Computer Science and Information Systems 13(2):659–676 DOI: 10.2298/CSIS160201021H

An Anomaly Detection on the Application-Layer-Based QoS in the Cloud Storage System

Dezhi Han[1], Kun Bi[1], Bolin Xie[2], Lili Huang[1], and Ruijun Wang[3]

[1] College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
dezhihan88@sina.com, kunbi@shmtu.edu.cn, 1710941959@qq.com
[2] College of information, Guangdong University of Foreign Studies, Guangzhou 510420, China
blxie@sina.com
[3] College of Electrical Engineering and Computer Science, University of Central Florida, Orlando, USA 48126
ruijunxx@gmail.com

Abstract. Attacks based on the application layer of the cloud storage system have been dramatically increasing nowadays. However, the present detection studies of attacks are mainly focused on the network and transmission layer instead of the application layer. In this paper, we proposed an anomaly attack detection method based on the hidden semi-Markov model (HsMM) to secure the cloud storage system from the application-layer-based attacks. In this proposed method, observation serials are constituted by the time intervals between the I/O requests made by normal users and their characterization using the hidden semi-Markov model based on each protocol for application layer. By applying this technique in the cloud storage system, it is able to effectively detect and correct their abnormal behaviors. In addition, to ensure the QoS(Quality of Service), a Priority Queuing and flow controlling module is proposed in this paper, which can allocate more I/O bandwidths and resources to normal users. Besides, the experimental results have shown that the proposed method can describe such normal I/O behaviors of users based on each protocol for the application layer in the cloud storage system with 99.2% higher detection ratio and 0.7% lower false positive ratio when detecting abnormal behaviors of users, and it can ensure the QoS for normal uses.

Keywords: cloud storage system, application layer anomaly detection, quality of service for I/O request, hidden semi-Markov model.

1. Introduction

Cloud storage is a system to ensure the data security and save the storage space through the functions such as clustered application, grid technology and distributed file system. The Cloud storage enable different types of storage devices to work together to provide data storage services and business access functions through applications. The users can connect to the cloud and access the data easily through any devices connected to the Internet [6].

At present, cloud storage technology has become a hot topic in the field of computer research. More and more companies are beginning to introduce this technology to build their own cloud storage platform and provide storage services for the enterprises and individuals. IDC claims that there are 4% of the world's IT expenditures spending for

cloud services and the proportion will reach 9% by 2012. Due to the cost and space limitations, data storage is very suitable for the use of cloud solutions and the proportion in the cloud services spending in cloud storage will increase from 8% to 13% [6]. Richard Villars, Vice President of the IDC Storage System and Implementation Strategy, points out that the costs for both public and private cloud storages would be up to $22.6 billion all over the world in 2015 [6]. At the same time, cloud storage has also brought some security issues, the attacks on the cloud storage system has shifted from the network layer and transport layer to application layer.

Traditional defense method such as firewall and intrusion detection system focuses on network layer access control and it is an important way to defend against network intrusion [9]. But it has limitations on application layer anomaly detection. Nowadays, application layer attack, such as phishing attack, HTTP flooding, has become a big security threat. At present, there are three main categories on anomaly detection in application layer. One is based on the load characters statistical method [11]; another is based on load keywords [7, 13]; the third is based on hidden Markov model [15–17, 20]. There are hundreds of applications using application layer protocol and there are various methods of attacking application layer. The above mentioned anomaly detection methods usually perform well on a specific class of application layer attack, but they are not applicable to all the application layer attack. Those methods usually consider little on the dynamic user behavior procedures and have some limitations on application anomaly detection.

This paper studies the characteristics of cloud storage system and proposes an application layer anomaly detection system for cloud storage system based on QoS mechanism. In this paper, the abnormal request sequence is detected by comparing the users I/O request sequences log likelihood probability with that of the normal users in cloud storage system using the Mahalanobis distance [13]. The normal users I/O requests qualities in the cloud storage system is guaranteed by using priority queuing and flow control methods and those mechanisms guarantee that normal users I/O requests can be allocated more bandwidth and resources when application layer attack occurs on the peak load in the cloud storage system.

The rest of this paper is organized as follows: section 2 is the related work; section 3 describes the components of anomaly detection system; section 4 describes the anomaly detection module; in section 5, the bandwidth allocation and flow control module is presented; section 6 describes the experimental results and analysis; section 7 is the conclusion.

2. Related Work

During the recent ten years, many methods can detect distributed system DDoS attacks, usually using rate control, time window, the worst case of threshold and pattern matching method to distinguish the normal operation and malicious behavior [3]. Detection method based on time series analysis, the method through the adaptive autoregressive model to obtain the multi-dimensional feature vectors of the user access behavior, and then using support vector machine to classify and recognize parameter vectors [4]. In [19], a detection method named VTP was proposed by Yang Xinyu, which could be used to calculate the hurst parameters in real time, and the DDoS attack could be judged according to the changes of the parameters. In [10], each page assigned a weight and established a his-

tory graph method for user access. However, for the real-time change of the dynamic web pages, the method is not very effective. Bolin Xie [14] proposed a request of the keywords application layer DDoS attack detection method based on the method using hidden markov model to describe the behavior of normal users, in a normal user within the unit time from the request of keywords frequency distribution and the number of requests as a model for the observed value, in order to detect application layer DDoS attacks. In [10], a stream correlation algorithm was used to distinguish the suspicious flow, this method involved to a set of routers package to compute and record the process, but in the actual backbone network, the router which is difficult to achieve.

In addition, there are a lot of other methods for detecting DDoS attacks [1,2,18,21,22]. For example, some detection methods used the characteristics of DDoS attack or burst flow, but some of the current complex attacks cant be detected when the attacker changes the attack plan. All of the above methods are just for the DDoS attacks on the network layer and transport layer, but cant effectively identify the DDoS attacks on the application layer. With the continuous development of network technology, in cloud storage system, the attacker usually selected attacks on application layer, but the existing methods cant effectively identify these attacks and application layer attack model, this paper proposed a DDoS attack detection method in cloud storage system application layer. This method compared of the user I/O request sequences log likelihood probability with that of the normal users in cloud storage system using the Mahalanobis distance to judge the abnormal request sequences [13], so as to distinguish the normal and abnormal users, and resolve cloud storage system application layer DDoS attacks using the corresponding defense measures at the application layer.

3. Components of Anomaly Detection System

Normal users accessing to the cloud storage system follow the similar process: authentication, retrieval, browse, read, write and delete. Retrieval is mainly to search for the desired information in mass storage pool. Browse is primarily on searching results or storage system document directory information. Read is mainly to download document information from the cloud storage system. Write is to upload files to cloud storage system. Delete is to delete the users or their own private directory of document information. From the process of the normal user accessing to the cloud storage system, the I/O flow of the cloud storage system has the self-similarity and the long-range dependence.

The anomaly detection system proposed in this paper is used in the front-end server cluster in the cloud storage system to guarantee the QoS of the cloud storage service. As shown in figure 1, the system includes the following four modules: application protocol recognition module, I/O request classification module, anomaly detection module as well as bandwidth allocation and flow control module. Application protocol recognition module is used to recognize the application protocol of the input packets; I/O request classification module is used to classify the users I/O requests based on the payload keywords containing in the application protocol packets; anomaly detection module is used to detect the anomaly of users I/O requests and mark different values on those I/O requests from the same sources according to the anomaly evaluation value for user behavior; bandwidth allocation and flow control module is used to allocate different I/O bandwidth and storage to different users according to the users I/O requests mark value computed in the anomaly

detection module. It puts different I/O requests into separate queues according to the mark value and different queues are allocated different I/O bandwidth and resources. In this way, it can guarantee the QoS of normal users I/O requests and correct the abnormal I/O requests.

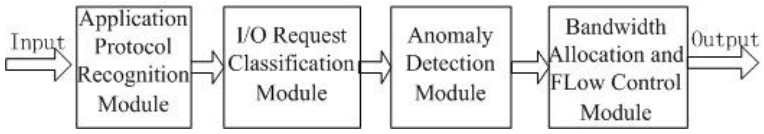

Fig. 1. Components of anomaly detection system

Application protocol recognition module uses the method proposed in paper [16] and it analyzes those protocols used in cloud storage system such as HTTP, FTP and etc. I/O request classification module analyzes the keywords from the application protocol and classifies the users I/O request according to those keywords with the method proposed in paper [16]. Those I/O requests in application layer mainly include authentication, retrieval, browser, read, write and delete, etc. Anomaly detection module together with bandwidth allocation and flow control module will be described in section 4 and section 5.

4. Anomaly Detection Module

4.1. Hidden semi-Markov Model

Hidden semi-Markov Model (HsMM) is developed from Hidden Markov Model (HMM). Different from HMM, HsMM introduces the state duration parameter. In HsMM, a state is corresponding to a serial of observations. The probability of transferring from one state to another is not only related to the current state, but also related to the duration time of the current state. Discrete HsMM model is usually represented by $\lambda = \{S, \pi, A, B, P\}$, and each parameter in HsMM is defined as follows:

(1) S is the model states set, $S = \{s_1, s_2, \ldots, s_G\}, s_g(1 \leq g \leq G)$is the possible state at time g, G is the total number of states in the model.

(2) π is the initial probability matrix, $\pi = \{\pi_g\}, \pi_g = Pr[q_1 = s_g], 1 \leq g \leq G$,and$\sum_g \pi_g = 1$. π_g represents the initial probability of the state in s_g at the initial time.

(3) A is the state transition matrix, $A = \{a_{gi}\}, a_{gi} = Pr[q_{(t+1)} = s_i | q_t = s_g], 1 \leq g, i \leq G, \sum_i a_{gi} = 1, q_t$ represents the state at time t, and a_{gi} represents the state transition probability from state s_g to s_i state at time t.

(4) B is the observation probability matrix, $B = \{b_g(v_k)\}, b_g(v_k) = Pr[O_t = V_k | q_t = s_g], 1 \leq k \leq K, 1 \leq g \leq G, o_t$ represents the observation value at time $t, b_g(v_k)$ represents the probability that the state is s_g and $O_t = V_k$ at time t.

(5) P is the state duration probability matrix, $P = \{p_g(d); 1 \leq d \leq D, 1 \leq g \leq G\}, p_g(d) = Pr[\tau_t = d | q_t = s_g]$ represents that the state is s_g at time t and will continue to stay in state s_g for the next d time slots.

The HsMM is trained by the normal users I/O requests sequences and the simplified Mahalanobis distance proposed in paper [13] is used to compute the difference between the HsMM model and. the observation sequences for actual cloud storage system.

4.2. Design Method

Users first have to log in with authentication when accessing the cloud storage system. After the Verification, the user can search, browse, upload, download or do other operations. When accessing the cloud storage system, the statistical characteristics of the normal user behavior have a certain similarities. For example, the I/O speed of normal users, search and browsing time, browsing process, upload process and download process have some similarity and the I/O requests for the same user have long-range dependency. Therefore, we can use those statistic characteristics to build a normal user model and the abnormal behavior of a user has a big difference from the statistical characteristics. For example, when a zombie host (BOT) accessing the front-end server initiates a HTTP request flooding attack, it randomly generates or repeats some simple I/O authentication requests. When a malicious user who steals a legitimate users account or posing as legitimate users of the application server, he can initiate I/O request flooding attack and send high frequency download or upload requests, so the number of the I/O requests and its frequency would be more than that of the normal users.

I/O request model of normal user accessing to the cloud storage system is shown in Figure 2. Figure 2 is divided into four groups and each group represents a state. The state ① is certification, state ② is document retrieval and browsing, state ③ is document upload, state ④ is to download the document. State ②, state ③ and state ④ can appear more than once and they also can appear alternately. Among them v1, v2, v3, v4, v5 and et.al are the I/O requests which are issued by the user and the user can send multiple I/O requests in each state. The q1 represents the time interval between v2 and v1, and a_{12} represents that the user is transferred from state ① to state ②.

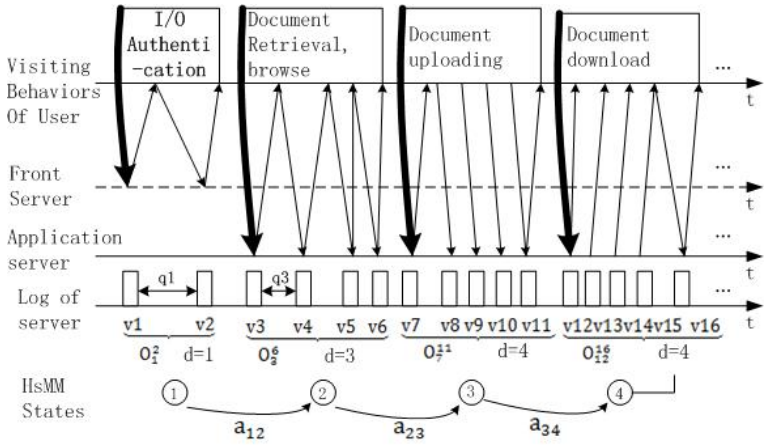

Fig. 2. I/O request model of the cloud storage system

The database records the time of each I/O request, and the source IP address together with destination IP address corresponding to each I/O request. Different IP addresses can be used to distinguish different users. The users I/O request sequence and the interval between two adjacent I/O requests can be computed by analyzing the users I/O request log information. User I/O request sequence can reflect the user I/O access behavior. Therefore, we get the I/O request sequence of the cloud storage system based on the log data and then get the observation sequence of the HsMM model of the user request behavior. The set of HsMM parameters can be computed from those user request observation sequences. Then we can compute the log likelihood probability (entropy) for any given observation sequence according to the HsMM. We use the log likelihood probability computed from those normal users to build the initial log likelihood probability distribution, and then calculate the log likelihood probability of the observation sequence of other online cloud storage users. The abnormal behavior can be judged by measuring the distance difference between itself and normal user with a simplified Mahalanobis distance, so as to recognize application layer DDoS attack on cloud storage system and other anomaly behaviors.

4.3. Model Building

When a large number of users access to a cloud storage system with a certain application layer protocol, the log data information stored in the log server could be used to describe this process. The I/O request observation sequences are similar in the statistical distribution when most users are normal users. For example, when a normal user access to the cloud storage system, the search process, the browsing directory, the duration time and the process of document upload and download have a certain similarity. Here we use the time interval between user I/O requests to train the model. The observation sequences generated by abnormal users have a lot of differences from normal users in statistical distribution. For instance, the access or download click frequency of normal users is ordinary, while abnormal users download frequency may be far greater than that of the normal users and the number of per unit time click to download files may be far more than a normal user.

Normal users behaviors will change when they use some kind of application layer protocol to access the cloud storage system, so the observation sequence will change at the same time. For example, when a normal user use some application layer protocol to access the cloud storage system, he can do upload, download, search and browse operations respectively. So the observation sequences will change as he does different operations. Those operating behaviors are the states shown in Figure 2. Suppose that normal users in the use of some kind of application layer protocol to access the cloud storage system have G different states, noted by $s_1, s_2 \ldots s_g$, The user behavior observation is corresponding to a given state. The value of G is between 6 and 10 for a cloud storage system. When the behavior of the user is considered as a Markov process, the future state is only related to the current state, and does not related to any past states. So the transfer relationship between normal users from one behavior to another behavior can be described by a Markov chain with G states. Let A represent the state transition probability matrix, its element $\mathrm{a_{gi}}$ represents the transition probability from state s_g to state S_i in time t when user access the cloud storage system,$S_g \in S$,$S_i \in S$, and $g \neq i$.

Assume that the user has K different I/O requests when accessing storage system and can be represented as $v_1, v_2, \ldots, v_k$ and we use number $(1, 2 \ldots K)$to represent them in

brief. Let P be the state duration probability matrix and its element $P_g(d)$ is the probability that a normal user in state S_g continuously produces d observations, in other words, it is the probability of producing d I/O requests, where$1 < d < D$ and D is the maximum state duration time and $\sum_d P_g(d) = 1.\pi$ is the initial probability matrix and its element π_g denotes the probability that the user state is $S_g$when the first request arrives at the front end server. When a user in the use of some kind of application layer protocol, let O_t be the t-th observation of the user recorded by the front end server, and it includes the time interval $r_t$which is the time interval between the t-th request v_t and the $(t-1)$-th request v_{t-1}, that means $O_t = (v_t, r_t)$, and $\mathrm{v_t} \in \{1, 2, \ldots, K\}$. Different numbers in v_t represent different kinds of operations, such as 1 represents of the retrieval, 2 represents the browse, 3 represents the upload, the 4 represents the download, and 5 represents the delete operation, etc. r_t is discrete integer value, that is $r_t \in 0, 1, 2, 3, \ldots$, and in this paper the time unit of r_t is s. Let $O = o_1, o_2, \ldots, o_T = o_1^T$ be a two-dimensional observation sequence whose length is T when a user uses some kind of application layer protocol to access the cloud storage system. Because there is no direct relationship between r_t and v_t, it is assumed that r_t and v_t are independent.

Let B be the observation probability matrix, and its element $b_g(k, d)$ represents the probability that the observation value is $v_t = k, r_t = d$ in a given state s_g , $1 \leq k \leq K, 0 \leq d.b_g(k, d)$ is defined as:

$$\begin{aligned} & b_g(k,d) = Pr[v_t = k, r_t = d | q_t = s_g] \\ & = Pr[v_t = k | q_t = s_g] * Pr[r_t = d | q_t = s_g] \\ & = b_g(k) * b_g(d) \end{aligned}$$

Note that $\lambda = \{\mathrm{S}, \pi, \mathbf{A}, \mathbf{B}, \mathbf{P}\}$ is the hidden semi-Markov model parameter set which represents the behavior statistical characters of normal users, and q_t represents the user state when the t-th observation is got in the front end server, and satisfy:
$\sum_{\mathrm{k}} \mathrm{b_g\,(k)} = 1$ and $\sum_{\mathrm{d}} \mathrm{b_g\,(d)} = 1$

4.4. Model Training

The I/O request sequences of normal users are collected at the front end of log server in cloud storage system. Those collected data are used as the training data for the HsMM and the parameters of the HsMM are computed from those training data. The trained HsMM is used to describe the dynamic user behavior. Because some abnormal data are also collected during the data collection process, in order to clarify the collected data and drop those abnormal data, a data filter process is required. The abnormal data filter process steps are as follows:

(1) For each observation sequence $O^{(h)}(1 \leq h \leq H), p(k)$ is the ratio of each observation value $v_k(1 \leq k \leq K)$ in its observation sequence. p(k) is computed by formula (1):

$$\mathrm{p}\,(k) = \frac{sum\left(v_k | O^{(h)}\right)}{\sum_{h=1}^{H} sum\left(v_k | O^{(h)}\right)} \tag{1}$$

(2) The information entropy is used to evaluate the distribution probability of observation sequences. The abnormal user requests will be filtered according to the information

entropy computing results. The information entropy is computed by formula (2):

$$\text{Entropy}\,(h) = -\sum_{\mathrm{k}=1}^{K} p\,(k) \log p\,(k) \tag{2}$$

Information entropy is used to compute the uncertainty of the information. When the information entropy computed by formula (2) is small, it means the users observation sequences are centralized. The result means some operations repeatedly appear in the observation sequence and it has a bigger probability that the observation sequence is abnormal. In contrast, if the information entropy is large, it means the users observation sequences are scattered and those sequences are probable from normal users. So the abnormal requests sequences can be filtered by computing the information entropy.

After the above filter steps, a set of normal requests sequences are received and we use H to represent the number of sequences in the filtered set and T^* is the length of the corresponding sequence. The basic processing flow chart of the algorithm in this paper is as follows. Firstly, it uses the information entropy method to get the normal users requests sequences and compute the parameters of the HsMM; then it uses an improved forward algorithm to compute the log likelihood probability; lastly, it uses the simplify Mahalanobis distance to judge the abnormal users behavior. Due to the computing of the likelihood of the users I/O requests sequences is only related to the forward computing in HsMM, the computing cost will be reduced. Thus it will accelerate the DDoS detecting speed in application layer. We use the improved forward algorithm proposed by Yu in [23] for training the HsMM proposed in this paper. In model training process, we first compute the mean value μ and the standard deviation δ of the initial log likelihood probability distribution, and then computing the parameters of HsMM. The concrete computing steps are described as follows.

Step 1. Compute the forward parameter $\alpha_t^{(h)}(g,d)$ of every user's observation sequence $O^{(h)}\ (1 \leq \mathrm{h} \leq \mathrm{H})$.$\alpha_{\mathrm{t}}^{(h)}\ (g,d)$ represents the probability that the state s_g stays for d time slots when the first t observations o_1^t got in the front end of the cloud storage, $1 \leq t \leq T^*$. The forward parameter is defined in formula (3).

$$\alpha_{\mathrm{t}}^{(h)}\ (g,d) = P_r[(o_1^t)_h, (q_t, \tau_t) = (s_g, d)] \tag{3}$$

Step 2. Compute the log likelihood probability P_h of each normal user's observation sequences using formula (4), $1 \leq \mathrm{h} \leq \mathrm{H}$. In the same way, the log likelihood probability P_{h^*} of other users observation sequences can be computed using formula (5). Then the log likelihood probability P_H of all normal users can be calculated using formula (6).

$$P_h = \ln\left(P\left(O^{(h)}|\lambda\right)\right) = \ln\left(\sum_{g=1}^{G}\sum_{d=1}^{D}\alpha_{T^*}^{h}\,(g,d)\right) \tag{4}$$

$$P_{h^*} = \ln\left(P\left(O^{(h^*)}|\lambda\right)\right) = \ln\left(\sum_{g=1}^{G}\sum_{d=1}^{D}\alpha_{T^*}^{h^*}\,(g,d)\right) \tag{5}$$

$$P_H = \prod_{h=1}^{H}\ln\left(P\left(O^{(h)}|\lambda\right)\right) = \prod_{h=1}^{H} P_h \tag{6}$$

In those formulas, G represents the total number of states in the model, D is the maximum state duration time, H is the total number of observation sequences, and $T^*$is the length of the corresponding sequences.

Step 3. Compute the mean value μ and the standard deviation δ of the initial log likelihood probability P_H of normal users. The parameters of μ and σ are computed using formula (7) and (8).

$$\mu = \frac{\mathrm{P}_H}{H} \tag{7}$$

$$\sigma = \sqrt{\frac{1}{H-1}\sum_{h=1}^{H}(P_h - \mu)^2} \tag{8}$$

Step 4 to step 7 use the method proposed in paper [15]. Lastly, the value of P_H is approaching to be stable.

4.5. Attack Detection

The model training phase described in section 4.4 can compute the normal user's initial log likelihood probability distribution and the log likelihood probability calculation formula. After that, the simplified Mahalanobis distance is used to compute the distance between the log probability of normal users and online user who is accessing the cloud storage system. The simplified Mahalanobis distance is defined in the formula (9):

$$\mathrm{d} = \sum_{h^*=1}^{H}\left|\frac{P_{h^*} - \mu}{\sigma}\right| \tag{9}$$

The value of d in the above detection reflects the abnormal degree of the behavior of a large number of users in the cloud storage system using some application layer protocol to access the cloud storage system. Here we can define a threshold for the user's normal behavior Q, when the value of d is close to Q, it is considered that the user's behavior is normal. When the user's behavior is a bit off Q, the user's behavior is considered to be a little abnormal. When the user's behavior is far from Q, we can determine the user is abnormal and an attack has occurred occurs on the cloud storage system.

In the experiment, we selecte $H = 16$ as the sequence length threshold, and the maximum time interval is $\Delta\mathrm{T} = 1800s$. Only when the observation sequence length of users is more than H, we calculate the average log probability of the sequence. In a front end server, if in excess of paragraph time ΔT it does not receive the user I/O request, we recounted the user sequence of observations, and it makes the observation sequence be better to reflect the user's current behavior. We follow the following steps to detect the abnormal situation of the user's behavior in the application layer accessing to the cloud storage system.

Step 1: when the user's first I/O authentication request reaches the front end server we can record the request v_1 and the arrival time I_0, and let t=1, r_0=0.Then it executes the second step.

Step 2: at the present time, if it is detected at the front-end server the user I/O requests, let t=t+1 and record the request v_t and arrival time I_t, and compute the time interval

$r_t = I_t - I_0$ from the request with a request v_{t-1} arrived at the front-end server, and finally let $I_0 = I_t$. If $r_t \geq \Delta T$, then go to the step 1; if $r_t \leq \Delta T$, then go to the third step.

Step 3: calculate the forward variable $a_t(g, d)$, if $t < H$, then go to step 2; if $t \geq H$, according to the formulas in section 4.4, calculate the average log likelihood probability P_{h^*} and the Mahalanobis distance d of the sequenc o_1^t using the parameters in λ , and go to step 2.

In the process of the above mentioned cycle, the behavior of the user application layer is reflected by the Mahalanobis distance d. When the abnormal behavior is detected, we can use priority queuing and traffic control measures to filter out the abnormal I/O request or inhibit various attack flow on peak time to resolve the DDoS attacks against the cloud storage system, and ensure the availability for the legitimate users in the cloud storage system. The corresponding detection and defense system is shown in figure 3. HsMM detector and filter located in the front end server can control the access speed in the cloud storage system at the peak time in the server priority queuing and flow control module.

5. Bandwidth Allocation and Flow Control Module

Our cloud storage system is based on Xen virtual platform and we add a bandwidth allocation and flow control module into the existed schedule module in Xen virtual platform using the method proposed in paper [12]. In this module, it uses a modified token bucket algorithm as the bandwidth control algorithm and builds a feedback mechanism to improve its precision. All the I/O requests with different mark value in the anomaly detection module will be putted into different queues. Firstly, those requests whose mark value is 3 will be discarded. Other requests with mark value 0, 1 and 2 will be putted into queue 1, queue 2 and queue 3 separately. Those I/O requests in queue 1 will be allocated more tokens; in other words, it will be allocated more bandwidth and more flow can be allowed to input into the cloud storage system. Those I/O requests in queue 2 and queue 3 will be allocated 1.25 percent and 0.25 percent tokens of that in queue 1 and the ratio also can be adapted according to the whole load in the cloud storage system. When the DDoS attack is detected or in the peak time of the system, the ratio is decreased to adapt those load changes. The control procedure is shown in figure 3.

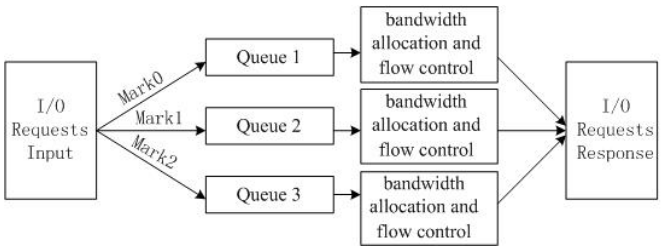

Fig. 3. Bandwidth allocation and flow control module

6. Experimental Results and Analysis

6.1. Off-line Test

In this section, we used log data from a university cloud storage system spread over a week (from 8:00 a.m on Monday to 8:00 p.m on Friday) to validate our anomaly detection algorithms. There are 4500-5000 users in the cloud storage system, we used information entropy to filter the request sequences with the anomaly data. Among the selected data, we randomly used two thirds of sequences (about 6600 sequences) within the first two hours to implement the HsMM training. The remainder data are used for testing. In our HsMM model, the states express I/O authentication, document retrieval, document browse, document download, document upload, online test, test submit, teaching content of feedback, online interaction for teachers and students, online jobs check, etc. In each state, the user may have multi-interactions with the cloud storage system, these interactions can be used as the observed value for the state.

The state distribution of the request sequences is shown in Fig.4. In the Fig.4, the most sequences contain five, six and seven states. In the Fig.5, the time of duration is the longest for the state 6 (online test) and the state 8 (the online interaction of teachers and students).

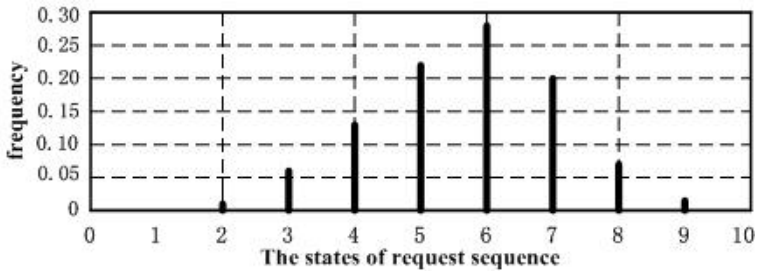

Fig. 4. The state distribution of request sequences

The DOSHTTP [8] is used to produce real HTTP Request FloodGET requests,and 4150 observation sequences is extracted. We tested the normal request sequences and application layer attack sequences (GET request), the log likelihood (entropy) distribution is shown in the Fig.6. These are significant differences in the entropy distributions between these two groups: the entropy for the most of the normal users is larger than -6, and is mainly between (-6, -1.9); but for the attack nodes, it is less than -6, and is mainly between (-8, -6). We can compute the threshold for the user's normal behavior Q and the value of Q is close to 1.32. When the Mahalanobis distance d is less or equal to the value of Q, it is considered that the user's behavior is normal. When the value of d is bigger than the value of Q, we can determine the user's behavior is abnormal. Therefore, the model can distinguish the attackers from normal users by their entropies or the Mahalanobis distance. The relation of the log likelihood (entropy) and the detection ratio (DR) is showed in the Fig.7. The relation of the log likelihood (entropy) and false positive ratio (FPR) is

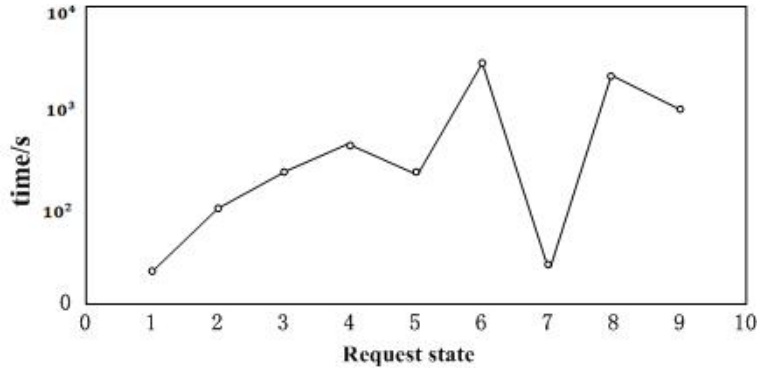

Fig. 5. The time distribution of states

showed in the Fig.8. Fig.7 and Fig.8 show that if we take -6.0 for the threshold value of average entropy, the detection ratio is about 98.5%, the false positive ratio is about 1%.

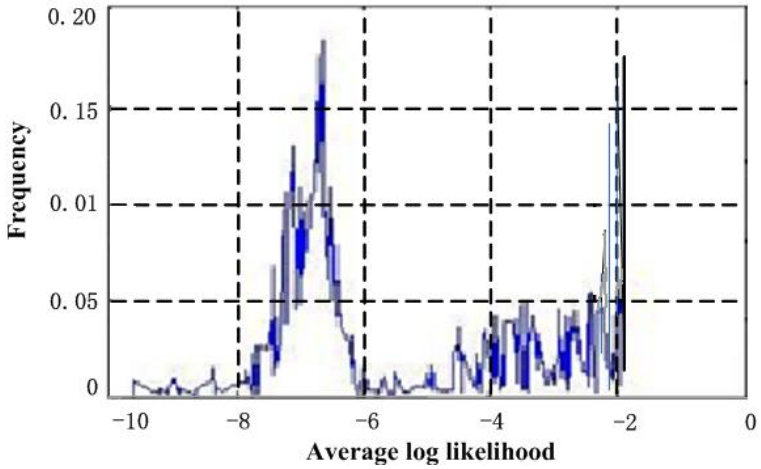

Fig. 6. Log likelihood distribution

6.2. On-line Test

General cloud storage system includes an export API for accessing the storage front end. This front end includes communication access point (CAP) and application access point (AAP). CAP accepts a variety of network protocols for the user's I/O access request and the request is authenticated and handed over to the AAP. AAP is a service scheduler which assigns different service requests to the cloud storage system of a single application server.

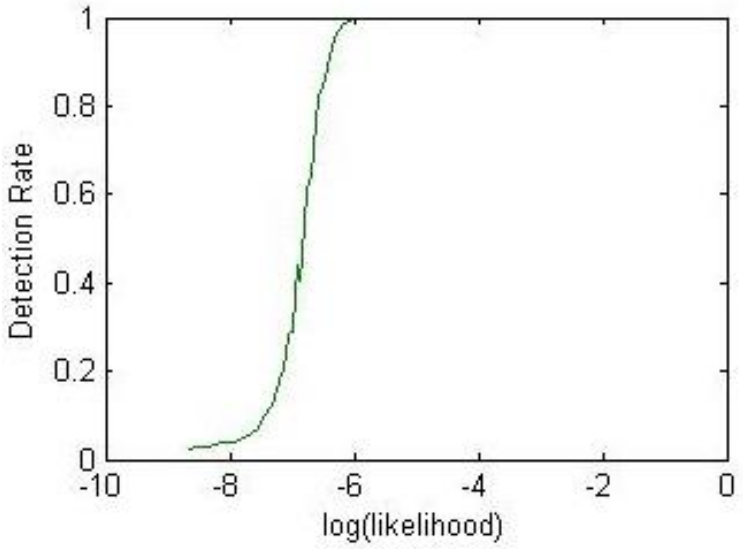

Fig. 7. Relation of DR and entropy

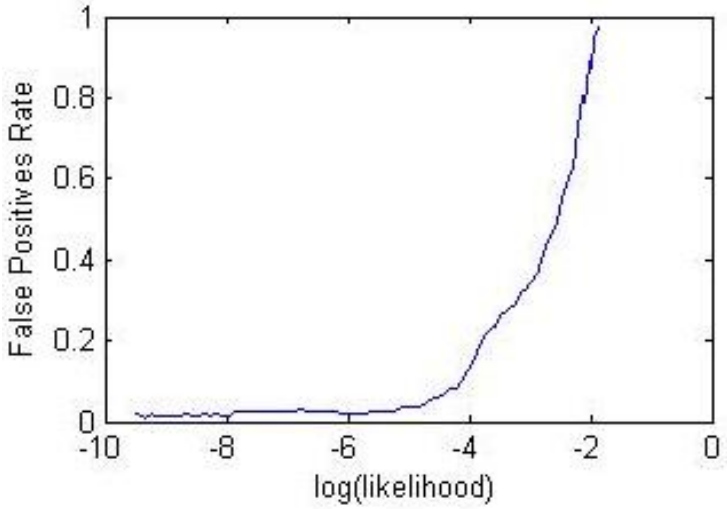

Fig. 8. Relation of FPR and entropy

AAP user is assigned an API (application program interface), and the user can access their own virtual storage system or public virtual storage system through the API.

In order to test the on-line performance of our system, we have tested the performance in a real network gateway in our campus and the topology is shown in figure 9. The front-end communication access point (CAP) servers are two Sun Fire X2250 with 2.8GHz CPU, 8GB memory and 1 TB hard disk. The front-end application access point (AAP) servers are a cluster of Lenovo Think Server RD550 with two 2.4GHz CPUs (6 cores per CPU), 64GB memory and 4 TB hard disk. The back-end storage system is 16 TB RAID5.

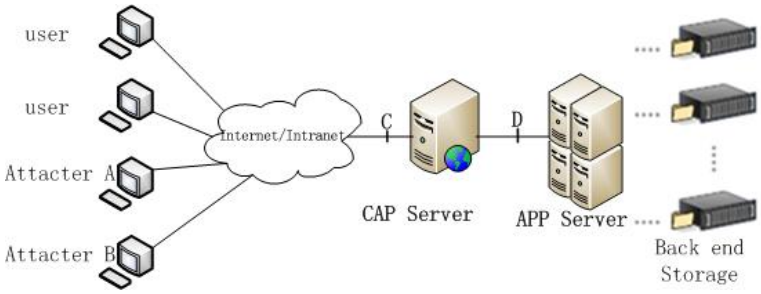

Fig. 9. On-line testing topology

In figure 9, the left side is 320 computers which located in four laboratories and there are two computers are attackers to generate application layer attack flow. In AAP servers, we set eight virtual machines with 2-core virtual CPU, 8GB memory, 1TB hard disk, 1000Mbps network adapter and centos 7.1. The total I/O bandwidth allocated to queue 1, queue 2 and queue3 are 400Mbps, 5Mbps and 1Mbps respectively. It makes sure that normal I/O requests can be responded and abnormal requests can be controlled and denied.

We use the HTTP and FTP I/O requests arrived at CAP servers to test the performance of our system, because the primary application layer protocols used in cloud storage system are HTTP and FTP.

We pick out one virtual machine in AAP servers and test its on-line performance. Firstly, we collect the normal users HTTP and FTP data in the gateway, analyze those data to generate the observer sequences and train our model using those data. After that, we test its on-line performance in two settings. One is without attack and the other is with application layer attack.

The on-line test lasts for 180 minutes and there is no attack flow in the first 60 minutes. From the 60^{th} minute, HTTP and FTP attacks are added into the test flow. Attacker A uses HTTP flooding attack which sends 120 requests per second to AAP server to send the same document. Attacker B uses FTP flooding attack and it sends 60 requests per minute to AAP server to download or upload the same file with the size of 20MB. Those attacks continue for 60 minutes. Those attacks stop at the 120 minute and in the last 60 minutes of the test, there is no attack in the system. During the tests, we have collected the HTTP and FTP data in C and D noted in Figure 9. After the test, we have analyzed the detection rate and false positive rate using the method in paper [16] and the results are shown in table 1.

Table 1. Detection ratio and false positive ratio

phases	detection ratio/%	false positive ratio/%
phase1(0m60m)	N/A	0.6
phase2(60m120m)	99.2	0.7
phase3(120m180m)	N/A	0.4

Next, we have tested the system performance of ensuring the quality of service (QoS) for normal users I/O requests when application layer attack occurs. We have selected a virtual machine in CAP server as the test virtual machine and use storage system I/O performance test software IOMeter [5] to test its storage I/O performance. We test the storage system I/O performance in CAP server with or without our system separately and the I/O performance test results are shown in figure 10. We can find that when the CAP server is without our system, the storage I/O performance has a significant decrease when the attack occurs; when the CAP server is with our system, the storage I/O performance does not change much during the attack. Because those abnormal I/O requests are putted into queue 2 and queue 3, it has little influence on those normal I/O requests in queue 1. As shown in figure 11, without our system, the CPU usage rate is obviously higher when the attack occurs, because handling those abnormal I/O requests consumes a lot of CPU resources; with our system, the CPU usage rate does not change much, because little abnormal I/O requests can get the handling chances and normal I/O requests load does not change much during the attack.

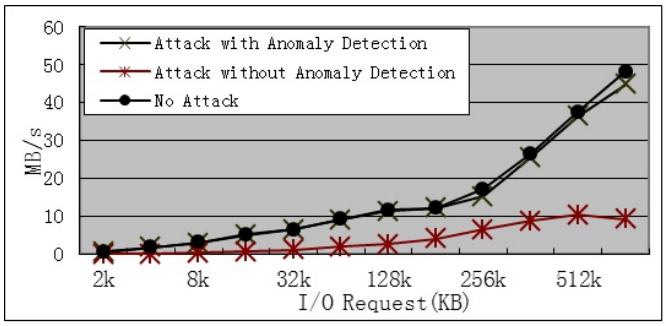

Fig. 10. The compare of Data transfer rate

7. Conclusion

In order to effectively detect the application layer attacks in cloud storage system, we proposed an Anomaly Detection on the Application-Layer-Based QoS in the Cloud Storage System.This method uses the time interval between user I/O requests and requests as the observations for training model, and it also uses an off-line way to train the hidden semi-Markov model(HsMM) parameters. The parameters of the HsMM can reflect the log like-

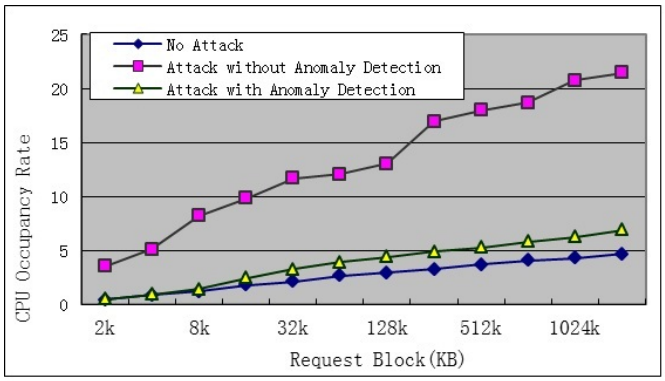

Fig. 11. CPU Occupancy Rate

lihood probability of the normal user I/O behavior and then compared the Mahalanobis distance of their log likelihood probability between normal users and online user who is accessing the cloud storage system to achieve anomaly detection. Because the calculation of ordinary users online log likelihood probability only relates to the forward computing process of the hidden semi-Markov model, it can greatly reduce the computation cost and help to improve the speed of the application layer anomaly detection. Simulation experiments show that (1) the entropy method can effectively filter out some abnormal user request sequence, and it can be used to extract the needed normal user request sequences for training the HSMM model; (2) the application layer anomaly attack detection method based on Hidden semi-Markov model has higher detection rate and lower false positive rate compared with other methods; (3) The bandwidth allocation and flow control method based on QoS guarantee the service quality of the I/O requests for normal users.

Acknowledgments. This work was partly supported by National Natural Science Foundation of China (No. 61373028 and No. 61070154), Science & Technology Program of Shanghai Maritime University (20130471).

References

1. Beitollahi, H., Deconinck, G.: Analyzing well-known countermeasures against distributed denial of service attacks. Computer Communications 35(11), 1312–1332 (2012)
2. Beitollahi, H., Deconinck, G.: Tackling application-layer {DDoS} attacks. Procedia Computer Science 10, 432–441 (2012), {ANT} 2012 and MobiWIS 2012
3. Ficco, M., Rak, M.: Stealthy denial of service strategy in cloud computing. IEEE Transactions on Cloud Computing 3(1), 80–94 (2015)
4. Gu, X.Q., Gu, H.Y., Ni, T.G., Ding, H.: Application layer real-time proactive defense system based on application layer protocol analysis. Journal of Computer Applications 33(8), 2228–2231 (2013)
5. IOMeter: (2006), [Online]. Available: http://www.iometer.org/doc/downloads.html
6. Jun, W.: Cloud storage technology advantages and its development trend (2014), [Online]. Available: http://www.cstor.cn/textdetail_6058.html

7. Lamport, L.: The temporal logic of actions. ACM Transactions on Programming Languages and Systems 16(3), 872–923 (1994)
8. socketsoft: (2009), [Online]. Available: http://www.socketsoft.net
9. Sun, C.H., Liu, B.: Survey on new solutions against distributed denial of service attacks. ACTA ELECTRONICA SINICA 37(7), 1562–1570 (2009)
10. Walfish, M., Vutukuru, M., Balakrishnan, H., Karger, D., Shenker, S.: Ddos defense by offense. ACM Trans. Comput. Syst. 28(1), 3:1–3:54 (2010)
11. Wang, K., Stolfo, S.J.: Anomalous payload-based network intrusion detection. In: Proceedings of The Seventh International Symposium on Recent Advances in Intrusion Detection. pp. 203–222. IEEE, Paris,France (2004)
12. Wang, X.B.: Disk I/O Bandwidth Allocation Mechanism of Virtual Machines. Ph.D. thesis, Dissertation for the Master Degree in Huazhong University of Science & Technology, Wuhan, China (2012)
13. Xie, B.L.: Research on Application-layer Anomaly Detection and Proactive Defense. Ph.D. thesis, Dissertation for the Doctoral Degree in Sun Yat-sen University, Guangzhou, China (2010)
14. Xie, B.L., Jiang, S.Y., Zhang, Q.S.: Application-layer ddos attack detection based on request keywords. Computer Science 40(7), 121–125 (2013)
15. Xie, B.L., Yu, S.Z.: Application layer anomaly detection based on application layer protocols keyword sequences. Journal of Computer Research and Development 48(1), 159–168 (2011)
16. Xie, B.L., Yu, S.Z.: Application layer real-time proactive defense system based on application layer protocol analysis. Chinese Journal of Computers 34(3), 452–463 (2011)
17. Xie, Y., Tang, S., Huang, X.: Detecting latent attack behavior from aggregated web traffic. Computer Communications 26(5), 895907 (2013)
18. Xie, Y., Tang, S., Huang, X., Tang, C., Liu, X.: Detecting latent attack behavior from aggregated web traffic. Computer Communications 36(8), 895–907 (2013)
19. Yang, X.Y., Yang, S.S., Li, J.: A flooding-based ddos detection algorithm based on non-linear preprocessing network traffic predicted method. Chinese Journal of Computers 34(2), 395–405 (2011)
20. Yu, J., Fang, C., Lu, L.: A lightweight mechanism to mitigate application layer ddos attacks. In: Social-Informatics and Telecommunications Engineering, Lecture Notes of the Institute for Computer Sciences, vol. 1281, pp. 175–191. Springer-Verlag, Berlin Heidelberg New York (2009)
21. Yu, J., Fang, C., Lu, L., Li, Z.: Scalable Information Systems: 4th International ICST Conference, INFOSCALE 2009, Hong Kong, June 10-11, 2009, Revised Selected Papers, chap. A Lightweight Mechanism to Mitigate Application Layer DDoS Attacks, pp. 175–191. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
22. Yu, S., Zhou, W., Jia, W., Guo, S., Xiang, Y., Tang, F.: Discriminating ddos attacks from flash crowds using flow correlation coefficient. IEEE Transactions on Parallel and Distributed Systems 23(6), 1073–1080 (2012)
23. Yu, S.Z., Kobayashi, H.: An efficient forward and backward algorithm for an explicit duration hidden markov model. IEEE Signal Processing Letters 10(1), 11–14 (2003)

Dezhi Han (corresponding author) received the Ph.D. degree from Huazhong University of Science and Technology. He is currently a professor of computer science and engineering at Shanghai Maritime University. His research interests include cloud computing, cloud security and cloud storage security technology.

Kun Bi received the Ph.D. degree from University of Science and Technology of China. He is currently a lecturer of computer science and engineering at Shanghai Maritime University. His main research interests include network security, big data and cloud security.

Bolin Xie received the PhD degree from the Sun Yat-sen University. He is a Charles N. He is currently a professor of computer science and engineering at Guangdong University of Foreign Studies. His main research interests include network security, big data and cloud security.

Lili Huang is currently a master degree candidate. His main research interests include network security and cloud security.

Ruijun Wang received her BS degree in Electrical Information Engineering from the University of Petroleum China (Beijing) in 2007 and received the M.S. degree in Information systems from the University of Central Queensland in 2009. She is currently working toward the PhD degree in the School of EECS at the University of Central Florida, Orlando. Her research interests include energy-efficient computing and storage systems.

Received: February 1, 2016; Accepted: May 27, 2016.