Lecture Notes in Electrical Engineering 376

Kuinam J. Kim
Nikolai Joukov
Editors

Information Science and Applications (ICISA) 2016

Springer

Editors
Kuinam J. Kim
Kyonggi University
iCatse
Seongnam-si, Kyonggi-do
Korea (Republic of)

Nikolai Joukov
Chair of IEEE CS STCOS
IBM T.J.Watson Research Center
Yorktown Heights, NY
USA

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-10-0556-5 ISBN 978-981-10-0557-2 (eBook)
DOI 10.1007/978-981-10-0557-2

Library of Congress Control Number: 2016930285

Printed on acid-free paper

This Springer imprint is published by SpringerNature
The registered company is Springer Science+Business Media Singapore Pte Ltd.

Synflood Spoof Source DDoS Attack Defence Based on Packet ID Anomaly Detection - PIDAD

Tran Manh Thang and Van Khanh Nguyen

Abstract A distributed denial-of-service (DDoS) attack characterized by flooding SYN packets is one of the network attacks to make the information system unavailable. This kind of attack becomes dangerous and more difficult to prevent and defense when attackers try to send flood SYN packets with spoof source, especially, there packets have information fields as the normal SYN packets. In this study, we propose a method called Packet Identification Anomaly Detection - PIDAD used to defense type of DDoS attack mentioned above. This method based on abnormal information of identification field in IP Header when observing the set of packets received in the victim system.

Keywords DDoS attacks · DBSCAN · PIDAD

1 Introduction

Distributed Denial-of-Service attack (DDoS) is a type of network attack that attackers make use of a large number of compromised computers to attack, which makes the applications, service or an information system unavailable. This type of attacking has become more and more dangerous and harder to prevent when the number of computers connected to the Internet, the weakness of information security and malwares get increasingly going up. There have been many researches on how to prevent and defense against this kind of attack; however, it is shown that indeed there have been no effective measures of defense and resulting in serious impact on agencies, organizations, and business during the recent time.

T.M. Thang(✉) · Van K. Nguyen
Department of Software Engineering, School of Information Technology and Communication, Hanoi University of Science and Technology, No. 1 Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam
e-mail: thang197@gmail.com, vannk@soict.hust.edu.vn

K.J. Kim and N. Joukov (eds.), *Information Science and Applications (ICISA) 2016*, Lecture Notes in Electrical Engineering 376,
DOI: 10.1007/978-981-10-0557-2_72

DDoS attack comprises of various forms of attacks, wherein, TCP SYN Flooding [1] is one of the attacking types that is of most difficulty to be prevented and defensed. As to this kind of attack, attackers aim at 3 steps of TCP handshaking phase to initiate new TCP connection. In this phase, attackers send TCP SYN packets as if to initiate a TCP connection with its victim. These SYN packets contain spoofed source IP addresses (hereinafter spoof packets), which cause the victim to waste resources that are allocated to half-open TCP connections which will never be completed by the attacker. Because spoof packets are sent as the normal SYN packets, so the victims cannot differentiate the actual SYN packets. To really create a large number of spoof packets, attackers can create malware (virus, worms) and then try to distribute them to as many devices as possible. Through which, attackers have created an botnet with infected devices which are controlled some C&C servers.

Attackers can use various methods of sending spoof packets, wherein 3 methods [4] popularly used are random spoof of any source address, random spoof of source address in a subnet and random spoof of source address in a pre-established list. The basic research question made here is how to find a way of identifying these spoof packets. Because, it can be found difficult to tell the difference between spoof and real packets when they may be completely the same. However, basing on some thorough observations, together with the conditions that attackers usually use, we can build the methods with specific effectiveness.

The primary idea of our method is following: Basing on the operation principles of TCP/IP protocol stack, we realize that value of Packet Identification field (PID) in IP Hearder will continuously increase by 1 unit for a packet was created and sent out from a computer. On the other hand, we find that the packets sent from the same computer are the packets sharing the same IP source and continuously increasing PID value. However, one computer can utilize many applications at the same time, working on many progresses in the meanwhile, thus on the receiving computer can we only oversee packets with interruptedly increasing PID value (i.e. it continues only for some values and then is interrupted). To illustrate more clearly, supposing that a computer sending with 3 progresses A, B and C is at the same time communicating with 3 other computers. Then, if we overlook the very computer, PID values stream is created continuously, but if observing PID values stream sent from the progress A (or B, or C) to some receiving computer, it is obviously to increase interruptedly. The above discovery shows that, we can detect spoof packets sent to a computer as packets with interruptedly continuously increasing PID values, but with random IP source address rather than the same source IP address.

To detect spoof packets used in DDoS attack, we study the method of grossing packets with different source addresses with continuously increasing PID value into the same Cluster using the algorithm DBSCAN [3]. As of each Cluster, we will define the expected value (Expected PID - EPID) of PID value each Cluster. Thence, any new packet send to the victim server will be considered as a spoof packet if PID value of new packet is equal EPID value in any Cluster.

In this paper, we will use some English specialized terms considered to be common in this area. In our opinion, using English terms with basic definition, common in international papers, will help specialized readers be easier to follow than translating into Vietnamese. We would like to state some English terms as below:

- DDoS: Distributed Denial-of-Service attack
- SYN: packet in TCP protocol stack used initially to establish connection
- PID: the value of Packet Identification field in IP Hearder.
- Cluster: cluster/ the nearby object in the sample space.
- Training phase: computer training phase
- Detection phase: detecting phase of spoofing packet
- Core poin: point/object inside cluster
- Border point: point/object on the border

2 Field Overview

As the overview research of DDoS attack [16], the solutions of defense DDoS attack are applied in two main directions: deployment location and point in time that defense takes place.

Applying method basing on deployment location is divided into 2 classes of defense: the first class is defense solutions to DDoS attacks occurring at the Transport layer of OSI model downwards (network/transport-level DDoS flooding attacks). The second class is defense solutions at application layer of OSI model. With the defense method basing on point in time that defense takes place location is divided into 3 phases: pre-attack phase, attack phase, and post-attack phase.

For the solution of defense against DDoS attack at network/transport-level is divided into categories: source-based, destination-based, network-based, and hybrid and for the defense solutions against application level is divided into categories: destination-based, and hybrid based on their deployment location.

In this study, we focus on studying the method of defensing DDoS attack type SYN flood spoofing source address. This is the solution applied for defensing against DDoS attack at network/transport-level mentioned above. We will analyze in detail some related researches as following:

Method of IP Traceback mechanisms [17-18] is the method of saving information of the root from source to victim in the Packet Identification information field of IP packets corresponding to each source IP address. Basing on this information about this route to detect the spoof packets. Spoof packets are packets of route information mismatched its source IP stored beforehand. However, this method has a big limit which requires mediate router to support the mechanism of marking the route, thus it is difficult to carry out in reality.

Method of Management Information Base (MIB) [19-21] is the method of combining information existing in each packet and linear information to detect

DDoS attack. When DDoS attack occur, this method compares the information in each packet of ICMP, TCP, UDP with the respective information which was analyzed and stored when DDoS attack had not occurred to find out the packets with abnormal information. Wang et al. [27, 28] proposed a method for detecting SYN flood attacks at leaf routers that connect end hosts to the Internet. Based on the observation that SYN and FIN packets form pairs in normal network traffic, they proposed using a nonparametric CUSUM method to accumulate the pairs.

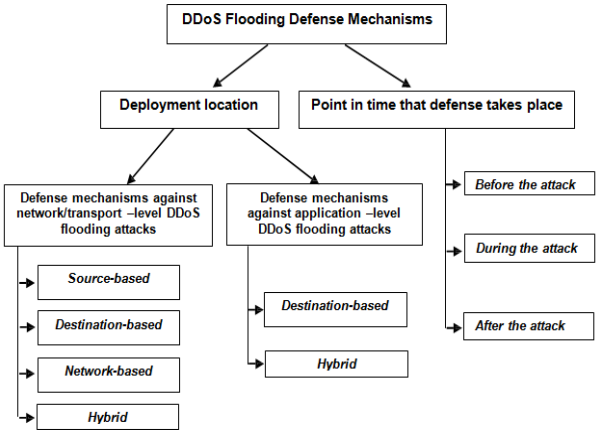

Fig. 1 Categorize the method of defensing DDoS attacks

Method of Packet marking and filtering mechanisms includes the following researches: History-based IP filtering [22], this method stores the information of source IP addresses which frequently connect to the system when DDoS attack has not occurred. When DDoS attack occurs, IP addresses existed in the list stored will be connected to the system; Method of Hop-count filtering [23], this method informs of source IP addresses as corresponding hop-count information in each packet which will be stored when DDoS attack has not occurred. When DDoS attack occurred, the packets with source addresses stored beforehand while information about hop-count different from information stored beforehand relating with that source IP addresses will be considered as spoof; Method of Path Identifier (Pi) [24], this method stores the values which are regarded as identifying the route of each packet when traveling from source to victim corresponding to each source IP address. The packets with the same route to the victim will share the same route information. This information will be used to filter all spoof packets of the same route with a spoof packet detected beforehand.

Method of Packet dropping based on the level of congestion wherein Packetscore [25] is a typical research for this method. Packetscore sets up priority level for each packet basing on the algorithm of Detecting-Differentiating-Discarding routers (3D-R) using Bayesian-theoretic metric. A barrier will be set up to filter the packets with low priority level when DDoS attack occurs and bring about stuck at Routers.

3 Theoretical Basis

3.1 Packet Identification (PID)

This is Identification field (16 bits) in IP Header [2]. This field is used to determine each packet sent by a computer. By default, the value of PID field will be increased by 1 unit when the computer sends out a packet. This characteristic has been used in the method of Idle Scan to check one port opened in server [26].

3.2 TCP Handshake and TCP SYN Flooding Attack

Transmission Control Protocol (TCP) [4] is protocol used popularly in the Internet to create a trustful connection oriented between any two computers in the Internet. Wherein, TCP handshake is a mechanism used to initiate a new TCP connection. TCP Handshakes can be described briefly as following: In the process of initiate new TCP connection, first, client sends a packet with SYN flag to server to require initiating new connection. After receiving the requirement, server will response a packet with SYN-ACK flags to inform the readiness for connection and require initiating new connection to client. The last step, client also responses a packet with ACK flag to server informing its readiness. Because this protocol has been developed from early time (when the modern Internet has not formed, the potential risks of Internet attacks have been unknown), it can be said that this connection principle is a dangerous weakness of information security that attackers can make use of it to perform different kinds of attacks.

The weakness here is: when receiving requirement of initiating new connection from client, immediately server reserve the necessary resources for new connection. Taking advantage of this loose, attacker will try to send overwhelm spoof packets and never complete TCP handshakes progress so that the server gradually depletes its resources.

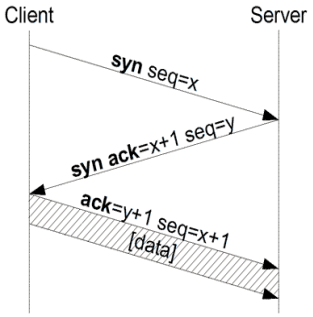

Fig. 2 TCP handshakes

Spoof packets are created by randomly spoofing source IP address and other information in each spoof packet is completely the same as normal packet.

This leads to: server cannot differentiate which connection packet which packet is the real or spoof one, so server will spend resources for such unreal connections. As the number of connection spoof packets is flooded sending, server has no longer had resources to serve real connections.

3.3 DBSCAN Algorithm

DBSCAN (Density-based spatial clustering of applications with noise) [29] is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996. It is a density-based clustering algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature.

The main idea of the algorithm is basing on bringing up *core point* term (core point, lying inside one cluster) - which has the number of neighbors reached standard level in a given neighboring radius - and the forming of clusters as spreading type based on the definition of connection or reachable destiny. In each cluster formed as DBSCAN, point objects are categorized into 2 types: *core point* (as the real object inside cluster) and *border point* (object lying on the border of cluster). We will present briefly summary of basic definitions and operative mechanism of DBSCAN algorithm as following.

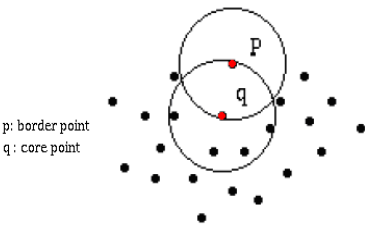

Fig. 3 DBSCAN Algorithm

<u>*Definitions*</u>

Neighboring radius area ε of object p, symbol $N_\varepsilon(p)$, is a collection of object q under condition that the distance between p and q, symbolized as $dist$(p,q), is smaller than given radius ε>0:

$$N_\varepsilon(p) = \{q \in \mathrm{D} \mid dist(p,q) \le \varepsilon\}$$

An object can be considered to have thick or thin neighboring density through the size of this collection $N_\varepsilon(p)$ We also define the term *reachable* density-based (directly density-reachable) from p to q: p regarded as (ε,m)-*reachable directly* with q if $p \in N_\varepsilon(q)$ and $|N_\varepsilon(q)| \ge m$ (m usually symbolized as *minPts* in the completed documents on DBSCAN). Generally speaking, this concept shows that

p can be reachable to *q* thanks to locating nearby and there are many mediate points within the neighborhood of *q.* The relationship *directly density-reachable* can be on one side or two sides (or symmetry). If it is two sides (p reachable to q and vice verse), both two points are remarked as core point, but if only *p* is reachable to *q*, q then is remarked as *core* and p is *border point*

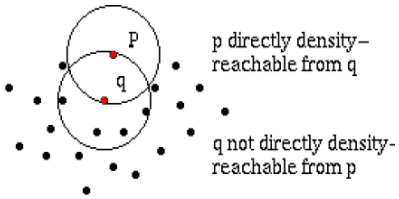

Fig. 4 Directly density-reachable Relationship

In overall, object *p* is considered (ε,*m*)-*reachable* with *q* if there is string p_1, p_2, ..., p_n (p_1 =q, p_n= p) existing while p_{i+1} directly density-reachable to p_i: $p_{i+1} \in N_\varepsilon(p_i)$ và $|N_\varepsilon(p_i)| \geq m$.

Object *p* connected as the density, symbolized as (ε,*m*)-connected, wit object *q* if there is object *o* existing while both *p* and *q* are density-reachable, i.e. (ε,*m*)-*reachable,* from *o*.

A cluster satisfies the standard of density (ε,*m*)-*dense* if all of its objects are (ε,*m*)-*connected* with one another and any object (ε,*m*)-*reachable* from some object of cluster is also considered to belong to cluster.

Algorithm

DBSCAN uses two parameters set up before which are ε and *m=minPts,* from which searching detecting (actually building and forming) *cluster*s under the condition of density (ε,*m*)-*dense*. Algorithm of DBSCAN begins from any object and then searches every object which is (ε,*m*)-*reachable* from object *p*. If *p* is *core point*, this task will generate a cluster satisfying (ε,*m*)-*dense*. If *p* is border object, it will be impossible to search an object density-reachable from *p*, then DBSCAN scan the next object in the database. Due to using the sharing parameter ε and *m* for all of the Cluster, DBSCAN can combine 2 clusters into 1 if these two clusters are nearby. The distance between 2 collections of S1 and S2, symbolized as *dist* (S1, S2) is the value of the smallest distance of any 2 objects *p*, *q* insides: *dist* (S1, S2) = min{*dist*(p,q) | p∈S1, q∈S2}. Two collections of objects in the same cluster can be separate if their distance between them increases.

4 PIDAD Method

PIDAD method we suggested is based on the main idea detecting strings of packets with their PID value continuously increasing but source addresses are random, without coincidence. As we mentioned above (Introduction part) the packets created and sent from the same computer will have PID value increase

continuously as short section; these continuously increasing sections make up the sections without coincidence and somehow interrupted. The reasons as stated are that in the same computer there can be many progresses having transactions connected to many other computers in the Internet. However, in TCP SYN flooding attack, attackers are supposed to create spoof SYN packets with spoof source IP address selected randomly. Therefore, we suggest a method to detect and filter packets, which have PID value continuously increasing short sections with different source IP addresses (due to random generation).

To realize this idea, we recommend using DBSCAN algorithm: each PID string with increasing short sections of different source IP addresses will be collected as a cluster as DBSCAN. With each cluster, we will determine E_{PID} value of each cluster, i.e. PID value is predicted to be sent continually. The process of forming clusters and find out E_{PID} in our method is called *Training phase*.

To detect the next spoof packets sent to the system, we will check PID value of each packet sent to see if there is coincidence with E_{PID} of any Cluster (list of E_{PID} collected through *training phase*); in case of finding PID value coinciding with any E_{PID} value, this packet will be the next spoof packet sent to the system. In case of unable to find PID value which coincide with any E_{PID} value, the packet sent can be real packet or the first spoof packet of a new Cluster (unformed in *training phase*).

We also suggest a technique using another restricted condition of time to build a new Cluster. Beside the phenomenon of PID continuously increasing short sections as mentioned above, normally the spoof packets sent from a minion computer in DDoS attack will be rather nearby and regular in terms of time (as it is necessary to create a large number of spoof packet without attention to responses from victim server). Thus, we suggest another condition of forming Cluster: in the time period of $T_{CC} = 2*T_{PC}$ there must be at least 2 coming packets sent and having full conditions to create new Cluster with the packet needed to be checked, then the new Cluster will be set up with the respective E_{PID} value. Wherein T_{PC} is the maximum time period of each Cluster needed to add a new member. If during the time period T_{CC} it is under conditions to set up new Cluster to the packet being checked, that packet will be considered to be real packet. The process of detecting next spoof packet sent to the system in our method called *Detection phase*.

As above, we have described the basic mechanism of the suggested algorithm; we would like to demonstrate each phase of the algorithm in detail as following.

4.1 PIDAD Training Phase

To have enough input sample for DBSCAN algorithm in training phase, first PIDAD method needs to collect N_{pc} initial packets with different source IP addresses sent to the system. After having N_{pc} packets, PIDAD will apply DBSCAN algorithm to find out Clusters which have packets with continuously increasing PID. To describe the characteristic of continuously increasing by 1 unit of each Cluster, DBSCAN algorithm chooses parameter ε (Eps) = 1 (the distance between 2 points in one Cluster). To describe the characteristic of the packets sent

from the same computer the value MinPts = 3 (this value has the meanings of at least 3 packets with continuously increasing PID creating one Cluster). In case the value MinPts = 2 the estimate PID value of this Cluster miscollected into other Cluster will be of high potential, which reduces the accuracy of the method.

```
For each Pi in MC
   if (Sp(Pi) is notcheck)
      set CorePoint = Pi;
      // Scan to get new member
      for each Pk (j < k < Npc)
         //check BorderPoint P(k)
         if ((TTC(Pj) – TTC(Pi) < TPC) and (PID(Pj) == PID(Pi) + 1))
            set CorePoint = Pj
            for each Pk (j < k < Npc)
               //check BorderPoint P(k)
               if ((Ttc(Pk) - Ttc(Pj) < TPC) and (PID(Pk) == PID(Pj) + 1))
                  Create new Cluster CT
                  add CT infor to MC
                  set CorePoint = Pk;
                  set EPID = PID(Pk) + 1;
                  add Pi,Pj,Pk infor to MPC
                  set Sp(Pi),Sp(Pj),Sp(Pk) = checked
               end
            end
         end
      end
end
```

Fig. 5 PIDAD algorithm in Training phase

Due to characteristic of directly density-reachable relationship ((ε,m)-reachable) of DBSCAN algorithm, whenever Cluster has new member (a new packet satisfied the conditions and belonging to that Cluster) the position of core point will be updated into the position of that new member (border point).

PIDAD uses a multi-dimension matrix M_C to be able to store information of Clusters including: value of Cluster ID, E_{PID} and the time Cluster add the last member (T_{LC}). Using M_{PC} to store the information of each packet sent to the system including PID, Cluster ID and S_P indicates the state of whether the packets checked become member of Cluster or not. Supposed P_i the packet i ($0 < i < N_{pc}$) which is checked of all conditions to bring into a Cluster, T_{TC} is the time to check packet P_i. Algorithm and algorithm flowchart in training phase are following:

4.2 PIDAD Detection Phase

After setting up Clusters and E_{PID} correspond from N_{PC} packets in training phase, PIDAD will transfer to detection phase, to detect the next spoof packets sent to the system. Each next packet sent to the system can be spoof packet or real packet. Spoof packets are packets with PID coincident with any E_{PID} value in M_C. The real packets are the packets satisfying both two conditions at the same time: no PID value coinciding with any E_{PID} value and under conditions to create one Cluster in

a period of time T_{CC} (after the time period T_{CC} the packets being checked cannot collect the next two packets to make a new Cluster). T_{CC} value set up is smaller than TCP timeout to allow PIDAD to determine the real packets before occurring TCP retransmission at Client side.

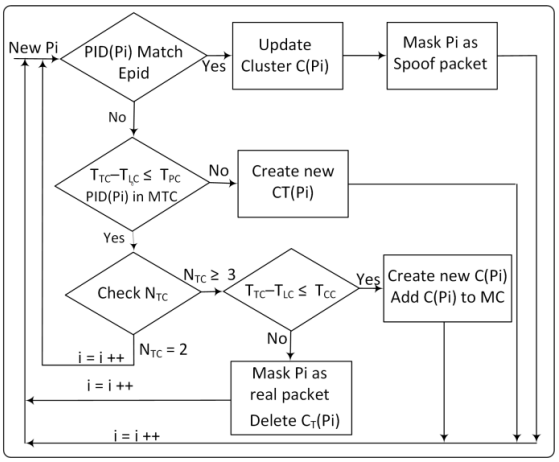

Fig. 6 PIDAD algorithm in Detection phase

To check if a packet has enough conditions to make a new Cluster or not, PIDAD method uses a temporary Cluster (C_T). Each packet sent to the system under the state of being checked will be assigned to be core point of C_T, after a time period T_{CC}, when C_T collects 2 border points C_T will be updated into new Cluster, or else C_T will be deleted. MT_C is used to store the information of C_T, the information needed to be stored of C_T as well as information of Cluster was set up in training phase. Supposed N_{PT} as the number of packets in each C_T.

In detection phase, to meet the ability of detecting spoof packets when having changes of attacking flow and minimizing the number of Clusters needed to be handled, PIDAD method should have updating method, deleting and adding new Clusters. Wherein, a Cluster is updated E_{PID} information when that Cluster add another new member. A Cluster will be deleted if after the time T_{CC}, that Cluster does not have new member added. A Cluster will be new created when after the time T_{CC} the packets being checked collect the next 2 packets with full conditions to set up a new Cluster.

Supposed P_i the packet number i being checked, C(p) is Cluster with p as core point. Algorithm in detection phase is described as the following figure:

5 Experiment Evaluation

The experimental results in this study, we take the sample of packets with real time in some real systems when meeting DDoS attack with the above type of attack. To evaluate the effectiveness of the method more accurately, we have

collected connection initial packets when the system work at normal mode and mixed randomly with the sample packets into experimental data. Experimental data are stored as file PCAP. Wherein the packets at normal functional mode and the sample packets during the attack will be marked to have basis of calculating the efficiency of the algorithm.

To collect SYN packets in file PCAP, we use the software Wireshack [14] to filter SYN packets in PCAP. To have information on the arrival time of the packets, and PID information from the file SYN packets collected, we use the software Tshark [15] to filter two information fields time_stemp and identification of each packet.

After having information of arrival time and PID of files of packets and programming the algorithm on the software Matlab, the experimental results show that our method can detect most of spoof packets sent to the system when meeting DDoS attack. The specific results are following:

Table 1 Experimental results of PIDAD

N_{pc}/T_{pc}	0.1 ms	0.2 ms	0.3 ms	0.4 ms
10000	86,2%	92,1%	84,1%	76,1%
20000	87,1%	91,1%	8,1%	77,1%
30000	88,1%	89,1%	94,1%	79,1%
40000	85,1%	95,1%	88,1%	80,1%

6 Future Works

In this research, we have studied and suggested a new method to detect spoof packets used in DDoS attack. The results show that our method has high rate of detecting spoof packets. However, in this research we still have to conduct manually some steps of solution in some parts such as: collecting initial packets with different IP in N_{PC}; the storing information of the packets was confirmed and connected initial successfully. To be able to complete and apply the method in reality, we anticipate to doing research on using Bloom filter algorithm suggested by Bloom [3] in the future study, this algorithm was used in some methods of preventing, defensing from DDoS attack [5, 6, 7]. Specifically, we will continue to study using Bloom filter algorithm to determine the first SYN packets sent to the system, to store information of connection initiated and store EPID value of each Cluster.

7 Conclusion

PIDAD method allows detecting and filtering connection initiate spoof packets used in DDoS attack typed flooding connection spoof packets. This method has advantages of allowing detecting spoof packets sent to the system without requiring Client side to resend connection initiate packets the second time. Basing on experimental results, it shows that this method can detect most of the spoof

packets in DDoS attack. However, in this research, we just focus on the method of detecting spoof packets without having recommendation of overall solution to completely apply in reality. In future studies, we need to do research using Bloom filter algorithm to complete and improve the troubleshooting efficiency of the proposed method.

References

1. CERT. TCP SYN Flooding and IP Spoofing Attacks. Advisory CA-96.21, September 1996
2. http://en.wikipedia.org/wiki/IP_header
3. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
4. Postel, J.: Transmission Control Protocol: DARPA internet program protocol specification, RFC 793, September 1981
5. Abdelsayed, S., Glimsholt, D., Leckie, C., Ryan, S., Shami,S.: An efficient filter for denial-of-service bandwidth attacks. In: IEEE Global Telecommunications Conference (GLOBECOM 2003), vol. 3, pp. 1353–1357, December 2003
6. Snoeren, A.C.: Hash-based IP traceback. In: Proceedings of the ACM SIGCOMM Conference, pp. 3–14. ACM Press, August 2001
7. Yaar, A., Perrig, A., Song, D.: Pi: A path identification mechanism of defend against DDoS attacks. In: IEEE Symposium on Security and Privacy, p. 93 (2003)
8. Yaar, A., Perrig, A., Song, D.: StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense. CMU-CS-02-208 (2003)
9. Chen,W., Yeung, D.Y.: Defending against TCP SYN flooding attacks under different types of IP spoofing. In: Fifth International Conference on Networking (ICN) (2006)
10. Changhua, S., Jindou, F., Lei, S., Bin, L.: A novel router-based scheme to mitigate SYN flooding DDoS attacks. In: IEEE INFOCOM (Poster), Anchorage, Alaska, USA (2007)
11. Chan, E., Chan, H., Chan, K., Chan, V., Chanson, S., et al.: IDR: an intrusion detection router for defending against distributed denial-of-service(DDoS) attacks. In: Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN 2004), pp. 581–586 (2004)
12. Wang, H., Jin, C., Shin, K.G.: Defense Against Spoofed IP Traffic Using Hop-Count Filtering. IEEE/ACM Trans. on Networking **15**(1), 40–53 (2007)
13. Peng, T., Leckie, C., Ramamohanarao, K.: Protection from distributed denial of service attacks using history-based IP filtering. In: ICC 2003, vol. 1, pp. 482–486, May 2003
14. https://www.wireshark.org/
15. https://www.wireshark.org/docs/man-pages/tshark.html
16. Zargar, S.T., Tipper, D.: A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks, February 11, 2013
17. John, A., Sivakumar, T.: DDoS: Survey of Traceback Methods. International Journal of Recent Trends in Engineering ACEEE (Association of Computer Electronics & ElectricalEngineers) **1**(2), May 2009

18. Joao, B., Cabrera, D., et al.: Proactive detection of distributed denial of service attacks using MIB traffic variables a feasibility study. In: Proceedings of Integrated Network Management, pp. 609–622 (2001)
19. Jalili, R., ImaniMehr, F.: Detection of distributed denial of service attacks using statistical pre-prossesor and unsupervised neural network. In: ISPEC, pp. 192–203. Springer-Verlag, Heidelberg (2005)
20. Li, M., Liu, J., Long, D.: Probability principle of reliable approach to detect signs of DDOS flood attacks. In: PDCAT, pp. 596–599. Springer-Verlag, Heidelberg (2004)
21. Peng, T., Leckie, C., Ramamohanarao, K.: Protection from distributed denial of service attacks using history-based IP filtering. In: ICC 2003, vol. 1, pp. 482–486, May 2003
22. Wang, H., Jin, C., Shin, K.G.: Defense Against Spoofed IP Traffic Using Hop-Count Filtering. IEEE/ACM Trans. on Networking **15**(1), 40–53 (2007)
23. Kim, Y., Lau, W.C., Chuah, M.C., Chao, H.J.: PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks. IEEE Trans. on Dependable and Secure Computing **3**(2), 141–155 (2006)
24. https://en.wikipedia.org/wiki/Idle_scan
25. Wang, H., Zhang, D., Shin, K.G.: Detecting SYN flooding attacks. In: Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM), vol. 3, pp. 1530–1539, June 23–27, 2002
26. Wang, H., Zhang, D., Shin, K.G.: Change point monitoring for the detection of dos attack. IEEE Transactions on Dependable and Secure Computing **1**(4), 193–208 (2004)
27. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (1996)