Prevention Of DOS & DDOS Attack Using Count Based Filtering Method In Cloud Computing

R. K. Yadav
Delhi Technological University
Computer Engineering Department
Delhi, India

Daya Gupta
Delhi Technological University
Computer Engineering Department
Delhi, India

Devendra Dadoriya
Delhi Technological University
Computer Engineering Department
Delhi, India

Abstract

Cloud computing is a technology that provides services on-demand. Freedom, compliances, data security, data recovery, auditing and availability are the security issues in cloud computing. Denial of service (DOS) and Distributed denial of service (DDOS) attack affect availability issue. In the DOS and DDOS attack attacker sends large number of TCP ACK packets to victim. Victim does not process packets CPU exhaustion of the victim. Attacker sends large number of TCP SYN packets and connections open on different-2 port number then memory exhaustion of the victim. The count based filtering method handle these attacks using count number of packets and find within timestamp.

Index Terms—Cloud Computing, Count Based Filtering, Distributed Denial of Service.

1. INTRODUCTION

Cloud computing is a recent technology, which is widely used for storing data remotely. It is used in educational and organizational both fields. Cloud computing provides services on software applications, programming platforms, infrastructure and data storage [1]. A user has a unique account for using each and every service on cloud and pay money according to using cloud services. It will distribute the resources to the user on the basis of need of the services. The Cloud Computing will do all the computation work for assign resources through software [2].

In the deployment model cloud have four types of cloud as follows: Public cloud, Private cloud, Hybrid cloud, and Community cloud [3]. Public cloud uses resources multiple customers and pay for service. It is public for all customers. It has less security because of publicly open. Private cloud uses some limited users. It is generally used managed data center [3]. It has more secure because of users are limited. Hybrid cloud is a combination of public cloud and private cloud. Community cloud creates by some set of organization and it has some standards and policies. Community cloud is operated by one of the organization in the community or third party. All the organization has trust on third party [3].

The rest of the organization of the paper is follows, section II describe cloud computing security issues. DOS attack and DDOS attack is discuss in III section. Section IV discus related work of this paper. Count based filtering method discuss in V section. Section VI discuss about performance analysis. Conclusion and future work discuss in VII section.

2. CLOUD COMPUTING SECURITY ISSUES

Recently, Cloud computing uses education and organizational areas. In the organization, cloud computing widely used for storing data on data centers and providing on-demand services. Data are storing globally then it has some security flaws. Users used services of cloud computing via internet through login id and password.

2.1 Data Security

In Cloud computing, where is the data more secure store in local machine or store in globally on data centers. If data store in local machine and machine does not connect to internet then it is secure. In the Berkeley paper's, If user want to store data on cloud then first user encrypt the data to own private key after that store data on cloud.

2.2 Freedom

In this issue, Cloud computing does not allow users to store data in its selected location. Cloud Service Provider (CSP) decides your data is storing which data centers [3]. Cloud computing allows users to retrieve data, store data and update data in cloud data centers.

2.3 Data Recovery

If customer lost the data due to some natural disaster, then customer ask to cloud service provider (CSP) how we recover data [4]. If CSP replicate the

data to another data centers then CSP provide data to other data centers. If CSP do not replicate data then ask to CSP how they provide data and it takes how much time.

2.4 Identification & Authentication

In Cloud computing, Identification is a process to identify user identity. Identify identity due to login id and password. Login id and password is valid for specific cloud. Cloud will allow services to user according to priority and permissions [5].

2.5 Availability

In present, Availability is a most popular security issue. In cloud computing SLAs defines server which types of services provide to customers. Availability means server will provide services all times. Availability can be affected temporarily or permanently. Denial of service attack affects the availability of cloud server [6].

2.6 Auditing

In the auditing check what happened in cloud. Auditing provides security of data and audit data periodically. Audit check store data is original data or modified data [7].

3. DOS & DDOS ATTACK

We are discussing aim of the attack and methods of the DOS and DDOS attack. DOS attack is one of the most famous attacks and DDOS attack is enhanced form of DOS attack. This attack has on end-systems.

3.1 Aim of DOS and DDOS attack

In the DOS attack, Attacker sends large number of packets over the network for consuming server resources and bandwidth. If any host sends a request to server for the resource but server does not entertained host and deny for the request. DOS attack has one-to-one mapping between attacker and victim and DDOS attack has many-to-one mapping between attacker and victim [8]. In the DOS attack, Attacker target to server memory, available CPU cycle, available disk space and maximum number of simultaneous connections [9].

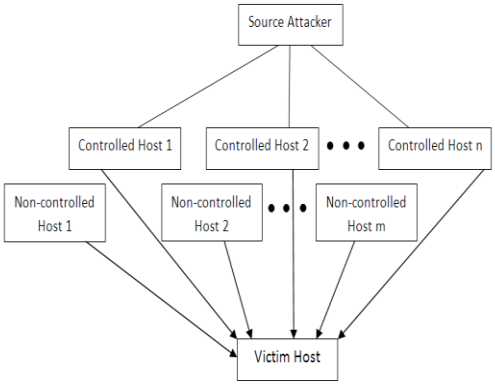

Figure 1. Architecture of DDOS attack [8]

In DDOS attack, Source attacker wants to DDOS attack to victim host. Source attacker control some specific n number (n = positive integer) of hosts and all n controlled hosts attack separately. Controlled hosts broadcast the packet for attack to victim host. Non-controlled hosts (m = positive integer) have not necessary all the hosts send a packet.

3.2 Methods of DOS and DDOS attack

In the DOS and DDOS attack, we discuss method of attack. How attack is done to end-systems. This attack is use flood in the network. In the flooding, source broadcast the packet in the network and destination address in the packet is victim address. Due to flooding victim receives multiple copies of the packet and source target victims packet processing capability [10].

3.2.1 ***TCP connection:*** In this attack, Zombie hosts creates the more number of three way handshake TCP connection to victim. These connections are creating until memory or resources are not exhausted [8].

3.2.2 ***TCP SYN flood attack:*** In this attack, Attacker sends a large number of TCP SYN packets to victim. Victim receives the packet and reply TCP SYN+ACK packet to attacker. Attacker does not response to victim TCP SYN packet and victim is on TCP half-open state [9]. After timeout victim close the half-open connection. In this attack attacker target the memory exhaustion of the victim [9]. If attacker creates more TCP half-open connection before closing previous half-open connection by victim then memory exhaustion problem occur.

3.2.3 ***TCP ACK flood attack:*** An attacker sends large number of TCP ACK packets to victim. Victim process all the packets. This attack affects on CPU exhaustion [9]. Victim process large number of packet and CPU processing is slow. In this attack, Victim suffers from livelock [9]. Because incoming number of packets are increases and CPU does not do useful work due to more number of interrupts.

4. RELATED WORK

In recent years, DOS and DDOS attack is a wide area of the research. This area have existed many successful solution. Filtering of the packets can be done on different-2 types like: victim-initiated, path-initiated, and source-initiated [11]. Victim-initiated reduce the incoming traffic of the packet. Path-initiated drop the packet if packet is not coming from appropriate router. In the source-initiated, source are responsible for incoming packets are attack free.

In other approach, DDOS defense using SOA-Based Traceback Approach (SBTA) [12]. SBTA is approach for identify true source of the packet in the cloud computing. SBTA placed on virtual machine in the cloud so that it is flexible and scalable. In cloud traceback (CTB) [13] main aim to apply service oriented architecture (SOA) for identify true source. CTB use deterministic packet marking (DPM) algorithm. DPM use some fields of IP packet and mark packet. Mark packet use for traceback the route.

Confidence based filtering (CBF) [14] is other method for prevent DDOS attack. In this method find the correlation pattern of the incoming packets. CBF method working on two periods: attacking period, Non-attacking period. If non-attack period, Nominal profile generate based on the fields of the packets and calculate confidence. If attack period, Nominal profile generate based on the fields of the occurrences of the packet and calculate confidence. According to confidence packet will be accept or discarded.

5. COUNT BASED FILTERING METHOD

In this approach, we are trying to prevent DOS and DDOS attack to server. Firstly, we are discussing some keywords that using in this method. In this method we are maintaining four tables are table_r, table_q, and table_p.

Timestamp: Timestamp is a positive integer value. It is a time in millisecond. In single timestamp, we are updating one table_r. Timestamp value do not fix too small or vary large. If timestamp value is very small then most of the time of CPU taken computation work. If timestamp value is large then update in the rejection_table is too late. Each timestamp creates one table. At the i^{th} timestamp three table exist are i^{th} time (table_r), $(i-1)^{th}$ (table_q) and $(i-2)^{th}$ table_p.

$$\text{Timestamp} <= \text{cycle time}/3 \quad (1)$$

Cycle time: After this time sequence number will be repeat.

After first timestamp is field entry goes from table_r to table_q. Second timestamp is field entry goes from table_q to table_p and third timestamp is entry delete from table_p and rejection_table also.

Table_r, table_q and table_p: Attributes of the table is source port number, destination port number, sequence number and count. Count is doing number of packets that contain remain three attributes are same.

Rejection_table: This table contains attributes are source port number, destination port number, sequence number and count. This table contains all the entries of the table_q and table_p. If source port number, destination port number, sequence number are same in table_q and table_p then rejection_table contain addition of the count of the both table.

Field Attribute (FA) : < source port number, destination port number, sequence number >
It is a combination of these attributes.

$Field_{(i)\_table\_name}$: where i= tuple number of the specified table
$Count_{(i)\_table\_name}$: where i= tuple number of the specified table that contain count

This method we are counting the number of packets per timestamp. According to its count packets will be accepted or rejected. rejection_table will update after each timestamp. This algorithm works as follow:

```
1.  if ( ack_bit == 1 )
2.  {
3.      update ( Field_packet )  //update algorithm written in below
4.      if ( check( Field_packet ) ) //check algorithm written in below
5.          Process packet
6.      else
7.          Discard packet
8.  }
9.  else if ( syn_bit == 1 )
10. {
11.     update ( Field_packet )//update algorithm written in below
12.     if ( check( Field_packet ) ) //check algorithm written in below
13.         Process packet
14.     else
15.         Discard packet
16. }
17. else
18.     process packet
```

After this algorithm, we are discussing update($Field_{packet}$) algorithm. This algorithm updates the table_r.

```
1   if ( Field_packet == Field_(i)_table_r )
2       count_(i)_table_r = count_(i)_table_r + 1 (
    corresponding entry in the Field_table_r )
3   else
4   {
5       add new entry in the table_r
6       Count = 1    // corresponding entry
7   }
```

Now, we are discussing algorithm for check($Field_{packet}$). This algorithm returns true if packet is processed otherwise return false.

```
1   if ( isEmpty ( rejection_table ) )  // return true if
    rejection_table is empty
2   {
3       if ( ( Field_packet == Field_(i)_table_r ) && (
    count_(i)_table_r >1 ) )
4           return false
5       else
6           return true
7   }
8   else
9   {
10      if ( Field_packet == Field_(i)_rejection_table )
11      {
12          if ( count_(i)_rejection_table >1 )
13              return false
14          else
15              return true
16      }
17      else
18       {
19          if ( ( Field_packet == Field_(i)_table_r ) && (
    count_(i)_table_r >1 ) )
20              return false
21          else
22              return true
23      }
24  }
```

Finally, we are discussing algorithm for update rejection_table. This table update after completion of each timestamp. All the update in the rejection_table according to algorithm is as follows:

Algorithm for modify rejection_table.

```
1   if ( isEmpty ( table_p ) )     //table_p is empty
2   {
3       if ( isEmpty ( table_q ) )  //table_q is empty
4           table_q = table_r
5       else                        //table_q is not empty
6       {
7           table_p = table_q
8           table_q = table_r
9       }
10      while ( ! isEmpty ( table_r ) ) // copy all the
    entry of table_r to rejection_table
11      {
12          if ( Field_(i)_table_r == Field_(j)_rejection_table )
13              count_(j)_rejection_table =
    count_(j)_rejection_table + count_(i)_table_r
14          else
15              Add Field_(i)_table_r and count_(i)_table_r
    in the rejection_table
16          delete i^th tuple in the table_r
17      }
18  }
19  else                    //table_p is not empty
20  {
21      while ( ! isEmpty ( table_p ) ) // delete all the
    entry in the rejection_table which is in the table_p
22      {
23          if ( Field_(i)_table_p == Field_(j)_rejection_table )
24              count_(j)_rejection_table =
    count_(j)_rejection_table - count_(i)_table_p
25          {
26              if (count_(j)_rejection_table == 0 )
27                  delete j^th tuple in the
    rejection_table
28          }
29          delete Field_(i)_table_p in the table_p
30      }
31      table_p = table_q
32      table_q = table_r
33      while ( ! isEmpty ( table_r ) ) // copy all the
    entry of table_r to rejection_table
34      {
35          if ( Field_(i)_table_r == Field_(j)_rejection_table )
36              count_(j)_rejection_table =
    count_(j)_rejection_table + count_(i)_table_r
37          else
38              Add Field_(i)_table_r and count_(i)_table_r
    in the rejection_table
39          delete i^th tuple in the table_r
40      }
41  }
```

6. PERFORMANCE ANALYSIS

In this section, we are using some statistics to test this method results. We are testing this method in the environment of 2.66 GHz Intel core i5 processor and 4GB

memory. We are implementing this method in the cloud computing environment using cloudsim3.0.2 simulator. The cloudsim3.0.2 is a java library. For using this java library must have install jdk6.0 or above version and eclipse SDK. We are checking this method for denial of service attack and results calculated and compare these results to CBF method [14].

6.1 Simulation Conditions

In this paper, the average rate of arrival packets are 6000 to 7000 packets per second and arrival of packets rate is 22.33Mbps [14]. Sequence number contain 32 bit so that sequence space is 2^32. According to [15] if data rate is 100Mbps then cycle time is 5.4 minutes and 22.33Mbps is much smaller than 100Mbps. We are using value of timestamp is 100ms. It means after every 100ms rejection_table will be updated and table_r will be empty. Timestamp value should not be large or small. If it is large then table size will be large and if it is small then increase cpu computation. Initially table_p, table_q, table_r and rejection_table are empty. In this simulation probability of arrival of attacking packets are 0.7 to 0.8.

TABLE1. NATURE OF THE ATTACK

Attack Intensity	Attack Packets	Legitimate Packets	Total no of Packets
1x	5,326	1,674	7,000
5x	25,326	9,674	35,000
10x	51,345	18,655	70,000
20x	1,02,929	37,071	1,40,000

6.2 Simulation results

In this section, we compare the count based filtering method and CBF method. Timestamp value is 100ms. Table2 denotes the comparison of both methods.

TABLE2. COMPARISON OF COUNT BASED FILTERING AND CBF METHOD

Attack Intensity	Process Time in Seconds	
	Count based filtering	CBF
1x	0.263	0.332
5x	0.758	1.073
10x	1.623	1.919
20x	3.179	3.661

We are improve the results because of count based filtering method does less computation work compare to CBF method.

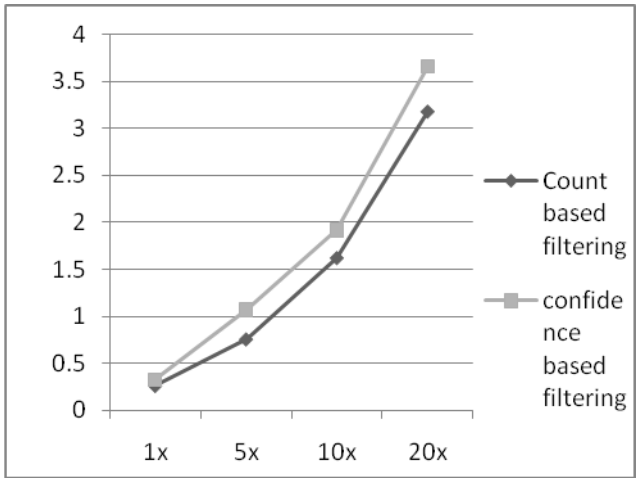

Figure 2. Comparison between confidence based filtering and count based filtering method

Result of count based filtering method may vary according to change the value of timestamp.

Since discarding or accepting a packet of CBF method first calculate confidence of packet and according to minconf packet will be accept or reject. Count based filtering method calculates count of packet and according to count packet will be accept or reject.

7. CONCLUSION AND FUTURE WORK

Cloud Computing is a recent technology that provides services remotely and users pay for service. Availability is a most important security problem. DOS and DDOS attack threats to availability. TCP SYN flood attack effect the memory exhaustion and TCP ACK flood attack effect the CPU exhaustion.

In this paper discuss count based filtering method. This method creates table and tables will be modify with the completion of timestamp and according to count of the same field packet will be processed or discarded. Performance of this method depends on the value of timestamp.

REFERENCES

[1] Dong Xu, "Cloud Computing: an Emerging Technology," in International Conference on Computer Design And Applications (ICCDA 2010) *IEEE*, vol. 1, pp. 100-104, 2010.

[2] Shufen Zhang, Shuai Zhang, Xubin Chen, and Shangzhou Wu, "Analysis and Research of Cloud Computing System Instance," in International Conference on Future Networks (ICFN 2010) *IEEE*, pp. 88-92, 2010.

[3] Jianfeng Yang, and Zhibin Chen, "Cloud Computing Research and Security Issues," *IEEE*, 2010.

[4] Krešimir Popović, and Željko Hocenski, "Cloud Computing security issues and challenges," *IEEE*, pp. 344-349, May 24-28, 2010.

[5] Ramgovind S, Eloff MM, and Smith E, "The Management of Security in Cloud Computing," *IEEE*, 2010.
[6] Huaglory Tianfield, "Security Issues In Cloud Computing," in International Conference on Systems, Man, and Cybernetics *IEEE*, pp. 1082-1089, October 14-17, 2012.
[7] Minqi Zhou, Rong Zhang, Wei Xie, Weining Qian, and Aoying Zhou, "Security and Privacy in Cloud Computing: A Survey," in International Conference on Semantics, Knowledge and Grids *IEEE*, pp. 105-112, 2010.
[8] An Lei, and Zhu Youchen, "The Solution of DDOS attack based on Multi-agent," in International Conference on Educational and Information Technology (ICEIT 2010) *IEEE*, Vol. 2, pp. 530-532, 2010.
[9] M. Handley, and E. Rescorla, "Internet Denial-of-Service Considerations," IETF, RFC 4732.
[10] W. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," IETF, RFC 4987.
[11] Yoohwan Kim, Wing Cheong Lau, Mooi Choo Chuah, and Jonathan Chao, "PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attack," in Transactions on dependable and secure computing *IEEE*, Vol. 3, No. 2, pp. 141-155, April-June 2006.
[12] Lanjuan Yang, Tao Zhang, Jinyu Song, JinShuang Wang, and Ping Chen, "Defense of DDoS Attack for Cloud Computing," *IEEE*, pp. 626-629, 2012.
[13] Bansidhar Joshi, A. Santhana Vijayan, and Bineet Kumar Joshi, "Securing Cloud Computing Environment Against DDoS Attacks," in International Conference on Computer Communication and Informatics (ICCCI-2012) *IEEE*, Jan. 10-12, 2012.
[14] Qi Chen, Wenmin Lin, Wanchun Dou, and Shui Yu, "CBF: A Packet Filtering Method for DDoS Attack Defense in Cloud Environment," in Ninth International Conference on Dependable, Autonomic and Secure Computing, *IEEE*, pp. 427-434, 2011.
[15] Jon Postel, "TRANSMISSION CONTROL PROTOCOL," IETF, RFC 793.