International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

MRI- A Map Reduce Interpretation Framework for mitigating DDoS attacks in Big Data

S.M.Lakshmanan., M.B.A.,(M.Phil)[1], P.Karthikeyan.,B.Sc(CS).,M.C.A.,M.Phil., (Ph.D)[2]

[1]Research Scholar, Department of Computer Science, Prist Deemed to be University.
[2]Research Advisor and Assistant Professor, Department of Computer Science, Prist Deemed to be University, Thanjavur.

Abstract - *Big data include capturing data, data storage, dataanalysis,search, sharing, transfer, visualization, querying, updating, information privacy and data source. Big data can be described by the following characteristics: Volume, Variety, Velocity, Variability and Veracity. Applications are in Government, International development, Manufacturing, Healthcare, Education, Media, Information Technology . DDoS attacks have a history of flooding the victim network with an enormous number of packets, hence exhausting the resources and preventing the legitimate users to access them. After having standard DDoS defense mechanism, still attackers are able to launch an attack. These inade-quate defense mechanisms need to be improved and integrated with other solutions. The purpose of this paper is to study the characteristics of DDoS attacks, various models involved in attacks and to provide a timeline of defense mechanism with their improvements to combat DDoS attacks. In addition to this, a novel scheme is proposed to detect DDoS attack efficiently by using Map Reduce programming model using K-means clustering algorithm*

***Key Words*: DDoS, Defense Mechanism, K-means Algorithm, MapReduce**

1.INTRODUCTION

The massive amounts of data that collect over time which difficult to analyze using common database management tools. Big data includes activity logs (machine generated data) which consist of unstructured format capture from web. The storage industry is continuously challenged as Big data increases exponentially where security is one of the challenging and harmful concern. To handle Big data technology takes cardinal part in analysis.

1.1 Distributed denial of service (DDoS) attacks

Distributed denial of service (DDoS) attack is some sort of malicious activity or a typical behavior, which cooperate the availability of the server's resources and prevents the legitimate users from using the service. DDOS attacks are not meant to alter data contents or achieve illegal access, but in that place they target to crash the servers, generally by temporarily interrupting or suspending the services of a host connected to the Internet. DOS attacks can occur from either a single source or multiple sources. Multiple source DOS attacks are called distributed denial-of service (DDOS) attacks.

The Agent-Handler model of a DDoS attack con-sists of agents, handlers and client. Fig 1 shows the Agent-Handler Model, in which the Agent and handler knows each-others identity. The client is the interface where the attacker/mastermind communicates with the rest of the DDoS Components. The handlers are software packages distributed all over the Internet so that it helps to client to convey its command to the agents. The agent software's are vulnerable systems, compromised by the handlers and actually launch the attack on victim's ma-chine. The agent's status and schedule for launching at-tack can be upgraded by the handler when it is required. Communication relation between agent and handler is either one to one or one to many. Most Common way to attack is by installing handler instructions either on com-promised route on network layer or on network server. This makes it difficult to identify messages exchanged by the client-handler and between the handler-agents.

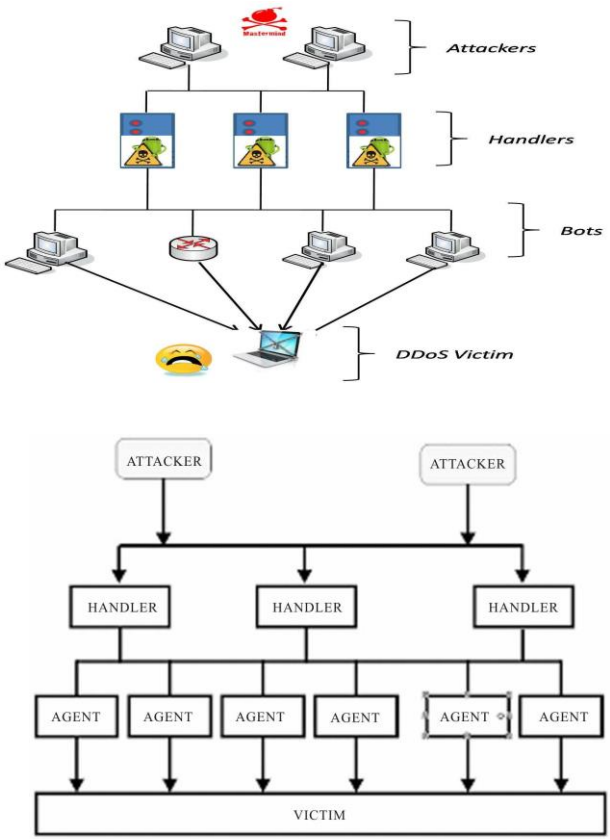

Fig. 1. Components of DDoS attack

1.2 Big Data

As everyday data are being collected from applications, networks, social media and other sources Big Data is emerging. Studies have shown that by 2020 the world will have increased 50 times the amount of data it had in 2011, which was currently 1.8 zettabytes or 1.8 trillion gigabytes of data . The basic reason for the sharp increase in data being stored over the years simply comes down to cost of storage. The IT industry has made the cost of storage so cheap that applications are capable of saving data at exponential rates. This brings the challenge of having existing network infrastructure learn how to manage and process this big data so that it can be utilized into useful information. Many big data applications work in real-time. Hence, these applications need to create, store and process large amount of information which produces a great deal of volume and demand on the network. When looking at data from a networking perspective, many different areas are needed to be explored These include network topology optimization, parallel structures and big data processing algorithms, data retrieval, security, and privacy issues. The topic of big data is still a new exciting area of research among the IT community and will be requiring much attention for the years to come. A typical organization has a limited network infrastructure and resources capable of handling these volumes of traffic flows which cause regular services (e.g., Email, Web browsing, video streaming) to become strained. This can reduce network performance affecting bandwidth and exposing hardware limitations of devices such as firewall processing being overwhelmed. Providing security and privacy has also become a major concern in Big Data as many critical and real-time applications are developed based on Big Data paradigm.

2. MRI- MAP REDUCE INTERPRETATION FRAMEWORK

A Denial of Service (DoS) attack is an attempt to make a computer resource unavailable to normal users. The Dos attacks are becoming more powerful due to bot behavior. Attack that leverages multiple sources to create the denial-of-service condition is known as The Distributed Denial of Service (DDoS) attack.

2.1 Theoretical Foundation

The Map Reduce framework first splits an input data file into G pieces of fixed size, typically being 16 megabytes to 64 megabytes (MB). These G pieces are then passed on to the participating machines in the cluster. Usually, 3 copies of each piece are generated for fault tolerance. It then starts up the user program on the nodes of the cluster. One of the nodes in the cluster is special the master. The rest are workers that are assigned work by the master. There are M map tasks and R reduces tasks to assign. M and R is either decided by the configuration specified by the user program, or by the cluster wide default configuration. The master picks idle workers and assigns them map tasks. Once map tasks have generated intermediate outputs, the master then assigns reduces tasks to idle workers. Note that all map tasks have to finish before any reduce task can begin. This is because a reduce task needs to take output from every map task of the job. A worker who is assigned a map task reads the content of the corresponding input split. It parses key/value pairs out of the input data chunk and passes each pair to an instance of the user defined map function. The intermediate key/value pairs produced by the map function are buffered in memory at the corresponding machines that are executing them. The buffered pairs are periodically written to a local disk and partitioned into R regions by the partitioning function. The framework provides a default partitioning function but the user is allowed to override this function by a custom partitioning. The locations of these buffered pairs on the local disk are passed back to the master. The master then forwards these locations to the reduce workers. When a reduce worker is notified by the master about these locations, it uses remote procedure calls to read the buffered data from the local disks of map workers. When a reduce worker has read all intermediate data, it sorts it by the intermediate key so that all occurrences of the same key are grouped together.

The sorting is needed because typically many different keys are handled by a reduce task. If the amount of intermediate data is too large to fit in memory, an external sort is used. Once again, the user is allowed to override the default sorting and grouping behaviors of the framework. Next, the reduce worker iterates over the sorted intermediate data and for each unique intermediate key encountered, it passes the key and the corresponding set of intermediate values to the reduce function. The output of the reduce function is appended to a final output file for this reduce partition. When all map tasks and reduce tasks have completed, the master wakes up the user program. At this point, the Map Reduce call in the user program returns back to the user code.

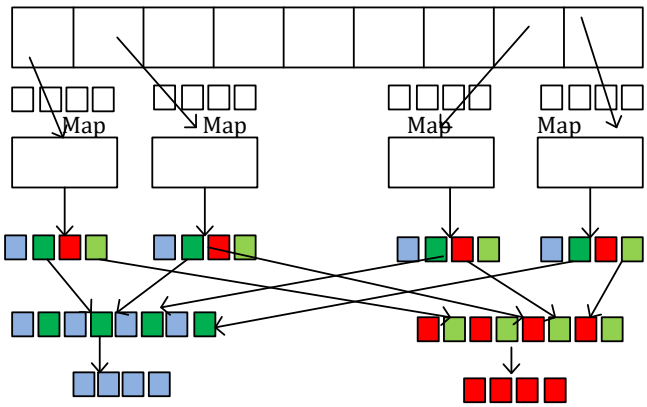

Fig. 2 MapReduce paradigm

There are several tools like Wireshark, tshark etc for displaying the packets. But when it comes to large number of packets say petabytes and terabytes, these tools don't contribute much. Map reduce is the best framework for

doing such work. Clients can write suitable map and reduce function for the particular task. As mentioned earlier Map Reduce framework requires a key value pair. In this work key is the source ip ,destination ip address and the protocol and the value is count. Mapper part gets the keys and the value. Reducer part shuffles, sorts and merges and gets the output as the number of packets of specific type from a particular source to destination. It can also be used to find the number of packets send to specific ports.

2.2 K-Means Clustering

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\left\| x_i - v_j \right\| \right)^2$$

where,

'$||x_i - v_j||$' is the Euclidean distance between x_i and v_j.

'c_i' is the number of data points in i^{th} cluster.

'c' is the number of cluster centers.

3. PROPOSED SCHEME

Map reduce is the data processing framework. It deals with the implementation for processing and generating large datasets with a distributed algorithm on a cluster . Map reduce is used in big data and Input data is splitted and fed to each node in the map phase. The results generated in this phase are shuffled and sorted then fed to the nodes in the reduce phase. The technique uses the historical information that is being stored in each node and using that information it finds the real slow tasks. Then it maps the slow tasks and reduces the slow tasks.

3.1 Process Technique of MRI

The k-means clustering technique is used to tune the parameters in the historical information and finding the slow tasks very accurately. It requires the number of clusters that we are going to use in our process. The algorithm finds k centroids, one for each cluster. During the map phase it finds the M1 temporary value and using this value it finds in the clusters which one is closest to the M1 value. Similarly in the reduce phase it finds the R1 temporary value and using this value it finds in the clusters which one is closest to the R1 value and the values are recalculated again.

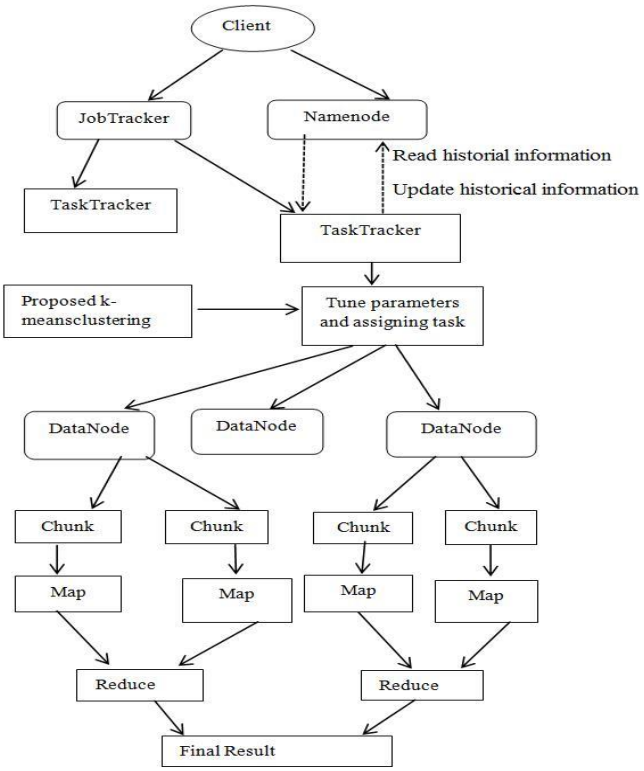

Fig. 3 Map Reduce Implementation

i. Map step

The average output of the map will be recorded ID as the key and retired as the value. Every mapper maintains a collection bearing the canopy center candidates it has learned thus far. During every map the mapper determines if each successive record is within the distance threshold of any already determined canopy center candidate. The intermediate output sent to the reducer has the record ID as the cay and the list of retired-rating pairs as the value.

ii. Reduce Step

The yield of the reduce step will simply output record ID as the key and concatenate the rater IDs for that record into a comma separated list. The reducer repeats the same procedure as the mappers. It meets the candidate

canopy center record IDs, but takes out those which are inside the same threshold limit . In other words, it removes duplicate candidates for the same canopy center. In order for this to operate correctly the number of reducers is set to one. As an industry-leading anti-DDoS solutions provider, To apply Big Data technology to the detection and prevention of covert DDoS attacks disguised as normal access requests. These heavy-traffic DDoS attacks are the easiest to detect, but require the highest processing performance to affect the necessary rapid response; otherwise, the network links will become jammed, completely flooded, while security devices deployed on the access side are failing. Until the recent arrival of cost-effective flow analysis technology, these super- large-bandwidth DDoS attacks were best handled by commercial anti-DDoS. Blocking such attacks requires the deployment of super-large capacity prevention systems on the upstream side of the network. Effective enterprise anti-DDoS systems must be based on high-performance hardware platforms with a minimum 100-Gbit/s defense capacity, or the defense device itself will likely become the network bottleneck. We have now entered the era where these high performance tools are now available for enterprises. Because DDoS attack detection systems rely on traffic models for attack detection, the better the traffic model the higher the probability of detecting attacks. The difficulty in detecting light-traffic attacks is that the small numbers of attack packets are concealed in massive volume of legitimate network access packets. Mitigating this type of attack using traditional prevention systems can only limit the connections of legitimate access sources.

3.2 Algorithm steps:

Step 1: Input: D-set of n data nodes, n-number of data nodes, C-set of k centroids, k-number of clusters
Step 2: Output: A-set of k clusters
Step 3: Compute distance between each data nodes to all centroids
Step 4: For each Di find the closest Ci
Step 5: Add Di to A
Step 6: Remove Di from D
Step 7: Repeat for all Di……Dn andCi...Ck

3.3 Merits of MRI

High security*:* The security solution must be able to defend against DDoS attacks of various types, regardless of the traffic attacks or application-layer attacks, to protect all online services from attacks.

High performance*:* To avoid being the bottleneck of the whole system, the security solution must feature high-performance defense capabilities so that it can deal with the traffic flooding attacks on Tencent's large-scale services.

High scalability*:* The security solution must support flexible performance expansion to vary with service requirement changes, catch up with service mode innovation, and form an architecture required for long-term service development, in order to protect previous investment and reduce total investment cost..

High availability*:* The security solution must ensure reliable service connections, precisely differentiate attack traffic from normal traffic, and accurately identify attacks

Low O&M cost*:* Considering that O&M cost significantly affects Ten cent, the security solution must be small-sized, consume low power, minimize occupied equipment room space and consumption with improved performance

4. CONCLUSION

This paper discusses the history the of DDoS attacks along with some major incidents to provide a better understanding and gravity of the problem. The paper includes latest techniques such as MRI along with other available techniques for prevention and detection of distributed denial of service attacks so that a comprehensive solution can be developed with several detection layers to trap the intrusion keeping in mind the limitations of these prevention and detection techniques. The paper also discusses some of the recent development happened in the sphere of DDoS. Though this technique sounds promising, it can be further optimized. In this paper we proposed a method to improve the efficiency of the map reduce scheduling algorithms. It works better than existing map reduce scheduling algorithms by taking less amount of computation and gives high accuracy. We used the proposed k-means clustering algorithm together with the Map Reduce algorithm. However this technique works well it can assign only one task to each data node. In the future we have decided to improve its efficiency by allocating more number of tasks to the data nodes.

REFERENCES

[1] S.Ezhilarasi, “HHH- A Hyped-up Handling of Hadoop based SAMR-MST for DDOS Attacks in Cloud”, International Research Journal of Engineering and Technology (IRJET) , Vol 05 Issue 03, March 2018.

[2] T. Kitten, “DDoS: Lessons from Phase 2 Attacks,” 2013.

[3] S. Zargar, J. Joshi and D. Tipper, “A Survey of Defense Mechanisms against Distributed Denial of Service (DDoS) Flooding Attacks,” Communications Surveys & Tutorials, IEEE, Vol. PP, No. 99, 2013, pp. 1-24

[4] “CERT Advisory: SYN Flooding and IP Spoofing At-tacks,” CERT® Coordination Center Software Engineer-ing Institute, Carnegie Mellon, 2010. http://www.cert.org/advisories/CA-1996-21.html

[5] CERT, "Tech Tips: Denial of Service Attacks," CERT® Coordination Center Software Engineering Institute, Carnegie Mellon, 2010.
http://www.cert.org/tech_tips/denial_of_service.html

[6] R. Mackey, "'Iranian Cyber Army' Strikes Chinese Website," New York Times Lede Blog, 2011.

[7] DDoS-for-Hire Service Is Legal and Even Lets FBI Peek in, Says a Guy with an Attorney," 2012.
http://www.ddosdefense.net

[8] J. Kirk, "Mt. Gox under Largest DDoS Attack as Bitcoin Price Surges," 2013.
http://www.computerworld.com/s/article/9238118/Mt._Gox_under_largest_DDoS_attack_as_bitcoin_price_surges

[9] "Mstream Distributed Denial of Service Tool (Zombie Detected) (DdosMstreamZombie)," 2013.
http://www.iss.net/security_center/reference/vuln/ddos-mstream-zombie.htm

[10] N. McAllister, "GoDaddy Stopped by Massive DDoS Attack,"2012.
http://www.theregister.co.uk/2012/09/10/godaddy_ddos_attack/

[11] D. Kravetz, "Anonymous Unfurls 'Operation Titstorm'," Wired Threat Level Blog, 2010.

[12] K. Zetter, "Lazy Hacker and Little Worm Set off Cyberwar Frenzy," 2009.
http://www.wired.com/threatlevel/2009/07/mydoom/

[13] L. Greenemeier, "Estonian Attacks Raise Concern over Cyber "Nuclear Winter"," Information Week, 2007.
http://www.informationweek.com/estonian-attacks-raise-concern-over-cybe/199701774

[14] Keman Huang, Jianqiang Li, and MengChu Zhou, Jan-2015," An Incremental and Distributed Inference Method for Large-Scale Ontologies Based on MapReduce Paradigm".

[15] Antoniou G and Bikakis A, 2007-"DR- Prolog: A system for defeasible reasoning with rules and ontologies on the Semantic Web," IEEE Trans. Knowl. Data Eng., vol. 19, no. 2, pp. 233–245.

[16] Billion Triples Challenge 2012 Dataset [Online]. Available: http://km.aifb.kit.edu/projects/btc-2012/

[17] Dean J and Ghemawat S, 2008-"MapReduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113,2008.

[18] Grau, B.C, Halaschek-Wiener C, and Kazakov Y, 2007- "History matters: Incremental ontology reasoning using modules," in Proc. ISWC/ASWC, Busan, Korea, pp. 183–196.