TCP DDos Load Balancer using Point Server

M. Karthi[1], J. Sakthivel[2] ,K. Vignesh[3], M Ramakrishnan M.E,(Ph.D..)[4]
Assistant professor of
Department of Computer Science and Engineering
Sree Sakthi Engineering College

***Abstract*— Cloudcomputing data centers are the most essential infrastructures in the information era. So, it is necessary to consider the user data security, the DDOs is the one of the most important thread in this world. It is very difficult to prevent and detect the attack. To detect and prevent from DDoS attacks, we propose the new type of server architecture which safeguard server from the DDOS attack. We also done some of the additional input filtration techniques to safeguard the user data and private data from the attacker, through other types of attacks such as SQL injection, XSS, Session Hijacking. We built the new form of our products to test its working efficiency by building some web apps. Our products show that our new framework and architecture prevent the 70% of attacks and detect 93% of attack without any datasets. This approach will be deployed to the victim-end vtechnosoft website and some of our client dataset products.**

I. INTRODUCTION

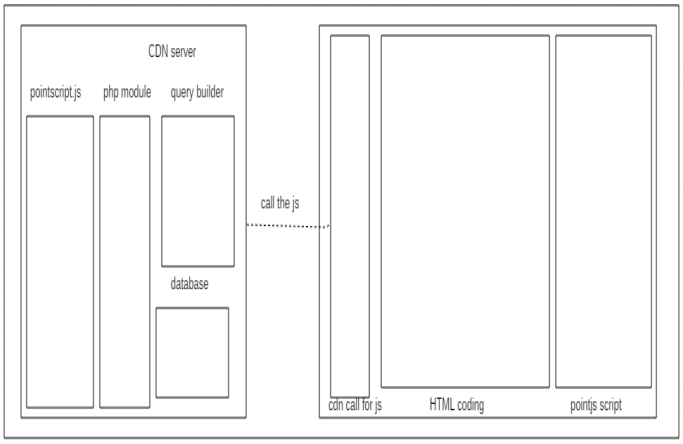
TCP traffic has recently been exploited broadly in Distributed Denial of Service attacks. At present, half of all network DDoS attacks are SYN flood attacks which are considered one of the most powerful attack methods [1]. At the same time, Challenge Collapsar (HTTP flood) attacks have been emerging frequently, along with the other types of attacks to steal the data TCP based DDoS attacks can utilize multiple attack types and different attack modes, which makes it extremely difficult, to detect and prevent from these attacks. There are various methods that used by attackers to steal the data, the most common method used by the attackers is Distributed Denial attack and SQL based attacks. Generally, attackers spoof their IP address (es) to launch attacks for the sake of hiding their own hosts [2] and their current geographical location. The spoofed IP address(es) could be fixed or random. Moreover, to avoid possible anti-spoofing mechanisms, the attackers can also launch the attacks from a botnet using non-spoofed IP addresses (fixed IP addresses) [3]. While detecting attacks based on IP addresses is crucial for defending against DDoS attacks, many detection methods do not defend against DDOS attacks. Several recent DDoS attack detection approaches, while successfully identifying DDoS attacks, fail to identify the balance it and failed to filter the malicious request filtration. As a result, the victims cannot initiate appropriate defensive measures. The approaches which detect DDoS attacks based on connections between two network hosts, but they are not suited to detecting attack if it is done by the Random address attack mode. Motivated by above challenges, we concentrate on how to detect TCP-based DDoS attacks under the filtering and scanning the header data. In this paper, a server-end detection approach is proposed, which uses the decision tree technique to achieve a high detection rate and low false alarm rate along with Incident Response action can be taken. The following are the contributions of this work various sectors but now we are concentrate in only the two modules that input filtration and the scanning of the header part of the packet to count the request and response from the server and the client for the particular time bound limit, in real time this server take account the all the request and response from the client and server the timebound limit is set the admin if the packet limit is exceed the request or response the limit does not exceed the packet is allowed to execute the process after it finishing it input query filtration

II. METHODS

This server backend is handle by the PHP7.0 with the database MYSQL , here the client side script and the API access is provide by the java script based Content Delivery Network is hosted in our company domain , JS file delivered by the CDN is called the point script , the script library is worked based on the Asynchronous based program , which the program does not stop, the execution unless it completely get the output or get the connection timeout error, for the evaluating purpose the Data Base related operations , form data collection based work is completed . The virtual server environment is created by using the .Net C# based software for the other validation like port listening and the other request data filtration to test the efficiency.

III. PROPOSED

The Point Script concept is execute the program from the CDN provide the JS , since the architecture is implementation of the server architecture, in its virtual environment using the C# .net framework base Windows Presentation Form software .



ARCHITECTURE DIAGRAM

The above shown picture describe the architecture of the point script server while not only the security architecture the point script is also concentrate in the various parts in web development to prevent DDOS based attack and the SQL based attack using Point Script

3.1 Point Script

The fundamental unit of blocks of operation of the server is stared from the CDN hosted point script the used each line of the code is treated as argument in this compiler the argument name and its number of the argument and the argument datatype is here compare to execute the syntax, if the syntax match the compile the called argument is passed and return the output the output returned here in the two modes

A**) Return by call Back:** In this method the return data is passed in to the call back function that passed in the argument as the third parameter created by the developer or the user in the argument, because the return by id simply display the output result to the passed id name, the future process is required before returning the acquired data from the CDN, user can use the Return by call back method.

Egg: functionname_fun (parameter1, parameter2, callback)

B) **Return by ID:** The second argument method is return by the id method here the html element id attribute is passed, which the element wants to display the result to the user here, there is no future process is takes place, and it directly returns the data to the particular input element by its ID.

Egg: functionname (parameter1, parameter2,"id")

Note: when passing the id as the third argument, the argument must be sent as the string datatype

3.2 QUERY BUILDER

The query is separate module in point script here the query for the database is created , which requirement and process of the Database can be get from the user client side the respected method is called by the user by the respective arguments is passed.

The query builder is take care of the function such as creating the tables, inserting data, alter the structure of the table, truncate and the drop the row from the table, the library of the architecture can either download from the git hub or use make the availability of fully core functionality of non minified framework user can use through the PHP eval () method which pass the PHP code as string from the response data return from the server

The non-minified core functionality provide the full supported database come along with framework with support for storage of blob data, as the result the query and the input validation of the server is ensured, in the non minified JS file the database support is provide from the CDN hosted server if the database require the user can use the functionality, to view and edit the database manually, the user should have to create an account point script CDN.

3.3 DATABASES

In the point script the MYSQL database is used , to execute the database from the server side the PHP built in module is already embedded in the library , the library is called by using AJAX method , in this method it also have the two modes of call

A) Return by id The return by id return the response code of the 200 if the query is executed in the backend database without error, the following response codes are replied as according to the result

a) 200 – query executed no error
b) 404 – the requested file is not found

B)Return the result by call back The returning the result by the call back the call back or the method is passed as the third argument , the response data is passed to the method , of the past call back and the result is display after completing the future document.

3.4 PHP MODULE

The PHP module is the set modules that contains the php files which takes of server side data processing, the each of the php files present in the is call by Ajax methods from the modules all the modules can have default input filtration before processing the request in the server, so the it is safeguard from the SQL injection attacks.

The php module in the database side can take place the below following functionalities the all mentioned in below

a) Creating tables
b) Insert data in tables
c) Alter table structure
d) Drop the data in table
e) Truncate the table

The above mentioned functionalities of the php module process the input filtration and sanitarization, so the architecture of the server is free from the SQL injection, when compare to the current server architecture.

VI. ATTACKS

4.1SQLAttack

Due to the default input filtration, of the input using SQL command table, it automatically safeguard from the SQL injection attack which prevent the server by using default server architecture

4.2 Broken Authentification Attack

Today the broken authenification is the one of the most important method to break the security of the web server or particular user account by executing the malicious code from the client side , it is prevent by filtering the command using eval() method by the query builder.

4.3DDOSAttack

Due to the header part of the each and every packet is scanned and each every request and response count is noted in the server if the bound limit is exceed it stop the response and stop receiving the request by stop listening in the particular IP

CONCLUSION:

In this paper we have explained that the secured server architecture which has the input filtration by the default , and reading the data in the header part to note the count of the bound limit in the server it prevent from the DDOS attack

REFERENCE:

[1] "The top 10 Dodos attack trends," http://www.imperva.com/docs/DSEncapsulate The Top 10 Dodos Attack Trends ebook.pdf.
[2] N. Long and R. Thomas, "Trends in denial of service attack technology," CERT Coordination Centre, 2001.
[3] L. Colace, G. Masini, V. Cencelli, F. Denotaristefani, and G. Assanto, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," ACM Computing Surveys, vol. 39, no. 3, pp. 273–302, 2007.
[4] M. L. Sang, S. K. Dong, J. H. Lee, and J. S. Park, "Detection of DDoS attacks using optimized traffic matrix." Computers & Mathematics with Applications, vol. 63, no. 2, pp. 501–510, 2012.
[5] T. Thapngam, S. Yu, W. Zhou, and S. K. Makki, "Distributed denial of service (DDoS) detection by traffic pattern analysis," Peer-to-Peer Networking and Applications, vol. 7, no. 4, pp. 346–358, 2014.
[6] M. H. Bhuyan, A. Kalwar, A. Goswami, and D. K. Bhattacharyya, "Low-rate and high-rate distributed DoS attack detection using partial rank correlation," in Fifth International Conference on Communication Systems and Network Technologies, 2015.
[7] H. J. Kashyap and D. K. Bhattacharyya, "A DDoS attack detection mechanism based on protocol specific traffic features," in International Conference on Computational Science, Engineering and Information Technology, 2012, pp. 194–200.
[8] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting DDoS attacks against data center with correlation analysis," Computer Communications, vol. 67, no. C, pp. 66–74, 2015.
[9] S. Behal and K. Kumar, "Detection of DDoS attacks and flash events using novel information theory metrics," Computer Networks, vol. 116, pp. 96–110, 2017.
[10] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Computers & Security, vol. 31, no. 3, pp. 357–374, 2012.
[11] "The CAIDA UCSD DDoS attack 2007 dataset," http://www.caida.org/ Data/passive/ddos-20070804 dataset.xml.