DDoS Attack detection and Prevention in Private Cloud Environment

Shahanaz Begum I
Department of Computer Science and Engineering
BIT Campus, Anna University, Tiruchirappalli, Tamilnadu, India
[1]shasmile23@gmail.com

Geetharamani G
Department of Mathematics
BIT Campus, Anna University, Tiruchirappalli, Tamilnadu, India
[2]geeramdgl@rediffmail.com

Abstract- DDoS Attack is launched by attackers by exploiting all the vulnerabilities in the present system design. The sole dependence of the Internet for many of the activities is also a main reason for the losses that are faced by the majority of the human population. In spite of the great amount of services that are offered by the Internet it also has the responsibility of providing security measures against the malicious activity that are caused by the attackers. The Web Applications that are deployed in the Server architecture are liable to get exposed to the DDoS Attack apart from other attacks such as SQL Injection, XSS, CSRF, etc. It is disappointing that for the eradication of this DDoS attack, the vulnerabilities causing this attack to get launched are not completely traced. To solve this problem, in this current work the countermeasure against this situation was developed. The experiment involves in performing the DDoS Attack targeting the Web application deployed in Apache Web server in Private Cloud Environment. To correctly differentiate the DDoS Attack traffic from the normal traffic flood , the threshold of the attack traffic attempted on SCO Website was utilized. The Hit rate was calculated from the NASA datasets and compared with the current work. The results obtained from Tableau graph with the help of NASA datasets for Flash crowd traffic were utilized in understanding the traffic patterns available with access matrix. The DDoS Attack was performed using the attack causing tool and with the help of java coding to analyze the attack patterns stored in the log files of Tomcat Webserver. The Attack was detected at a faster rate once the threshold had reached the higher traffic limit realized in the 1998 World cup Football datasets and the Attack traffic was stopped immediately through Servlet and XML Coding and the results were validated.

Keywords – ***Detection of Intrusions, Distributed Denial of Service Attack, Private Cloud Environment, Web Application***

I. INTRODUCTION

Researchers are fighting against Distributed Denial of Service (DDoS) Attack as it threatens the users of the Internet service. Though IP Spoofing has got defensive mechanisms, DDoS attack which is deployed in a widespread environment has never found successful countermeasures.

Application layer DDoS Attacks exhaust the server resources such as Sockets, CPU, Memory, disk/database bandwidth to make unavailable a legitimate user's services [2]. A well organized, widely scattered, remote network encompassing Zombies or Botnet computers institute a DDoS Attack for simultaneously and continuously send a traffic flood towards the target system. The target system is made to respond slowly so as to become unusable or to crash completely [1,3]. Botnet has compromised computers called Zombies which are created with the help of malicious softwares such as worms, Trojan horses or backdoors [4]–[6]. The resources of well organized networks are utilized for launching the DDoS Attacks. Adding to this there is a complexity involved for the defense mechanism to trace the original attacker because of the use of fradulent (i.e., spoofed) IP addresses by zombies under the control of the attacker [7].

The majority of the DDoS flooding attacks so far that have been attempted are with the purpose of denying the services to the victims which leads to revenue losses and increased costs of mitigating the attacks and restoring the services.

This current work analyzes the NASA datasets to learn the threshold that can be used as a reference for identifying the DDoS Attack traffic. The different Virtual Machines(VM) are used for deploying the Apache Web Server, the attacking tool in the VM where the Web server resides. The Web application is deployed in the Web server which is targeted by the DDoS Attack. The countermeasure is provided to dynamically stop the attack by checking the traffic threshold whether it has crossed the limit set for DDoS Attack the value obtained from 1998 World Cup Football datasets. The IP Address associated with the originator of this attack is blocked. This blocking has been experienced in a shorter time period compared to the attack that was mounted on a Real Website where CloudFlare Firewall was provided. The rest of the paper is organized as follows. Section II deals with the Related work in the domain of DDoS Attacks. Section III describes the proposed work and the experiment carried out related to the current work. Section IV discusses on the results obtained after the conduction of the experiments. Section V gives concluding remarks.

II. RELATED WORK

Solving DDos Attack is an overwhelming security challenge to the large and small organizations. Cataldo Basile et al described the importance of the proper configuration of firewalls, the error in which leads to lapse in providing security and availability [12]. Their work stated that the incorrect configuration of the firewall may also lead to introduction of vulnerabilities.

Supranamaya Ranjan et al. proposed a mechanism involving DDoS Shield to protect server's resources. DDoS Resilient was utilized in their work to schedule client's requests based on the continuous value assigned to the client session [8].

Meixing Le et al. presented an IDS where they monitored both web and database requests. They reduced the false positive rate for static websites but the reduction in the same posed a problem for dynamic websites. They could not mitigate DDoS Attack which could occur in Server architecture [14].

Yi Xie and Shun-Zheng Yu described the application layer DDoS Attacks utilizing legitimate HTTP requests. They used the concept of access-matrix dynamics to detect the attacks [16]. The countermeasure for this DDoS Attack is not proposed in this paper.

Vigna et al. analyzed the web requests with malicious behavior with the help of an Intrusion Detection system called WebStat [9]. They effectively detected the web-based attacks in general by correlating the attacks in terms of states and transitions .

Farzaneh Geramiraz et al. proposed an adaptive Anamoly based IDS to detect new attacks in the Networks using Fuzzy Controller. This technique was helped in improving detection accuracy [10].

David Gillman et al. introduced Secure delivery networks which acted as a shield against many attacks [11]. They discussed their work with the case study ABABIL.

Shingo Mabu et al. described a fuzzy class-association rule mining method [13] for detecting the attacks and they analyzed their work with the help of Datasets.

Moniruzzaman et al. conducted experiments with Open Nebula Cloud management platform to measure the performance of load balancing feature of the cloud platform [15].

The current work describes how DDoS Attack traffic is blocked in the private cloud environment by creating different VMs making use of the virtualization concepts. The performance of this work is evaluated with the help of DoS Attack performed through the Java Coding on the Templates offering website which is currently being used. That particular website blocked the source of attack after a time period of 10 min because of the high end system provided at their end to host their website and with the support of Cloud Flare Web application Firewall. In this current work the attack was blocked with the help of Servlet and XML Coding by checking against the threshold of requests, the value that was obtained from the World cup Football datasets. The time taken in blocking the attack was reduced to approximately 10 seconds and the results were validated

III. PROPOSED WORK AND EXPERIMENT CARRIED OUT

This current work involves the experimental setup for obtaining the DDoS attack datasets and normal user datasets. The experiment was carried out in the Private Cloud setup on Windows platform. In this setup as hypervisor ESXI was utilized. Three VMs were created and unique IP address was associated with each VM. VM1 had the Web Server installed in it. The Internet Connectivity was allocated to the VMs. The connectivity status between the different VMs was checked using ping command.

Private Cloud setup was obtained with different Virtual Machines. In one VM the Apache Web Server was installed to collect the attack traffic related datasets. Second VM had the Banking application deployed in the Web server. This attack was attempted with the help of a DDoS Attack creating Tool and with the aid of Java Coding. The third VM had the JMeter the Performance Testing software, which created legitimate traffic accessing the web application.

Both these legitimate traffic caused by JMeter and Attack traffic caused by the Attacking tool were considered as the Training datasets. A portion of the 70% of the datasets was used as the training and the remaining as the testing datasets.

After obtaining sufficient datasets Access Matrix is analyzed from the logs of the Web server using the following information. Information such as IP address of the user, the request time, the web site that was accessed etc were collected.
Document Popularity is defined by the Request hit rate as

Pit = bit / (Sum of bit's for all the requests that were made for the documents ranging from i=1 to N)

where bit is the number of the requests made for accessing the ith document at the ith time unit, and N the number
of the Web server's documents.
It was observed from NASA datasets the Average count of page hits per day as 23,420 and overall request rate per second as 37.
The access matrix traffic patterns obtained for the earlier work was compared with the patterns obtained for the current work. The results obtained from the plotting of the Tableau graphs using the NASA datasets were utilized for analyzing the traffic patterns stored in the access matrix. The DDoS Attack tool and the Java coding were utilized to perform the attack in order to analyze the attack patterns stored in the log files of Apache Web server. The attack was detected at a faster rate once the threshold had reached the higher traffic limit which was realized in the 1998 World cup Football datasets and the attack traffic was stopped immediately through Servlet and XML Coding. The results were validated.
The VMs were associated with unique IP Addresses. The attack was targeted at the Web Application hosted in the VM1 of the Private Cloud. The attack creating tool was executed in VM2. The JMeter was executed in VM3 to create legitimate traffic targeting the Application. The log files obtained from the Web Server had the traffic patterns from different users with different IP Addresses. The JMeter installed outside the Network were also allowed to access the Webapplication hosted in the Apache Webserver in VM 1. Even the DoS Attack tool was executed outside the Private Cloud targeting the Application deployed in the Webserver in VM1 to obtain attack traffic originating from different IP Addresses.
VM2 had the DoS Attack creating tool and with the help of the Servlet and XML Coding the attack that targeted the Web Application, deployed in the Apache Tomcat Server in VM1 was controlled. VM3 had the JMeter installed in it to create legitimate traffic targeting the Web application in VM1. In this current work a promising DoS Attack tool was utilized to mount the Attack traffic targeting the Web Application deployed in the Apache Tomcat Server. The logs stored in the Web Server for both legitimate traffic and Attack traffic were collected. The Access matrix had the traffic history obtained from NASA Websites. The popularity of any Website attracts many clients with different IP Addresses to access the different documents available in a WebServer. The details such as how many users accessed a particular Web site for how long ie for a week's duration or for a month's duration helps in analyzing the traffic patterns captured in Access matrix. This observation was with reference to the Flash crowd traffic. The threshold of flash crowd traffic was obtained from the NASA Datasets and the calculated hit rate for this traffic was compared with the legitimate traffic created using JMeter targeting the Web Application in VM1.
The DDoS Attack datasets were obtained from 1998 World Cup Football datasets. The maximum attack threshold obtained from the World Cup Football datasets was cross checked with the maximum traffic limit that was got for attack traffic created in this current work through Java Coding.
The attack traffic was created using both the Attack tool operated from VM2 and the Java Coding for mounting Attack traffic. The graphs were drawn depicting how quickly the DDoS Attack was blocked with the help of the countermeasure provided at the Servlet and XML Coding level.

The Pseudocode for the blocking of DDoS Attack is given below.
For each Request to the Webserver from attack agents and legitimate users the following steps were followed to block the attack traffic.

```
Input:  Request r, Request-ip c , Timestamp t , No.of requests n, RequestPage p
Output: Response /Block Message
    foreach 'r'

      do get c , t , p
    Initialize t1 (Time limit with respect to t)
          Initialize n
      If  c && p repeats
            begin
              n++
          (n< v ) && (t < t1)
            server responds
            end
      elseif
```

```
begin
    n++
    (n >= v ) && (t < t1)
            block r from c
end
```

The graphs in Figure 1 and Figure 2 depicted the time taken in blocking the attack traffic versus Number of requests generated during a particular time period.

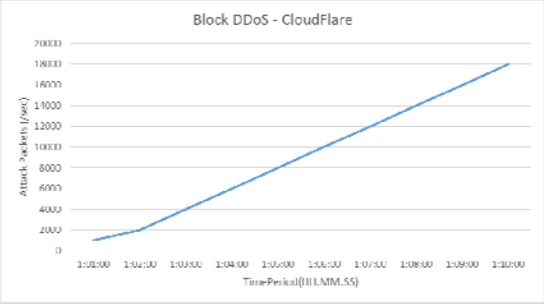

Figure 1

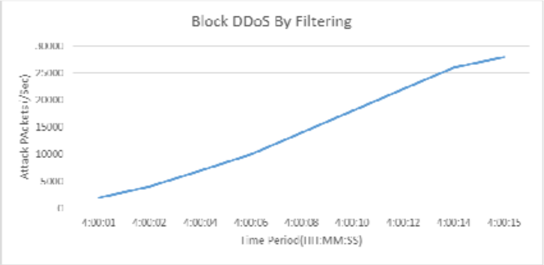

Figure 2

This gives a comparative study of the benchmark attack traffic whose threshold obtained from Worldcup Football datasets, with the block event occurred against the threshold value reached by DoS Attack caused by Java coding in the current work.

The performance of this current work ie the proposed countermeasure is evaluated with respect to the metric, the time taken in blocking source of the traffic originating the Attack. It is found to be convincing with respect to the attack traffic that was attempted focusing a Real Website which took more time in blocking our IP Address. In their website they had a licensed software Cloudflare which could withstand an approximate Attack traffic flood of approximately 8GB. The attack was realized and blocked only after some 10 minutes in that popular Template generating website whereas in this current work the attack traffic was blocked with the Servlet and XML coding executed in VM1 approximately within 10 seconds. This was achieved by checking against the threshold value dynamically for the given ip address, number of requests and for the given Website.

IV RESULTS AND DISCUSSIONS

In this current work the successful attempt was made in blocking the DDoS Attack traffic through Servlet and XML Coding executed in VM1. The experiment was conducted in the private Cloud Setup using ESXI as the hypervisor. The 3 VMs were created and each VM was assigned a Unique IP Address. The legitimate Flash flood traffic was obtained using JMeter installed in the VM3. The log files from the Web server recorded the legitimate traffic datasets. The DDoS Attack traffic was created targeting the Web Application hosted in the Web Server in VM1. The datasets for both Flash crowd traffic created using JMeter and those obtained from NASA datasets were compared for the maximum traffic requests. The hit rate was calculated for the legitimate flash crowd traffic created in both the cases. The historical datasets recorded in NASA Websites were utilized to obtain the threshold value for the maximum traffic flood.

These datasets also gave an insight about the legitimate IP Addresses that were utilized by the different Clients. These IP Addresses were used as a reference to identify the spoofed IP Addresses in the DDoS Attack traffic got from the 1998 World cup Football datasets. This gave an idea on filtering those HTTP requests with these IP Addresses stored in the log files indicating that the requests utilized IP Spoofing. The performance of this current work ie the countermeasure that was developed using Servlet and XML Coding in blocking the attack traffic at the shorter time was validated with the actual attack that was originated from the known IP Address to a famous Template generating Website. The time taken for blocking this attack was found to be more than the time taken in blocking the attack through appropriate Java coding that was proposed in the Current Work. Hence the effort taken in immediately stopping the attack was found to be encouraging and satisfactory.

V.CONCLUSION AND FUTURE WORK

The approach that was attempted in this current work was successful in blocking the DDoS Attack traffic through Servlet and XML Coding executed in VM1. The experiment was conducted in the private Cloud Setup using ESXI as the hypervisor. The 3 VMs were created and each VM was associated with a Unique IP Address. The DDoS Attack traffic was created targeting the Web Application hosted in the Web Server in VM1. The datasets for both Flash crowd traffic created using JMeter and NASA datasets were compared for the maximum traffic requests obtained in each case. The hit rate was calculated for the legitimate flash crowd traffic created in both the cases. The performance of this current work was validated with the actual attack that was originated from the known IP Address to a famous Template generating Website and the time taken for blocking this attack was found to be more than the time taken in blocking the attack traffic in the Current Work. Hence the effort taken in immediately stopping the attack was found to be encouraging.

It was found out that the IP Address that was blocked would be an intermediary IP Address not the original source of the Attack in the case of Botnet. Hence as a future work it was planned to trace back the attack traffic to identify the root originator of the DDoS Attack traffic.

REFERENCES

[1] J. Mirkovic and P. Reiher, *A taxonomy of DDoS attack and DDoS defense mechanisms"*, ACM SIGCOMM Computer Communications, Review, vol. 34, no. 2, pp. 39-53, April 2004.

[2] S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly, *DDoS-Resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection*, IEEE INFOCOM'06, 2006.

[3] R. K. C. Chang, " *Defending against flooding-based distributed denial of service attacks: A tutorial"*, Computer J. IEEE Commun. Magazine, Vol. 40, no. 10, pp. 42-51, 2002.

[4] R. Puri, *Bots and Botnet – an overview*, Aug. 08, 2003, [online] http://www.giac.org/practical/GSEC/Ramneek Puri GSEC.eps

[5] B. Todd, "*Distributed Denial of Service Attacks"*, 2000,[online] http://www.linuxsecurity.com/resource files/intrusion detection/ddos–whitepaper.html.

[6] CERT, *Denial of Service Attacks*, 2001, [online] http://www.cert.org/tech tips/denial of service.html.

[7] J. Liu, Y. Xiao, K. Ghaboosi, H. Deng and J. Zhang, "*Botnet: Classification, Attacks, Detection, Tracing, and Preventive Measures"*, EURASIP J. Wireless Communications and Networking, vol. 2009, Article ID 692654, 11 pages, 2009.

[8] Supranamaya Ranjan, Ram Swaminathan, Mustafa Uysal, Antonio Nucci and Edward Nightly, "DDoS Shield : DDoS –Resilient Scheduling to Counter application layer Attacks", IEEE/ACM Transactions on Networking, 2008.

[9] Giovanni Vigna William Robertson Vishal Kher Richard A.Kemmerer, " A Stateful Intrusion Detection System for World-Wide Web Servers, IEEE Proceedings of the 19th Annual Computer Security Applications Conference, 2003.

[10] Farzanesh Geramiraz, Amir Saman Memaripour and Maghsoud Abbaspour, "Adaptive Anomaly-Based Intrusion Detection System using Fuzzy Controller", International Journal of Network Security, 2012.

[11]. David Gillman, Yin Lin, Bruce Maggs and Ramesh K.Sitaraman, "Protecting Websites from Attack with Secure Delivery Networks", IEEE Computer Society, 2015.

[12] Cataldo Basile and Antonio Lioy, " Analysis of Application-Layer Filtering policies with Application to HTTP", IEEE/ACM Transactions on Networking , 2015.

[13] Shingo Mabau, Ci Chen, Nannan Lu, Kaoru Shimada, " An Intrusion-Detection Model based on Fuzzy Class- Association Rule Mining Using Genetic Network", IEEE Transactions on Systems, Man and Cybernatics", 2011.

[14] Meixing Le, Angelos Stavrou, Brent ByungHoon Kang, DoubleGuard: Detecting Intrusions in Multitier Web Applications,‖ IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 4, pp. 511-524, August 2012.

[15] Moniruzzaman, Kawser Wazed Nafi and Syed Akther Hossain, "An Experimental study of load balancing of OpenNebula Open-Source Cloud computing platform" , 3rd International Conference on Informatics, Electronics and Vision, 2014.

[16] Yi Xie and Shun-Zheng Yu, "Monitoring the Application-Layer DDoS Attacks for Popular Websites, IEEE/ACM Transactions on Networking , vol.17, no.1, 2009.