An New Efficient Cluster Based Detection Mechanisms for Distributed Denial of Services (DDoS) Attacks

K.Saravanan[1], R. Asokan[2]
[1]Faculty of Computer Science and Engg., Erode Sengunthar Engineering College, Thudupathi, India
[2]Principal, Kongunadu College of Engineering and Technology, Thottiam, India

Received: August 25, 2019. Revised: September 12, 2021. Accepted: October 14, 2021. Published: November 27, 2021.

Abstract: Cluster aggregation of statistical anomaly detection is a mechanism for defending against denial of service attack (dos) and distributed denial-of-service (DDoS) attacks. DDoS attacks are treated as a congestion-control problem; because most of the congestion is occurred in the malicious hosts not follow the normal end-to-end congestion control. Upstream routers are also notified to drop such packets in order that the router's resources are used to route legitimate traffic hence term cluster aggregation. If the victim suspects that the cluster aggregations are solved by most of the clients, it increases the complexity of the cluster aggregation. This aggregation solving technique allows the traversal of the attack traffic throughout the intermediate routers before reaching the destination. In this proposal, the aggregation solving mechanism is cluster aggregation to the core routers rather than having at the victim. The router based cluster aggregation mechanism checks the host system whether it is legitimate or not by providing a aggregation to be solved by the suspected host.

Keywords: DDoS (Distributed Denial of Service Attack),DoS, Cluster aggregation

I. INTRODUCTION

Resource consumption attacks, the most common type of network DoS seek to exhaust a server's resources, thus rendering the server incapable of providing its services to legitimate clients. Connection depletion attacks, a type of resource consumption attack, overwhelm a server by initiating a large number of connections and leaving them unresolved. The server then lacks the resources to service legitimate connection requests. Connection depletion attacks do not usually require special privileges because they exploit properties of the communication protocol itself. TCP SYN flood attack is a well-known example of this type of attack that takes advantage of a weakness in the TCP protocol that leaves connections unresolved.

Normally, a TCP connection is established through a three-way handshake. A client initiates a connection by sending a SYN packet to the server. The server acknowledges the request by sending a SYN ACK packet back to the client and allocating space for the connection in a buffer. The client then replies with an ACK packet, and the connection is completely established. In the TCP SYN flood attack, an attacker initiates many connections in which the SYN and SYN ACK packets are exchanged as usual, but the final ACK message is never sent to the server. Thus the connection is never completely established, and the server is left with buffer space allocated for all the incomplete connections. If the attack succeeds, the server fills up its buffer with incomplete connections, leaving no space for non-malicious connection requests, and thus preventing the server from establishing connections with legitimate (non-malicious) clients. According to the Computer Incident Advisory Capability,, the first DDoS attack occurred in the summer of 1999. In February 2000, one of the first major DDoS attacks was waged against Yahoo.com. This attack kept Yahoo off the Internet for about 2 hours and cost Yahoo a significant loss in advertising revenue. Another recent DDoS attack occurred on October 20, 2002 against the 13 root servers that provide the Domain Name System (DNS) service to Internet users around the world.

II. EXISTING SYSTEMS

- *TCP SYN Flood*

The TCP SYN flood is one of the most dangerous forms of the DDoS attacks. When two sides want to establish a TCP connection, the system, which asks for the connection (client), has to send initially a "SYN" message to the other system (server) in order to notify its intention. When the server receives the "SYN" message, it reserves some of its resources for the expected connection and sends a "SYN-ACK" message back to the client.

The reception of "SYN-ACK" message from the client triggers the transmission of a new message "ACK" which is sent to the server in order to complete the last stage of a three-step handshake protocol. After reception of the last message “ACK” from the server, the connection is successfully established and the two peers are able to start exchanging their data. If the server does not receive the “ACK” message from the other side then it discards the partially established connection and releases the set of resources reserved for the specific attempt. This three way handshake is shown in fig. 1. In order to keep a track of the requests, the server builds a backlog queue into its system memory. This queue maintains a limited number of half-open connections per port. Once the backlog queue limit is reached, the server discards every new connection request from the clients.

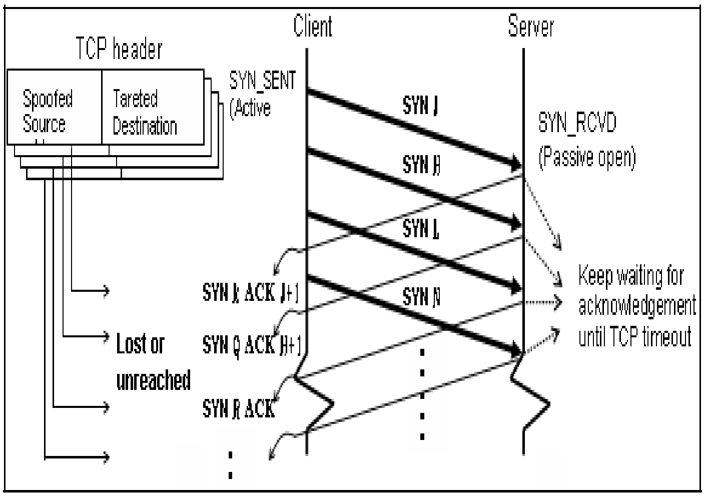

Figure 1 Three way handshake of TCP

However, until the SYN-ACK packet is acknowledged by the client, the connection remains in a half-open state for a period of up to the TCP connection timeout. That means that the server looses some of its resources for a specific amount of time. The adversary side sends successive waves of “SYN” messages to the target system by using connection request packets, which have spoofed source addresses of non-existent or currently inactive clients. Since the victim server never receives the final ACK packet to complete the three-way handshake, the net effect of each wave of inserted SYN messages is that the attacked side loses all its resources for a certain amount of time.

- *UDP Flood*

This is the second most popular DDoS attack method after TCP SYN flood. The basic idea in the UDP Flood attacks is to exploit UDP services, which are known to reply to the packets. The hacker is armed with a list of broadcast addresses, to which he sends spoofed UDP packets. These packets are sent to random and changing ports of the unsuspected target location. In most of the cases the packets are directed to the echo port 7 (echoes any character it receives in an attempt to test network programs) on the target machines.

However, there are attacks in which the malicious user sends packets to the chargen port. The chargen port is a port, which is used for testing purposes and generates a series of characters for each packet it receives. By connecting a host's chargen service to the echo service on the same or another machine, all affected machines can be effectively taken out of service as an excessively high number of packets are going to be produced. In addition, if two or more hosts are so connected, the intervening network can also become congested and deny service to all hosts whose traffic traverses that network.

- *Ingress /Egress Filtering*

Most of DoS/DDoS attacks use forged or spoofed source IP addresses in order to hide the attacker’s originality and also indirectly generate the massive traffic from the intermediary network to the target machine. As a result, a machine, that the spoofed address is belonging to, is also a victim of the attack. A packet leaving to Internet and arriving from Internet must have a source address originating from an interior network. By blocking packets with non-local source IP address from leaving a interior network, DoS/DDoS attacker’s source address spoofing become impossible. However, even though this scheme is most feasible in customer network, the universal deployment is not likely to be accomplished because of administrative burden, potential router overhead and complications with existing services that depend on source address spoofing. Moreover, if an interior network is quite large, or each sub-network does not have the address filtering capability, the attackers could still forge addresses from hundreds of thousands of hosts within a valid interior network.

- *IP Trace back*

Most existing traceback techniques start from the router closest to the victim and interactively test its upstream links until they determine which one is used to carry the attacker’s traffic. Ideally this procedure is repeated recursively on the upstream router until the source of traffic is reached. This technique has a critical assumption that an attacker will be remained while tracing mechanism is completed. To avoid the overhead of trace back, Burch and Cheswick proposed the possibility of tracing the flooding attacks by “marking” the packets, either probabilistically or deterministically, with the address of the routers they traverse. Therefore, the victim can use this marking information to trace an attacker back to its source.

- *Smurf*

The two main components to the Smurf Denial-of-Service attack are the use of forged ICMP echo request packets and the direction of packets to IP broadcast addresses. The Internet Control Message Protocol (ICMP) is used to handle the errors and exchange control the messages. Also, the ICMP can be used to determine whether a machine on the

Internet is responding properly or has connection problems. To do this, an ICMP echo request packet is sent to a machine with a return address that the contacted machine would return an ICMP echo reply packet when receiving the ICMP echo request packet. The most common implementation of this process is the “ping” command, which included with many operating systems and network software packages. ICMP can also be a valuable tool in diagnosing host or network problem, because it conveys status and error information, including the notification of the network congestion and of other network transport problems.

- *Traffic Shaping*

A number of routers in the market today have features to limit the amount of bandwidth that some type of traffic can consume. This is sometimes referred to as "traffic shaping”. This can be used in a proactive way if the traffic behaviour of the network is already known. It can also be used in a reactive way by crafting an access rule that would match some of the network traffic using by the DDoS attack. For example if the attack is employing ICMP packets or TCP SYN packets you could configure the system to specifically limit the bandwidth those types of packets. This would allow some of these packets that may belong to legitimate network flow to go through.

Because of the avalanche affect of the DDoS attacks for this option to be effective, it must be deployed as depth into the network as possible (closer to the source of the attack packets). Would the ISPs may be required to implement these filters in their routers. This would not be possible for many organizations for a number of reasons. Furthermore, DDoS attack tools can generate random packets such as that matching them with a set of access list. Rules can become difficult unless by using negative space while defining normal traffic pattern and assuming everything else is DDoS traffic.

- *Traffic Analysis*

A number of researchers in the academic fields have proposed different approaches to analyze the traffic patterns in order to infer the attacking packets and its characteristics. Most of methods are detecting the pattern of illegitimate packets or their source using probabilistic and statistic analysis. A critical point of those researches is that a large scale attacks can readily be identified by observing very abrupt changes in the network traffics and most of packets would have a certain type of pattern so that we can classify them according to each pattern.

For example, at first, randomly collect sample, classify the collected packets, and then normalize the data or build a temporary DB. Second, using a specific algorithm and modeling, find a pattern for bad-will packets from the data sampling. Most of differentiated methods are developed in second phase such as using Time series, Data Mining, Probabilistic Modeling and more complicated mathematical models. However, even though those methods can provide quite reasonable solutions to detect bad-will packets, we cannot be fully confident that every attacking packet can be detected or only illegitimate packets are likely to be detected since these mechanisms are relying on probabilistic model.

III. CLUSTERING OF TRAFFIC ANOMALY DDOS ATTACK

A network is connected with many number of sensor devices. The job of sensor event is to monitor the traffic in network. They are fed as input to the statistical traffic anomaly detection and k-means clustering algorithm is applied to the traffic anomaly detection so that the objects are grouped. In this way any anomaly DDoS detection whether identified as good node or bad node remains in the cluster.

- *ALGORITHM*

Our traffic anomaly detection approach deploys the K-mean clustering algorithm in order to separate the routing network with normal and anomalous traffic in the training dataset. The resultant obtained cluster centroids are then used for fast anomaly detection

STEP 1: Apply k-means clustering to the traffic data set - packet data such as header information, start and end time stamps, number of packets (total number of packets sent) and bytes(total number of bytes sent).

STEP 2: Determining the initial partition. Initialize K clusters by selecting packet, time stamps and bytes as elements.

STEP 3: Determine the optimal number of clusters based on optimization algorithm.

STEP 4: Update the clusters. Apply iteration algorithm to the training traffic data set. Calculate the distance using the Euclidean distance formula. Assign each and every object to the cluster with the nearest obtained centroid value.

STEP 5: Recalculate the centroid value of the modified clusters.

STEP 6: Repeat step 4 until the centroid values do not change.

We apply the K-means clustering algorithm to training datasets which may contain both normal and anomalous traffic The clustering technique used behind this approach is that normal and anomalous traffic form different clusters. The clustering algorithm divides the training data into K clusters, but does not determine if a cluster contains normal or anomalous traffic. Even the good nodes may be considered as anomalous and bad nodes may be treated as normal node.

IV. AGGREGATION OF CLUSTERED ANOMALY DDOS DETECTION (SIMILARITY OF ATTACK INTENSITIES)

Network DDoS Attack detection systems can be divided into two types as either signature based or anomaly based. A signature detector examines traffic for attacks that are known to the system using the rules which have been written by experts or administrator. When some other attack is occurred, new rules must be written and distributed into the system. An anomaly detection system models the normal traffic which distributes the IP addresses and ports. Hostile traffic module falls outside this distribution model. In case of anomaly detection, it can detect novel attacks without having the rules to be written. But it suffers from the disadvantage that it cannot consider about the nature and type of attack as it is a novel type and as such normal traffic can also deviate from the model, which generates false alarms. The job of anomaly detector is to bring the suspicious traffic to the attention of administrator, who must then note it out, if anything needs to be done to alter it.

To calculate the direction of traffic data correlation consider two adjacent sampling instants. The demonstration model defines the direction of traffic data correlation of the donor site. If one extends from the two measuring points, ie n-1, you get a positive contribution. In order to minimize storage and processing complexity, use a linked data structure. A report on location is used to record the number of packages to address the IJ in IP address with the extension. The use of the approximate representation of addresses allows us to reduce requirements for calculating and storing a key factor. To create the correlation of signaling messages at the end of the sampling point, multiply each segment of the correlation of the scales. From a statistical point of view on average about the same and the standard deviation of the dispersion and cross-correlation coefficient.

Cluster based statistical anomaly DDoS attack detection scheme measures the statistics of the traffic traces of non intrusive packet header data. Traffic is monitored at regular intervals and analyzed using the statistical method by comparing it to historical norms to find anomalies (change detection). Assault cases are regarded as random processes update for the approximate maximum likelihood parameter produce. With these random processes, from top to bottom of attack instances are detected. The cluster traffic streams contain all relevant information that is useful for the administrator, the process of DDoS Attack detection to govern effectively.

Aggregation is an important subtask of DDoS Attack detection. The goal is to identify and to cluster different aggregates-produced by low-level DDoS attack detection systems, firewalls, etc.-belonging to a specific attack instance which has been initiated by an attacker at a certain point in time. Thus, aggregation can be generated for the clusters that contain all the relevant information whereas the amount of data (i.e., alerts) can be reduced substantially. Basically, it can be regarded as a data stream version of a maximum likelihood approach for the estimation of the model parameters. With three benchmark data sets, we demonstrate that it is possible to achieve reduction rates of up to 99.96 percent while the number of missing meta-alerts is extremely low. The goal of aggregation is to reduce the complexity of hyper alert correlation graphs without sacrificing the structures of the attack scenarios; it allows analysts to get concise views of correlated alerts. For this reason, we also refer to aggregation as graph reduction. Aggregation allows analysts to selectively disaggregate certain aggregate thus providing the ability to examine the details of select aggregation.

V. PERFORMANCE EVALUATION OF AGGREGATED CLUSTER ANOMALY DDOS DETECTION

In order to assess the performance of the aggregation, evaluate the following measures

Percentage of detected instances (*p*)
An attack instance is being detected if there is at least one meta-alert that predominantly contains aggregation of that particular instance. The percentage of detected attack instances *p* can thus be determined by dividing the number of instances that are detected by the total number of instances in the data set. The measure is computed with respect to the instances covered by the output of the detection layer, i.e., instances missed by the detectors are not considered.

Number of aggregates (*A*) and reduction rate (*r*)
The number of aggregates (*A*) is further divided into the number of attack. *MA*attack which predominantly contain true aggregates and the number of non-attack aggregates *MA*non-attack which predominantly contain false alerts. The reduction rate *r* is 1 minus the number of created aggregates *A* divided by the total number of attacks A.

Average run-time (*t*avg) **and worst case run-time** (*t*worst)
The average run-time is measured in milliseconds per aggregate. Assuming up to several hundred thousand aggregates a day, *t*avg should stay clearly below 100 ms per aggregate. The worst case run-time *t*worse, which is measured in seconds, states how long it takes at most to execute the *while* loop.

Meta-aggregate creation delay (*d*):
It is obvious that there is a certain delay until a meta-aggregate is created for a new attack instance. The meta-aggregate creation delay *d* measures the delay between the actual beginning of the instance (i.e., the

creation time of the first aggregate) and the creation of the first meta-aggregate for that instance.

During the attack, the client of interest has zero link utilization, meaning the client completely stops getting HTTP data packets since almost all the bandwidth of the link 2–0 is used by the attack traffic. On the other hand, there is no visible difference in the link utilization of upstream server link nor in the link utilization of the bottleneck link after the attack. To detect this attack, the proposed aggregation use the nonlinear mutual information computed for the link utilization observed on the bottleneck link.

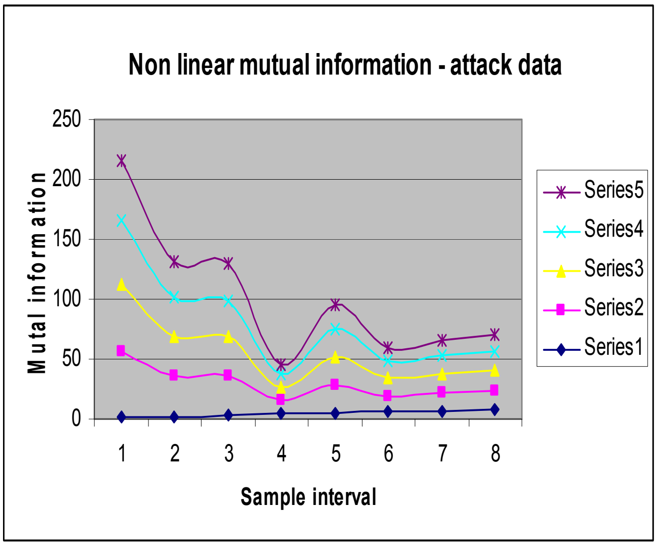

Figure 2: Nonlinear mutual information – attack data

Figure 2 shows the mutual information plots for this experiment for different trials. It can be seen that there is a significant change in the mutual information, even though the attack cannot be seen by visual inspection of the link utilization plots. It is important to note that since the link utilization remains constant during the attack, count based methods that simply consider the amplitude of the link utilization during a sample period are unable to detect the attack.

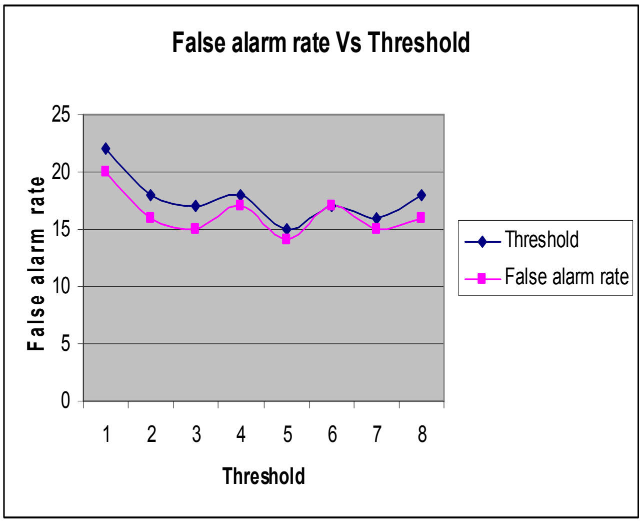

Figure 3: False alarm rate versus threshold

The performance of the detection scheme is related to the value of threshold. Figure 3 shows the relationship between the false alarm rate and threshold. A false alarm is said to occur if *Sk* exceeds threshold without attack. After a false alarm, *S* is reset to 0 and the time series is continued to be processed. As expected, as threshold grows, the false alarm rate decreases. No false alarms occurred for threshold above 160; hence no points are included for threshold > 160. However, as long as threshold is below 1600, the attack is detected.

VI. CONCULSION

Cluster aggregates varies subtask of DDoS detection. A different aggregate produced by low-level DDoS detection systems, firewalls, etc is evaluated to make the system more foolproof. It identifies and cluster DDoS attack detection to make the segregation of various attacks being generated. To improve the efficacy of DDoS attack detection system, aggregation are generated which contain all the relevant information The experiments demonstrated the broad applicability of the proposed aggregation approach. The simulation conducted for two different data sets and showed that machine learning based detectors, conventional signature based detectors, and even firewalls can be used as aggregation generators. In all cases, the amount of data could be reduced substantially. Although there are situations as described in clusters that are wrongly split the instance detection rate, none or only very few attack instances were missed. Run-time and component creation delay are well-suited for an on line application.

Here we presented the technique for data stream aggregation and generation of resultant aggregate values. It has shown that the sheer amount of data that must be reported to a human security expert or communicated within a distributed DDoS detection system, for instance, can be reduced significantly. The reduction rate with respect to the number of aggregates was up to 97 in our simulation. The number of missing attack instances is extremely low or even zero in some of our simulation and the delay for the detection of attack instances is within the range of some seconds only.

REFERENCES

[1] David K. Y. Yau, Member, IEEE and John C. S. Lui, Feng Liang, and Yeung Yam, (2005) 'Defending Against Distributed Denial-of-Service Attacks With Max-Min Fair Server-Centric Router Throttles', IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 13, NO. 1, Pages 29-42.

[2] John Ioannidis and Steven M. Bellovin, 'Implementing Pushback: Router-Based Defense Against DDoS Attacks', AT&T Labs Research, ji@research.att.com, smb@research.att.com.

[3] Michael K. Reiter and XiaoFeng Wang, (2004), 'Mitigating Bandwidth-Exhaustion Attacks using Congestion Puzzles'**,** *CCS'04,* October 25-29, 2004, Washington, DC, USA., Copyright ACM 1-58113-961-6/04/0010.

[4] Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, (2000), 'Practical Network Support for IP Traceback', ACM SIGCOMM Computer Communication Review, Volume 30, Issue 4, Pages 295-306.

[5] Tao Peng, Christopher Leckie, and Kotagiri Ramamohanarao, (2007), 'Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems', Department of Computer Science and Software Engineering, The University of Melbourne, Australia, ACM Comput. Surv., Volume 39, Issue 1, Article no.3.

[6] XiaoFeng Wang and Michel K. Reiter, (2008), 'A multilayer frame work for puzzle-based denial-of-service defense', ACM, International Journal of Information Security, ISSN:1615-5262, Page 243-263.

[7] Saravanan Kumarasamy (2011), 'An Effective Defence Mechanism For Distributed Denial-Of-Service (Ddos) Attacks Using Router-Based Techniques', Int. J. Critical Infrastructures, Vol. 6, No. 1, 2010,Page No. 73-80

[8] Saravanan kumarasamy, Dr.R.Asokan ,' Distributed Denial Of Service (Ddos) Attacks Detection Mechanism' International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol.1, No.5, December 2011,Page no:39-49.

Saravanan Kumarasamy received the M.E degree 2008 in computer science from Dr.MCET, Anna University, and Chennai, India. He is currently working as a Lecturer at the Faculty of Engineering, Erode Sengunthar Engineering College, Erode, Tamilnadu.

He has published 5 papers in International Journal, 10 papers in National Conference and 02 papers in International Conference.His current research interests are information security, computer communications and DDoS Attacks. He is currently pursuing Ph.D. under Anna University of Technology, Coimbatore.

Dr.R.Asokan received the Ph.D degree in Information and Communication Engineering from Anna University, Chennai. He has 25 years of teaching experience. At present he is working as Principal at Kongunadu College of Engineering and Technology, Thottiam, Trichy. He has published more than 70 papers in National and International Journals and Conferences. He has organized more than 20 Seminars, Workshops and Conferences.

He has delivered around 40 special lectures in various summer / winter school/ sponsored programmes. He is the associate editor for Journal of selected areas in telecommunication and also Editorial board member for five International Journals. His areas of interest include communication networks, network security and image processing. He is an active member in many Professional bodies like IETE, ISTE, CSI, ACS etc.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)