Detection Of Distributed Denial of Service Attack With Hadoop On Live Network

Suchita Korad[1], Shubhada Kadam[2], Prajakta Deore[3], Madhuri Jadhav[4], Prof.Rahul Patil[5]
[1,2,3,4]*Students, Department of Computer Engineering, University of Pune*
[5]*Assistant Professor, Department of Computer Engineering, University of Pune*
Pune, India

***Abstract*- Distributed Denial of Service i.e. DDoS flooding attacks are one of the biggest challenges to the availability of online services now-a-days. These DDoS attacks overwhelm the victim with huge volume of traffic and render it incapable of performing communication or crashes it completely. If there are delays in detecting the flooding attacks, we have to manually disconnect the victim and fix the problem. With the rapid increase of DDoS attack volume and frequency, the current DDoS detection techniques are challenged to deal with huge attack volume in reasonable and affordable response time. In this paper, we had proposed, a Hadoop based Live DDoS Detection framework to tackle efficient analysis of flooding attacks by using core components of Hadoop like MapReduce and HDFS. We implemented a counter-based DDoS detection algorithm for four major flooding attacks (TCP-SYN, UDP and ICMP, HTTP GET) in MapReduce, consisting of mapper and reducer functions. We deployed a testbed to evaluate the performance of Hadoop framework for live DDoS detection. Based on the experiment we showed that Hadoop is capable of processing and detecting DDoS attacks in affordable time.**

***Index Terms*— Hadoop, HDFS, DDoS, Flooding attacks**

I. INTRODUCTION

Distributed Denial of Service (DDoS) flooding attacks are one of the biggest concerns for security and network professionals.As the volume of Internet traffic increases explosively year after year, theIntrusion Detection Systems (IDSes) have faced the issue on how to assure both scalability and accuracy of analyzing the DDoS attack from these huge volume ofdata.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 behavior of attacker. Attack that leverages multiple sources to create the denial-of-service condition is known as The Distributed Denial of Service (DDoS) attack. DDoS attacks are big threats to internet services. Now a day there is massive growth in internet traffic.One of the major advantages of hardware-based DDoS defense system is that they can process packets at a higher speed but problem with this system are high false positive rate. In order to address these problem of false positive rate & big data traffic we are implementing software based system by which we can solve problem of high traffic rate. In this paper we are using hadoop framework in terms to address the problem of big data which is caused by DDoS attacks. We are proposing Counter based algorithm to detect DDoS attack to solve the problem of high false positive rate to detect DDoS attack using behavior of attacker.

II. HADOOP FRAMEWORK

Hadoop is an open-source software framework that supports data-intensive distributed applications. It enables applications to work with thousands of computationally independent computers and with petabytes of data. Hadoop increases the storage space and the processing power by uniting many computers into one. A small Hadoop cluster will include a single master and multiple worker nodes (slaves) as in Figure-1.

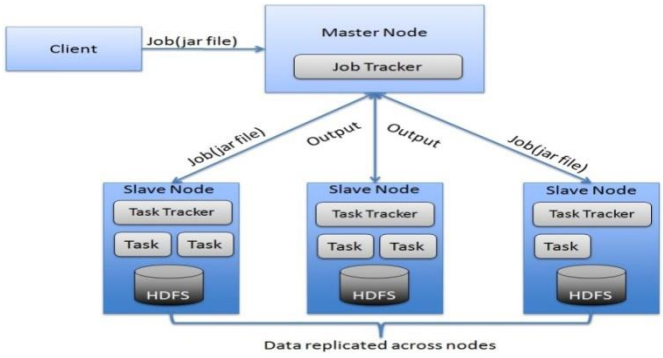

Figure 1:Hadoop Framework

The master node consists of a Job Tracker, Task Tracker,Name Node and Data Node. A slave or worker node acts as both a Data Node and Task Tracker. In a large cluster, HDFS is managed through a dedicated NameNode server to host the file system index and a secondary Name Node that can generate snapshots of the Name Node's memory structures, thus

preventing file system corruption and reducing loss of data. Hadoop Distributed File System (HDFS) is a distributed, scalable, and portable file system written in Java for the Hadoop framework. Map Reduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters of commodity hardware in a reliable ,fault tolerant manner.

III. HADOOP DDOS DETECTION FRAMEWORK

The Hadoop Based DDoS Detection Framework comprise of four major phases

1. Packet capturing and Log generation.
2. Log transferring to HDFS.
3. Detection of DDoS Attack.
4. Result.

Each of the above mentioned phases are implemented as separate components that communicate with each other to perform their assigned task. Packet capturing and log generation are handled at the capturing server, whereas DDoS detection and result notification is performed by the detection server. In the following subsections we have explained the functionalities for each of the phase/component in detail.

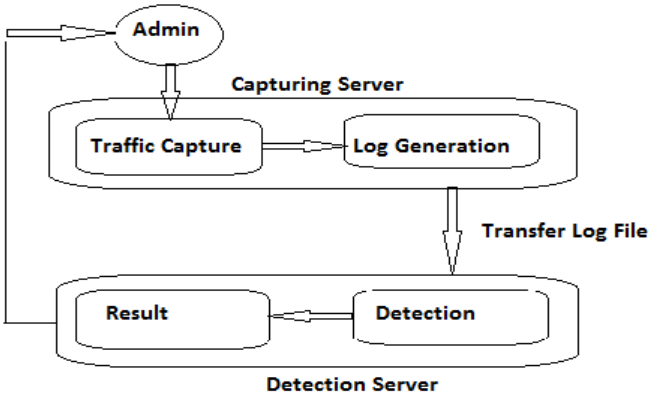

Figure 2: Different Phases

A. Packet Capturing and Log Generation

DDoS detection starts with the capturing of network traffic coming to the server.This system provides a web interface through which the admin can tune the capturing server with desired parameters .Wireshark is an open source library capable of capturing huge amount of traffic. Under default settings, python scriptwhich runs through command line, and outputs the result on console. To log the traffic for later use. We have also tuned script to output only the relevant information required during detection phase. This includes information of timestamps, source IP, destination IP, packet protocol and brief packet header information.

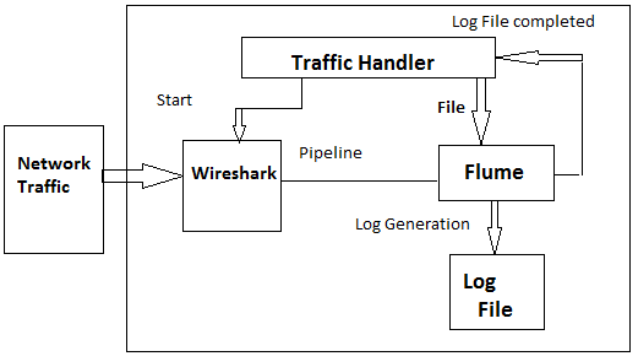

Figure 3 : Network Traffic Capturing and Log Generation Component

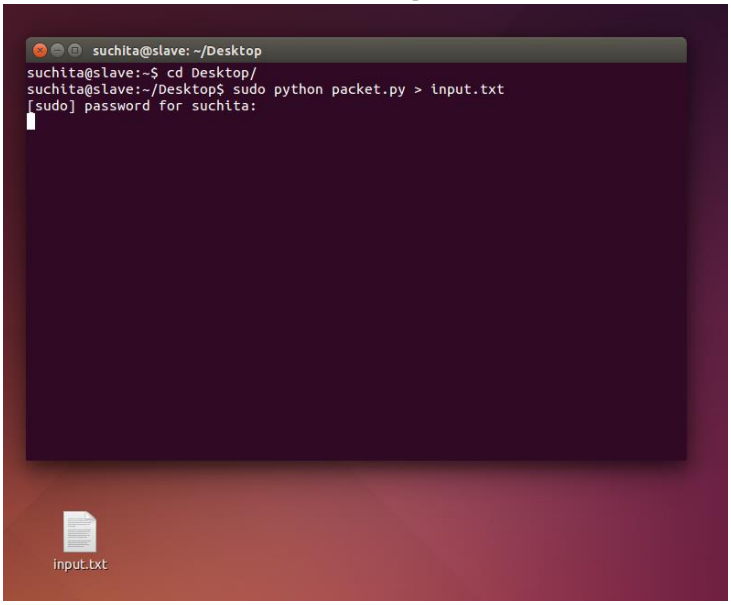

Figure 4:Packet Capturing

B. Log Transfer

After the log file is generated, the Traffic Handler in the capturing server will temporarily pause the traffic capturing operations of script. The traffic handler will then notify the detection server and also share the file information with HDFS.

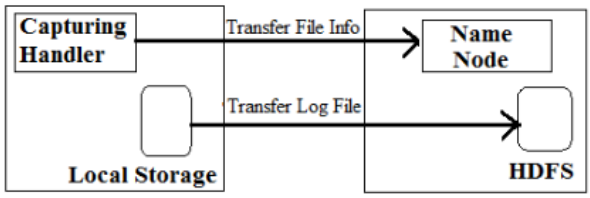

Figure 5 :Log Transfer Phase

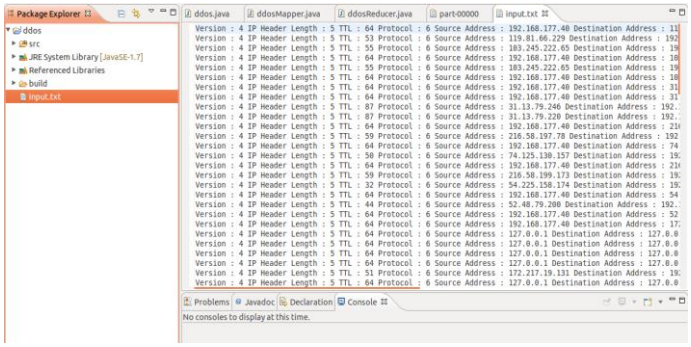

Figure 6 :Log Transfer

C. DDoS Detection

The Apache Hadoop consists of two core components i.e. HDFS and MapReduce . Hadoop's central management node also known as NameNode splits the data into large number of same size blocks and distributes them amongst the cluster nodes. Hadoop's MapReduce transfers packaged code for nodes to process in parallel, the data each node is responsible to process. The detection server mainly serves as the Hadoop's NameNode, which is the centerpiece of the Hadoop DDoS detection cluster. On successful transfer of log file/files, the detection server split the file into same size large blocks and starts MapReduce DDoS detection jobs on cluster nodes. We have discussed MapReduce job analyzer and counter based DDoS detection algorithm.

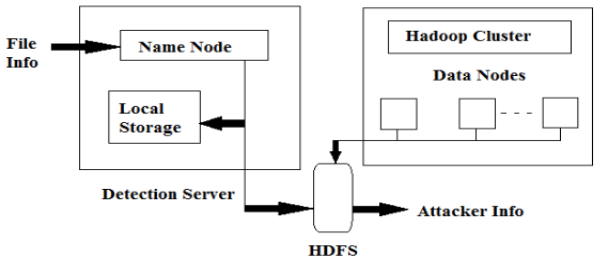

Figure 7: DDoS Detection on Hadoop Cluster

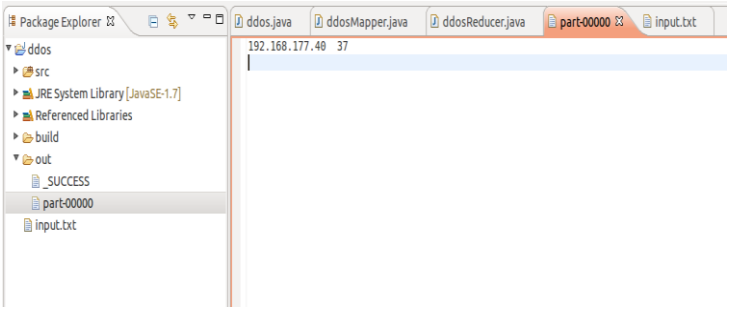

Figure 8 : DDoS Detection

Counter Based algorithm

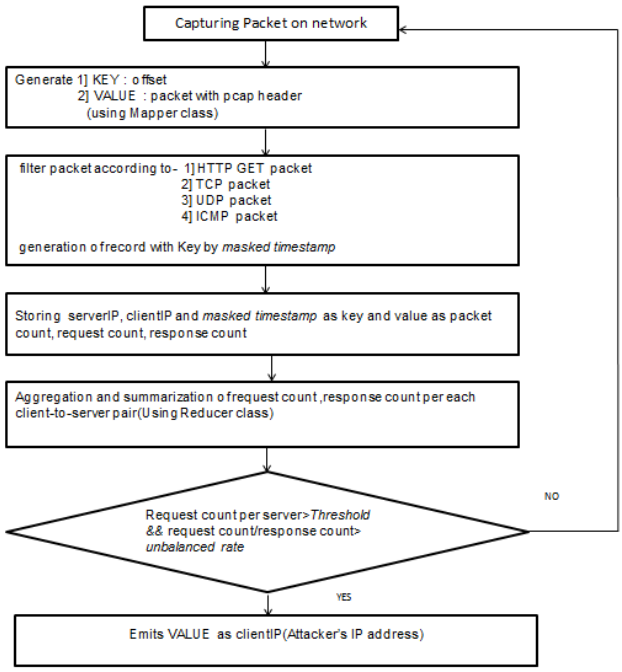

Figure 9 : The MapReduce algorithm for counter-based DDoS detection

In detection phase of our paper we are implementing this algorithm to detect DDoS attacks. This algorithm needs three inputs which are time interval, threshold and unbalance ratio, which can be loaded through the distributed cache mechanism of MapReduce. Time interval is to limit monitoring duration of the page request. Threshold indicates the frequency of the page request to the server against the previous normal status of network, which determines whether the server should be alarmed or not. The unbalance ratio denotes the ratio of response per page request between a specific client and a server. This value is used for finding out attackers from the clients. Finally, the algorithm aggregates and summarizes values per server. When total requests for a

specific server from client exceeds the threshold value limit, the MapReduce job emits that records whose response ratio against requests is greater than unbalance ratio, marking them as attacker's IP address. While this algorithm has the low computational complexity and could be easily converted into the MapReduce implementation, it needs a prerequisite to know the threshold value from historical monitoring data from network in advance.

D. Result Notification

Once the execution of all the MapReduce tasks is finished, Hadoop will save the results in HDFS. The detection server will then parse the result file from HDFS and send the information about the attackers back to the administrator via the capturing server. Once the results are notified both the input and output folders from HDFS will be deleted for better memory management by the detection server.

IV. MAPREDUCE JOB AND DDOS DETECTION

A MapReduce program is composed of a Map task that performs functions of filtering and sorting and a Reduce task that performs a summary operation. Here we have explained how we have implemented detection of DDoS flooding attacks (UDP, HTTP GET, ICMP and TCP-SYN) as a MapReduce task on Hadoop cluster using counter-based algorithm.

A. Mapper Job

After starting MapReduce task, the first task is a mapper task which takes input from HDFS as a block. In our case the block will represent a file in text format and the input for each iteration of mapper function will be a single line from the file. Any single line in the file contains only brief information of a network packet captured through Wireshark. Mapper job takes pair of data as input and returns a list of pairs. Mapper's output type may differ from mapper's input type, in our case the input of mapper function is pair of any number and the network packet. In our MapReduce algorithm, the map function filters network packets and generates key values of server IP address, masked timestamp, and client IP address. The masked timestamp with time interval is used for counting the number of requests from a specific client to the specific URL within the same time duration. Mapper job also use hashing for combining all the logs of data on the basis of source IP address, so that it becomes easier for reducer to analyze the attack traffic.

B. Reducer Job and Counter-Based Algorithm

Once the mapper tasks are completed, the reducer will start operating on the list of key/value pairs (i.e./Packet pairs) produced by the mapper functions. The reducers are assigned a group with unique key, it means that all the packets with unique key will be assigned to one reducer. We can configure Hadoop to run reducer jobs on varying number of data nodes. For efficiency and performance it is very important to identify the correct number of reducers required for analyzing the analysis job. Hadoop run counter-based algorithm to detect DDoS flooding attacks on reducer nodes. The reduce function summarizes the number of URL requests, page requests, and server responses between a client and a server. Counter based algorithm is the simplest, yet very effective algorithm to analyze the DDoS flooding attacks by monitoring the traffic volumes for source IP addresses.

V. CONCLUSION

In this paper, we present Hadoop framework, a scalable Hadoop based Live DDoS Detection framework that is capable of analysing DDoS attacks in affordable time. It captures live network traffic i.e. packets, process it to log relevant information in brief form and use MapReduce and HDFS of Hadoop to run detection algorithm for DDoS flooding attacks. Hadoop solves the scalability, memory inefficiency and process complexity issues of conventional solution by utilizing parallel data processing by Hadoop. The evaluation results showed that it takes would less than 5 mins to process (from capturing to detecting) 1GB of log file, generated from approx. 15.83 GBs of live network traffic. With small log file the overall detection time can be further reduced to couple seconds.

REFERENCES

[1] Hadoop. https://hadoop.apache.org/.
[2] Hadoop yarn. http://hortonworks.com/hadoop/yarn/.
[3] J. Mirkivic and P. Reiher, A Taxonomy of DDoS Attack and DDoS Defense Mechanisms, ACM SIGCOMM CCR, 2004
[4] H. Sun, Y Zhaung, and H. Chao, A Principal Components Analysis-based Robust DDoS Defense System, IEEE ICC, 2008
[5] H. Liu, Y Sun, and M. Kim, Fine-Grained DDoS Detection Scheme Based on Bidirectional Count Sketch, IEEE ICCCN, August 2011
[6] Y. Lee, W. Kang, and Y. Lee, A Hadoop-based Packet Trace Processing Tool,TMA,April2011