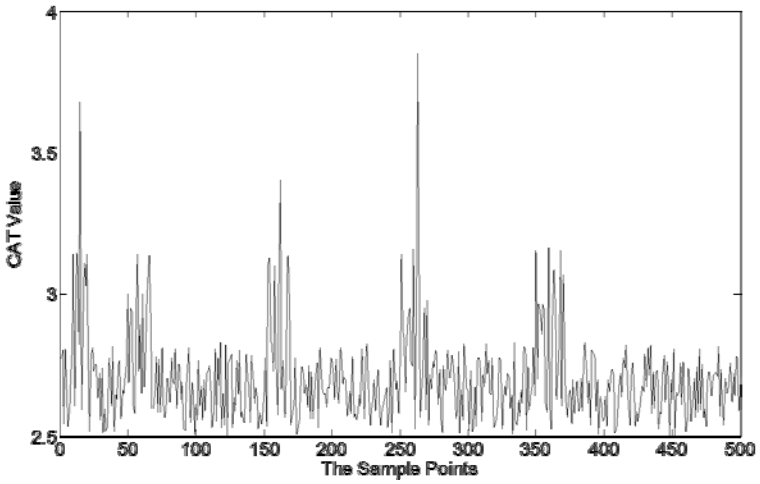
There are four parameters which may affect the CAT time series performances. The first is length of CAT, which is an important parameter in the system design. It adds detection time if it is too big; it disturbs the use of state changes in observations if it is too small. A standard DDoS attack lasts several minutes to several hours, and length of CAT is set as 50, so the detecting time 50*0.01=0.5(s) is appropriate for a DDoS attack. The second is parameter θ in Equation (3), which means the inverse proportion of two types of attacks in generation of ACT time series. Here θ is set as 0.5. The third is the degree of AAR model. In practice, the model degree is often fixed using some prior knowledge or guidelines. To optimize the goodness of fit vs. model complexity ratio, and also to ease the computational load, p is set as 4 which allowed the model to capture sufficiently well the normal traffic behavior. The forth is forgetting factor λ. It makes the filter more sensitive to recent samples, and it means more fluctuations in the filter coefficients. According to [13], λ is set as 0.99.



(a) CAT of a normal traffic



(b) CAT of a flooding DDoS attack



(c) CAT of a DNS amplification attack

Figure 3. CAT of different traffics (a) CAT of a normal traffic (b) CAT of a flooding DDoS attack (c) CAT of a DNS amplification attack

4.2. Evaluation Criteria

As detecting DDoS attack is a binary classification problem, the task is to learn how to classify unseen examples into one of two categories: positive categories and negative categories. True Positives (TP) means correctly classified attack traffic, True Negative (TN) means correctly classified normal traffic, False Positive (FP) means wrong classified normal traffic as attack traffic, and False Negative (TN) means wrong classified attack traffic as normal traffic. Commonly used performance metrics in classification problems are FPR, FNR, accuracy, precision and recall. They are defined as follows: The False Positive Rate (FPR) and the False Negative Rate (FNR) as the proportion of wrongly classified normal traffic and attack traffic respectively. Accuracy states the overall percentage of correct classified attack traffic. Precision as the classifier's safety, states the degree in which messages identified as attack traffic are indeed malicious. Recall as the classifier's effectiveness, states the percentage of attack traffic that the classifier manages to classify correctly. Receiver Operating Characteristic (ROC) as a classifier's balance ability between its FPR and its FNR is a function of varying a classification threshold. The corresponding functions are as follows:

$$FPR = FP/ (FP+TN) \quad (12)$$

$$FNR = FN/ (TP+FN) \quad (13)$$

$$Accuracy = (TP+TN)/ (TP+FP+TN+FN) \quad (14)$$

$$Precision = TP/ (TP+FP) \quad (15)$$

$$Recall = TP/ (TP+FN) \quad (16)$$

4.3. Experiment 1: Detect Flooding DDoS Attacks

The sampling interval Δt is 0.01s, and CAT time series length N is 50. The training sample sets of CAT are respectively gotten by sampling the normal traffic and attack traffic with attack rate from 100 to 2000 randomly. In order to test the robustness of our approach to the disturbance of normal traffic, we do five experiments called F-1, F-2,…, F-5, in which attack rate is fixed at 100 requests/s, 500 requests/s, 1000 requests/s, 1500 requests/s, 2000 requests/s with spoofing-based UDP packets. By mixing each group of attack traffic with normal traffic, the testing attack traffic of five groups in experiments is gotten, and normal traffic of the five groups is gotten correspondingly. By sampling and computing CAT time series respectively we can get five groups of testing CAT. In each experiment, normal traffic is collected 500 series; attack traffic is collected 300 series, and training data contains 60% of total dataset; testing data contains the rest of dataset.

The kernel function in SVM classifier is radial basis function (RBF) and the robustness of the classifiers is evaluated using 10-fold cross validation. Train SVM using the training sets, and carry out detection on testing sets. Moreover, in five experiments, the average training time is 5.21s and the average testing time is 1.49ms, it has enough detection speed in real-time.

Table 1. Performance of detecting flooding DDoS attacks

	Accuracy	FPR	FNR	Precision	Recall	ROC
F-1	93.35%	4.60%	4.27%	93.66%	94.51%	95.20%
F-2	97.44%	2.52%	3.03%	97.81%	96.57%	99.51%
F-3	99.19%	0.15%	2.27%	98.28%	97.13%	99.78%
F-4	99.41%	0.13%	1.85%	98.89%	97.85%	99.82%
F-5	100%	0%	0%	100%	100%	100%

The detection results are shown in Table 1. The FPR and FNR are reduced with the increase of attack rate, and the Accuracy, Precision, Recall and ROC are ascended with the increase of attack rate. The result indicates that our approach can accurately identity flooding DDoS attack and won't lead to high FNR when the traffic rate is very small.

4.4. Experiment 2: Detect DNS Amplification Attack

We do five experiments called A-1, A-2,…, A-5 in which attack rate is fixed at 50 requests/s, 100 requests/s, 200 requests/s, 400 requests/s, 800 requests/s. The cause of attack rate is much smaller than rate in experiment 1 is that sending requests by three open recursive DNS servers can generate a much larger response. Normal traffic is collected 500 series, and attack traffic is collected 300 series in each experiment. The training and testing method of SVM is as same as experiment 1. In five experiments, the average training time is 5.17s and the average testing time is 1.47ms.

The detection results are shown in Table 2. Our approach detects DNS amplification attacks with a high accuracy, precious, recall and ROC, while with a reasonable low FPR and FNR. When normal traffic is much larger than attack traffic, the detection ratio still keeps a high lever. The cause for FPR in two experiments is that the normal traffic is covered from different time periods and the detection results also depend on the quality of the training dataset. Furthermore, the traffic state shift will cause traffic random noise. The cause for FNR in two experiments is that when the attack rate is much smaller than normal traffic, the whole DNS traffic takes on normal state, which disturbs the extract of varied features about traffic state caused by attacks.

Table 2. Performance of detecting DNS amplification attacks

	Accuracy	FPR	FNR	Precision	Recall	ROC
A-1	97.33%	2.75%	3.39%	98.02%	98.50%	99.36%
A-2	99.29%	0.15%	2.18%	98.87%	97.79%	99.91%
A-3	99.77%	0.03%	1.78%	99.76%	98.21%	99.99%
A-4	100%	0%	0%	100%	100%	100%
A-5	100%	0%	0%	100%	100%	100%

5. Conclusion

In this paper, we proposed an efficient approach to detect DDoS attacks against DNS servers. This work provides two contributions: 1) CAT time series is defined to mix the basic characteristics of flooding DDoS attack and DNS amplification attack, such as the abrupt traffic change, spoofing-based UDP requests and flow dissymmetry. 2) A real-time detection scheme against DDoS attacks is proposed, and it can achieve high detection efficiency. In our future work, we will make a detail study of how to set all kinds of parameters in different application scenarios adaptively.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under contact (61070121).

References

[1] Beitollahi H, Deconinck G. Analyzing well-known countermeasures against distributed denial of service attacks. *Computer Communications*. 2012; 21(35): 1312-1332.
[2] Shao M, Sun B, Xie R. *The Research of DNS safe Analysis and Defense Technology.* Proceedings of 2012 International Conference of Intelligence Computation and Evolutionary Computation, 2012 Wuhan, China, 1009-1014.
[3] Guo F, Chen J, Chiueh T. *Spoof detection for preventing DoS attacks against DNS servers*. Proceedings of 26th IEEE International Conference on Distributed Computing Systems (ICDCS) 2006, Lisboa, Portugal: 1-37.
[4] Rikitake K. A Study of DNS Transport Protocol for Improving the Reliability, Ph.D. dissertation. School of Information Science and Technology, Osaka University, 2005.
[5] LI W, CHEN L. *Alleviating the impact of DNS DDoS attacks*, Proceedings of Second International Conference on Networks Security. Wireless Communications and Trusted Computing. 2010 Wuhan, China: 240-243.
[6] Vlajic N, Andrade M, Nguyen U. The Role of DNS TTL Values in Potential DDoS Attacks. *Procedia Computer Science.* 2012; 52(10): 466- 473.
[7] Kambourakis G, Moschos T, Geneiatakis D. *A fair solution to DNS amplification attacks.* Proceedings of International Workshop on Digital Forensics and Incident Analysis (WDFIA), 2007: 38-47.

[8] Lawton G. Stronger domain name system thwarts root-server attacks. *IEEE Computer*. 2005; 40(5): 14-17.
[9] Xie Y, Tang S, Huang X. Detecting latent attack behavior from aggregated Web traffic. *Computer Communications.* 2013; 36(8): 895-907.
[10] Kambourakis G, Moschos T, Geneiatakis D. *Detecting DNS Amplification Attacks*, Proceedings of 2nd International Workshop on Critical Information Infrastructure Security (CRITIS) 2007, Malaga, Spain: 185-196
[11] Sun C, Liu B, Shi L. *Efficient and low-cost hardware defense against DNS amplification attacks*. Proceedings of IEEE Global Telecommunications Conference. 2008.
[12] Viinikka J, Debar H, Méb L. Processing intrusion detection alert aggregates with time series modeling. *Information Fusion*. 2009; 23(10): 312-324.
[13] Sun Q, Zhang D, Gao P. Detecting Distributed Denial of Service Attacks Based on Time Series Analysis. *Chinese Journal of Computers.* 2005; 28(5): 767-773.
[14] Paxson V. Bro: A System for Detecting Network Intruders in Real-time. *Computer Networks*. 1999; 31(23): 2435-2463.
[15] Yan R, Zheng Q, Li H. Combining Adaptive Filtering and IF Flows to Detect DDoS Attacks within a Router. *KSII Transactions on Internet and Information Systems*. 2010; 4(3):428-449.
[16] Vapnik VN. *Editors*. The Nature of Statistical Learning Theory. New York: Springer. 1995.
[17] J Platt. Sequenital minimal optimization: A fast algorithm for training support vector machines. *Editors.* Advances in Kernel Methods - Support Vector Learning. MIT Press, 1998.