Detecting App-DDoS Attacks Based on Marking Access and d-SVDD

LI Jin-ling, WANG Bin-qiang
National Digital Switching System Engineering & Technological R&D Center
Zhengzhou , China
zifenglingsworld@163.com

***Abstract*---In order to enhance the extensibility of current attack feature extracted and detection means for App-DDoS(Application Layer Distributed Denial of Service, App-DDoS)attacks, a novel feature extracted method based on marking access and a new detection algorithm named d-SVDD are proposed. After expressing kinds of App-DDoS attacks as characteristic vectors by access marked strategy and feature extracted strategy, d-SVDD algorithm is used for secondary classification and detection of pre-set area around decision boundary based on SVDD. It is proved by experiments that the proposed feature extracted and detection means can realize effective detection for kinds of App-DDoS attacks, both have satisfying time, space and extensibility performance.**

Keywords-App-DDoS attack; Marking access; d-SVDD; Anomaly detection

I. INTRODUCTION

Being different only in purpose from normal behavior, App-DDoS attacks can easily cross the low-level defense systems for traditional DDoS attacks, along with the fact that dealing with a high-level application request is much more complex, finding out effective detection and defense means for App-DDoS attacks becomes more and more important[1].

Currently, most detection methods for App-DDoS attacks are mainly based on behavior analysis[2] and log analysis[3]. One typical detection method based on users' browsing information detects HTTP flooding attacks according to users' browsing order and the relationship between view time and page information[4]. Another detection method proposed by Xie Yi and Yu Shunzheng introduces HsMM model to detection algorithm[5][6]. In literature [7], App-DDoS attacks are divided into three categories by session parameter: request flooding attacks, asymmetric workload attacks and repeated one-shot attacks. According to this classification, a session suspicious degree model is proposed for anomaly detection and filtering.

After analyzing the detection methods above, we can conclude: 1) App-DDoS attacks have lots of different types because of the differences among application layer services and protocols, while means based on behavior analysis and log analysis only consider Web server mostly, the extracted characters have poor extensibility; 2) Anomaly detection methods deeply depend on extracted characters, so the poor extensibility of characters directly affects the extensibility of detection methods.

In order to enhance the extensibility of extracted characters and corresponding detection algorithms, achieve effective detection for various attacks, a novel feature extracted method based on marking access and a new detection algorithm named d-SVDD are proposed in this paper. The flow chart of such detection algorithm is shown in Figure 1.

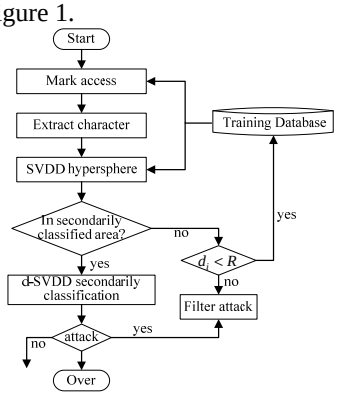

Figure 1. The flow chart of detection algorithm

The detection algorithm is divided into training and detection phases in detail. In training phase, mark normal users' access behavior with no marking strategy firstly, and then set marking period t_s and average interval between consecutive access stamps t_d according to the initial marked results. After that process the initial access stamps with marking strategy. When all of these are completed, feature extracted method is applied to obtain effective detection characters, and SVDD algorithm is used to get normal users' SVDD hypersphere. All of the results will be saved in training database and get fully prepared for the next detection phase.

II. FEATURE EXTRACTED METHOD BASED ON MARKING ACCESS

In detection phase, every user accessing to the server will be marked with marking strategy, including marking access time and access page. According to different servers, different requests will be selected to mark. For example, HTTP GET requests will be marked in attacks against Web server, while DNS requests will be marked during attacks against DNS server.

Take attacks against Web server for example, access time marked method adopts strategy as follow: if the current stamp interval $t_d^{'}$ is less than t_d , 1 will be assigned to current stamp, otherwise 0 will be assigned. Access page is marked using the following strategy: if the page requested by current user doesn't exist in protected Web

Published by Atlantis Press, Paris, France.

server, 2 will be assigned, otherwise no-treatment will be taken. Access marked strategy for attacks against DNS server is only slightly different from Web server in access page marked strategy: if the domain name requested doesn't exist in current server cache, and no answer received after a recursive query, mark current page as 2, otherwise, do nothing.

Obviously, the marked results can not be used as detection characters, it's necessary to obtain effective characters by suitable feature extracted strategy. The effective characters should meet the following conditions: 1) the extracted characters can fully reflect the time and space distribution of users' access behavior; 2) the characters will change obviously when attack occurs.

Take attacks against Web server for example, the character extracted strategy is described as following:

Input: users' marked results during t_s

Output: the character vector of user *i*

Method:

For user *i* during t_s :

1) Count the total number of stamps 1 and 0, expressed as s_{ic} ;

2) Count the maximal number of continuous stamp 1, expressed as s_{i1} ;

3) Count the maximal number of continuous stamp 0, expressed as s_{i0} ;

4) Count the total number of stamp 2, expressed as s_{i2} ;

5) Calculate the ratio of s_{i2} to s_{ic} , expressed as $p_{i2} = s_{i2} / s_{ic}$;

For all the users during t_s :

6) Calculate the entropy of all the requested pages, expressed as H ;

Then user *i* can be expressed as character vector $C_i =< s_{ic}, s_{i1}, s_{i0}, p_{i2}, H >$, where s_{ic} expresses the total requests of user *i* during t_s , s_{i1} and s_{i0} express the view time of user *i*. Under normal behavior, s_{i1} and s_{ic} are smaller while s_{i0} is larger than attack time. p_{i2} expresses the frequency of forged requests launched by user *i*. which is very effective for detecting Forged-URL Flood attacks. *H* expresses the distribution of users' interest, which is larger under attack time.

Access marked and feature extracted means are not limited to specific server, which can also be useful for detecting attacks against other servers via slightly modifying corresponding strategies. Take attacks against DNS server for example, access users can be expressed as character vector $C_i =< s_{ic}, s_{i1}, s_{i0}, p_{i2} >$ .In summary, for different servers, although there exist differences between access marked strategies and character extracted strategies, undeniably the similarity between them brings us more surprise, which enhance the extensibility of extracted characters perfectly.

III. d-SVDD SECONDARY CLASSIFICATION AND DETECTION ALGORITHM

A. SVDD classification algorithm

Considering the workload and purity of obtaining kinds of App-DDoS attack samples in training phase, our paper adopts a classification algorithm named Support Vector Data Description, SVDD for short[8], in which only normal samples will be needed. In SVDD, normal samples $c_i \in R^d (i = 1,2,\cdots,l)$ will be mapped to high -dimensional space by Φ (where $K(c_i, c_j) = \Phi(c_i) \cdot \Phi(c_j)$,K is the kernel function selected), then a minimum hypersphere as decision boundary will be obtained, containing normal samples as many as possible. The center and radius of the sphere are denoted as C_0 and R .

The algorithm SVDD is transformed into solving optimization problem stated as:

$$\min R^2 + \frac{1}{vl}\sum_{i=1}^{l}\xi_i$$

$$\text{s.t} \quad \| \Phi(c_i) - C_0 \|^2 - R^2 \le \xi_i , \xi_i \ge 0, i = 1,2,\cdots,l \qquad (1)$$

where slack variable ξ_i represents the penalty associated with the deviation of the *i*th training sample outside the sphere, and $1/vl$ ($v \in [0,1]$, l is the number of samples)is a trade-off constant controlling the relative importance of each sample. The dual problem of equation 1 is:

$$\max \quad W(\alpha) = \sum_{i,j=1}^{l}\alpha_i\alpha_j K(c_i, c_j) - \sum_{i=1}^{l}\alpha_i K(c_i, c_i)$$

$$\text{s.t} \ \sum_{i=1}^{l}\alpha_i = 1 , \quad \alpha_i \in [0, \frac{1}{vl}], i = 1,2,\cdots l \qquad (2)$$

Solving the dual problem above, a small amount of samples with zero value of α_i are taken as support vectors, the center of hypersphere can be expressed as

$$C_0 = \sum_{i=1}^{l}\alpha_i \Phi(c_i) \qquad (3)$$

and radius *R* can be computed by utilizing the distance between C_0 and any support vector:

$$R^2 - (K(c_i, c_j) - 2\sum_{j=1}^{l}\alpha_j K(c_i, c_j) + \alpha^2) = 0 \qquad (4)$$

The final form of decision function is

$$f(c) = R^2 - [K(c,c) - 2\sum_{i=1}^{l}\alpha_i K(c_i, c) + \sum_{i,j=1}^{l}\alpha_i\alpha_j K(c_i, c_j)] \quad (5)$$

In order to find optimal number of support vectors, set -ting trade-off constant $1/vl$ and selecting kernel function *K* play an important role. The smaller the parameter *v* is, the more the samples contained in hypersphere.

We use the popular Gaussian radial basis function (RBF) as the kernel function, defined as

$$k(x_1, x_2) = \exp[-\frac{\| x_1 - x_2 \|^2}{2\sigma^2}] \qquad (6)$$

RBF kernel function[9] can map nonlinear samples to unlimited high-dimensional space with setting only one parameter σ , where σ determines the complexity of decision boundary.

Published by Atlantis Press, Paris, France.

B. d-SVDD secondary classification algorithm

False alarm rate and detection rate are taken to judge the performance of SVDD classification algorithm. In order to response normal users' requests as many as possible, good algorithm extensibility and low false alarm rate should be considered during getting SVDD hypersphere, which will lead to some decline in detection rate. In order to achieve a better balance between the false alarm rate and detection rate, guaranteeing high detection rate while ensuring reasonable extensibility, a new classification algorithm named division of Support Vector Data Description, short for d-SVDD is proposed. Based on SVDD, the internal and external space of hypersphere near the decision boundary will be classified secondarily, that is called secondarily classified area. K-means clustering algorithm will be used to realize sub-region segmentation of secondarily classified area and the abnormal degree of each sub-region will be calculated. For character vector in secondarily classified area, abnormal degree will be assigned according to specific sub-region, and the corresponding user's request behavior will be tracked for limited integer multiple of t_s. Finally, we can realize secondary detection for suspicious users by average abnormal degree during the tracking time. The d-SVDD algorithm is described as following:

Input: the sphere of SVDD, samples for detection

Output: anomaly detection results

Method:

1) During t_s, distance between C_i and C_0 can be calculated through $d_i = \sqrt{K(c_i,c_j) - 2\sum_{j=1}^{l}\alpha_j K(c_i,c_j) + \alpha^2}$. After setting the factor of secondarily classified area λ （$0<\lambda<1$）,the secondarily classified area can be expressed as $(1-\lambda)\cdot R \le d_i \le (1+\lambda)\cdot R$.

2) During training phase, samples in secondarily classified area will be clustered by K-means clustering algorithm[10]. The sub-region of isolated sample is a sphere with the sample itself as center, and following the principle that there are no overlaps with any other existing sub-regions. Take the rest space of secondarily classified area that doesn't belong to any existing sub-regions as the last region named blank region. We suppose there are *m* sub-regions after secondary partition.

3) If there are *n* samples in secondarily classified area and n_j samples in the *j*th sub-region, the abnormal degree of *j*th sub-region can be calculated according to the equation $f_j = 1 - n_j / n$, where $0<j<m$, the abnormal degree of blank area is defined as 1.

4) If distance d_i between C_i and C_0 is smaller than *R*, and the sample is out of secondarily classified area, it will be treated as normal user denoted as C_{ii}, the abnormally degree is set as 0. If $d_i > R$, and the sample is out of secondarily classified area, it will be treated as attack user denoted as C_{io}, the abnormal degree is set as 1.

5) If the current sample lies in the *j*th sub-region, it will be denoted as C_{is} with the corresponding abnormal degree set as f_j.

6) Track C_{is} for limited integer multiple of t_s, and calculate its average abnormal degree during tracking time. If the result is higher than threshold, the sample will be treated as attacker, otherwise it will be treated as normal.

Because of the fact that d-SVDD further classifies the suspicious samples in secondarily classified area, a good balance is obtained between algorithm extensibility and detection rate, which improves the detection performance greatly than SVDD algorithm and is more suitable for App-DDoS attack detection. According to step 4, we can filter the obvious attackers while anomaly detection, and mitigate the pressure of the protected server to be attacked.

IV. SIMULATION

In order to verify the effectiveness of character extracted method based on marking access and the detection algorithm d-SVDD, this paper builds Web server, DNS server test network environments respectively according to references [11] and [12]. Access data from 700 t_s to 200 t_s before the attack are taken for training, while the following 200 t_s normal data before attack are taken for testing. Simulate 120 t_s CC attacks against Web server and 60 t_s attacks against DNS server by CC attack software and DNS Abuer v1.0 respectively. Attack data are randomly injected into test data, and detected by the proposed method. The detection results are shown in table 1.

TABLE I. The detection results of CC attack and DNS Attack

attack type	judge standard	SVDD		d-SVDD	
CC attack	detection rate (Attack t_s /Total t_s)	78.2% (120/320)	81.3% （60/200）	97.2% (120/320)	98.0% （60/200）
	false alarm rate (Normal t_s /Total t_s)	6.8% （200/320）	5.6% （140/200）	1.02% （200/320）	0.91% （140/200）
DNS rebounding attack	detection rate (Attack t_s /Total t_s)	80.8% (60/260)	84.7% （30/200）	97.1% (60/260)	98.5% （30/200）

false alarm rate (Normal t_s /Total t_s)	7.9% (200/260)	6.3% （170/200）	0.96% (200/260)	0.83% （170/200）

There can be concluded from experiment results in table 1: 1)Both CC and DNS rebounding attacks have been detected effectively, which expresses relatively strong extensibility of access marked strategy and algorithm d-SVDD; 2) Under the same conditions, the detection performance of d-SVDD is obviously better than SVDD, this is because the decision boundary of SVDD is set without attack information, while the further detection for secondarily classified area by d-SVDD compensates this drawback well; 3) Both two algorithms show better detection performance for DNS rebounding attack than CC attack, which is mainly due to the matching degree between attack and extracted characters.

V. PERFORMANCE ANALYSIS

A. The space complexity of character extracted strategy based on marking access

There are only two states during marking access time: 1 or 0, so one bit is enough to express. Similarly, one bit is also enough to express the two states during marking access pages. Therefore, if the average stamps during t_s are 20, the corresponding space occupied is 40 bits, and 80 Mbits(=10Mbytes) are enough for marking 2×10^6 users. Thus, even there are lots of access users, the space occupied for marking is still very low.

B. The time complexity of algorithm SVDD and d-SVDD

When SVDD is used, there are two completely different processes in fact: training phase and detecting phase. For anomaly detection in our paper, we mainly concern the accuracy and completeness of training phase, the time complexity in acceptable range has little affection on algorithm performance, so we only concern the time complexity of detection phase. Most of the time consumed during detection phase is spent on calculating distance d_i and comparing it with R, because center C_0 and radius R are known, so the time complexity is O(1).

The time consumed during d-SVDD detection phase is mainly spent on k-means clustering and calculating abnormal degree, the time complexity of them are O(knt) and O(1) respectively, where k is the number of clusters, t is the number of iterations, n is the number of vectors for clustering, generally t<n, k<n. Because vectors in secondarily classified area are a small portion of the total vectors, so the time complexity of d-SVDD is acceptable.

In summary, the total time complexity is O(knt).

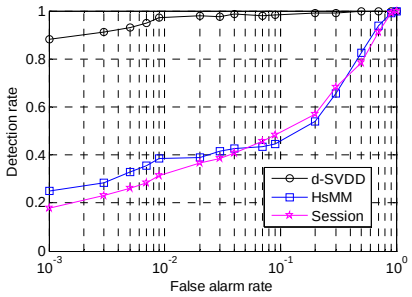

Figure 2. Comparison of ROC curves for DNS server

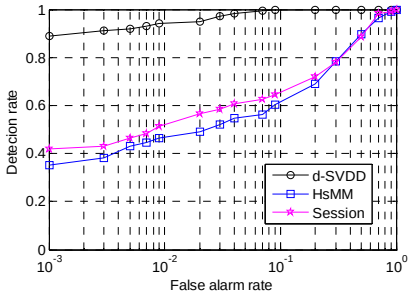

Figure 3. Comparison of ROC curves for FTP server

C. Comparison with other algorithms

Due to the relatively strong extensibility of character extracted method based on marking access and the detection algorithm d-SVDD, detection performance obviously extends that of Session model and HsMM model when detecting attacks against FTP and DNS attacks, shown in figures 2 and 3. We can see that d-SVDD algorithm has satisfying performance for all kinds of App-DDoS attacks, while Session model and HsMM model have poor detection performance for FTP and DNS attacks.

VI. CONCLUSION

In order to enhance the extensibility of extracted characters and detection algorithm, achieve effective detection for various App-DDoS attacks, this paper gives a new character extracted method based on marking access and an improved detection algorithm d-SVDD. Access marked and feature extracted means are not limited to specific server, which can also be suitable for detecting attacks against other servers based on slightly improving corresponding strategies. The algorithm d-SVDD compensates SVDD's drawback well through further detection for secondarily classified area, which greatly improves detection performance under the same training and detecting conditions. Due to the ideal space, time and detection performance of detection algorithm proposed in this paper, it will achieve satisfying detection performance for kinds of App-DDoS attacks.

Published by Atlantis Press, Paris, France.

ACKNOWLEDGMENT

This work is supported by National High-Tech Research & Development Program of China (No. 2011AA01A103).

REFERENCES

[1] V Durcekova, L Schwartz, N Shahmehri. Sophisticated Denial of Service Attacks Aimed at Application Layer[C]. ELEKTRO, Rajeck Teplice, 2012:55-60.

[2] Anuja. R. Zade, Suhas. H. Patil. A Survey on Various Defense Mechanisms Against Application Layer Distributed Denial Of Service Attack [J]. International Journal on Computer Science and Engineering, 2011, 11(3):3558-3563.

[3] DUAN Jian-li, LIU Shu-xia. Research on Web Log Mining Analysis[C]. International Symposium on Instrumentation & Measurement, Sensor Network and Automation, 2012:515-519.

[4] Yatahai T，Isohara T，Sasase I. Detection of HTTP-GET Flood Attack Based on Analysis of Page Access Behavior[C]. Proceedings of the IEEE Pacific Rim Conference on Communications，Computers and Signal Processing，2007：232-235.

[5] XIE Yi, YU Shun-zheng. Monitoring the Application-Layer DDoS Attacks for Popular Websites[C]. IEEE/ACM Transaction on Networking, 2009, 1(17):15-25.

[6] XIE Yi, YU Shun-zheng. A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors[C]. IEEE/ACM Transaction on Networking, 2009, 1(17):54-65.

[7] Ranjan S, Swaninathan R, Uysal M, Knightly E. DDoS-Shield: DDoS-resilient scheduling to counter application layer attacks[C]. IEEE/ACM Transaction on Networking, 2009, 1(17):26-39.

[8] Agrawal, P.K, Gupta, B.B, Jain, S. SVM Based Scheme for Predicting Number of Zombies in a DDoS Attack[C]. European Intelligence and Security Informatics Conference, Athens, 2011:178-182.

[9] ZHU Xiao-kai, YANG De-gui. Multi-Class Support Vector Domain Description for Pattern Recognition Based on a Measure of Expansibility[J]. Chinese Journal of Electronics. 2009, 3(37):464-469.

[10] Tapas kanungo, David M. Mount, Nathan S. Netanyahu. An Efficient k-Means Clustering Algorithm: Analysis and Implementation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 7(24):881-892.

[11] ZHI Jian. Research on DDoS Attack Based on The Application Layer. [Master dissertation], Dalian Maritime University, 2011.

[12] OU Shuai. Research and Design of Defense System Against DNS Distributed Denial of Service Attack. [Master dissertation], Southwest Jiaotong University, 2009.