Simulation and analysis of DoS attack in cloud environment

Rajakumaran Gayathri* and
Venkataraman Neelanarayanan

School of Computing Science and Engineering,
VIT University Chennai,
Chennai, India
Email: gayathri.r@vit.ac.in
Email: neelanarayanan.v@vit.ac.in
*Corresponding author

Abstract: Cloud computing is a dynamic environment in terms of both topology and network technology. The resources in the cloud environment are scalable and it is offered to the customer's on-demand which leads to a drastic increase in the customer's count. The key design feature of the internet and the protocols used to access the cloud services makes it vulnerable to various security issues. Out of the critical security issues, denial of service (DoS) ranks first since it disrupts the availability of cloud services. This paper aims to provide a detection mechanism for DoS attack in the cloud computing environment using the simple network management protocol (SNMP). SNMP is efficient in detecting the TCP-SYN flood attack in the cloud environment. Impact of the attack is estimated in terms of the system parameters CPU utilisation, I/O and latency.

Keywords: DDoS; simple network management protocol; SNMP; TCP-SYN.

Reference to this paper should be made as follows: Gayathri, R. and Neelanarayanan, V. (2017) 'Simulation and analysis of DoS attack in cloud environment', *Int. J. Knowledge Engineering and Soft Data Paradigms*, Vol. 6, No. 1, pp.52–61.

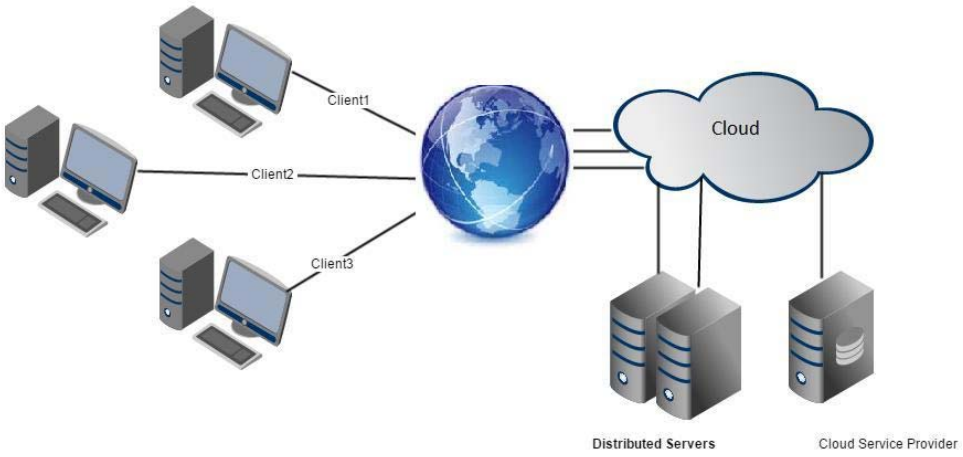
Biographical notes: Rajakumaran Gayathri received her BTech (CSE) from Rajiv Gandhi College of Engineering and Technology and MTech from Pondicherry Engineering College in 2011, Puducherry. She is currently employed at VIT University Chennai campus as an Assistant Professor in the Department of Computing Science and Engineering. Her research interests include cloud computing, information security and network security.

Venkataraman Neelanarayanan is an Associate Professor in the School of Computing Science and Engineering, VIT University, Chennai, India. He received his PhD in Computer Science, IT University of Copenhagen, Denmark in 2007. His research interests are distributed computing, cloud computing, grid computing, network management and security, context – aware computing.

1 Introduction

Cloud computing is a technology which comprises the existing concepts of web services, grid computing, network computing, utility computing and pervasive computing which aims to provide flexible services to users. Yu et al. (2014) found that cloud computing is developing rapidly in both academics and industry due to its essential characteristics, which includes on-demand self-service, resource pooling, rapid elasticity, and measured service. It paves a way for the industries irrespective of their function, to increase their business capabilities on the fly without investing much in the new infrastructure, training of personals or licensing. It follows a simple pay as you go model which enables any organisation or industries to pay only for the resources utilised by them. Cloud is a collection of computers and servers which can be accessed through the Internet using various kinds of browsers, protocols and platforms. Platform provides a way to deliver the cloud services to the customer.

Figure 1 Components of cloud (see online version for colours)



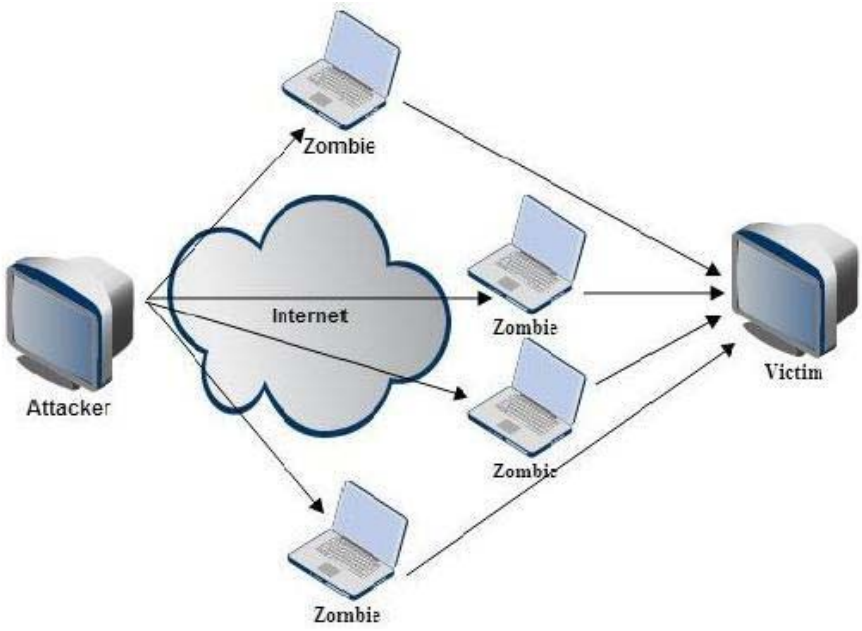
Somani et al. (2015b) found that cloud computing offers numerous services on different abstraction levels namely software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). In SaaS, the applications will be hosted by the cloud service provider and made available to the clients through a web browser or program interface. PaaS offers platforms for the customers to develop, to deploy and manage applications by eliminating the complexity of building and maintaining the infrastructure required for launching an application. IaaS provides virtualised computing resources to the customer on-demand. The tasks handled by IaaS CSP's includes system maintenance, backup, dynamic scaling, desktop virtualisation, administrative tasks, policy-based services and resiliency planning. The cloud services can be deployed as public, private, hybrid or community cloud. In public cloud, the services are rendered over a network that is open for public use. Few of the leading public cloud players are Amazon web services

(AWS), Google, Microsoft and Rackspace. Despite of the rapid emergence of cloud computing, security requirements plays a vital role in the cloud core. Such requirements include confidentiality, integrity, availability, accountability and privacy preserving. Among these, availability is identified as a crucial requirement for the on-demand provisioning of services.

2 Related work

Denial of Service (DoS) attack and distributed DoS attacks are the dominant players which disrupts the availability constraint in the cloud environment. Somani et al. (2015a) explained that DDoS is an attack which disrupts the legitimate users from accessing the cloud services by flooding the target server with bogus packets which is represented in Figure. 2. Armbrust et al. (2009) and Subashini and Kavitha (2011) suggested that the DDoS attacks should be given higher priority since it leads to a drastic increase in service demands.

Figure 2 Distributed DoS attack (see online version for colours)

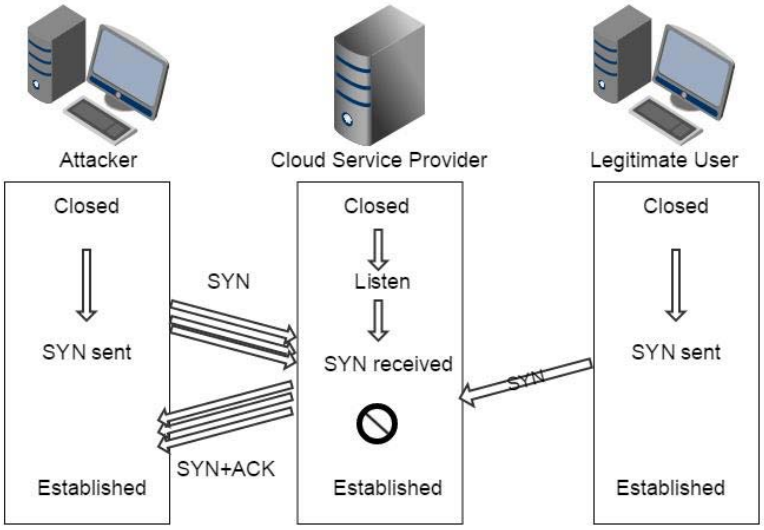


Common DDoS attack categories described by Bhadauria et al. (2011) includes

- SYN floods
- DNS amplification
- malformed TCP, UDP packets.

Among these, TCP SYN flood attack is focused in this paper. In this attack, the usual routine of the TCP three-way handshake is exploited to consume the resources on the target server to make it unresponsive for legitimate client request.

Figure 3 TCP-SYN flood attack (see online version for colours)



TCP SYN flood attack is represented in Figure 3 which illustrates the overview of attack process. In a three-way TCP handshake process, when a client attempts to initiate connection with the server need to exchange the series of messages to complete the handshake process.

- The client requests the connection by sending the SYN message to the target server.
- The server needs to send acknowledgement SYN-ACK to the client.
- The client replies with ACK.
- Connection established between client and server.

It performs spoofing on the IP address of the legitimate client and initiates series of SYN requests which makes the server to send the SYN + ACK to falsified IP which will send back an ACK as it never sent such SYN requests. The objective of this paper is to detect and prevent DDoS attacks in the cloud computing environment. DDoS Detection is achieved through the simple network management protocol (SNMP). Lin et al. (2014), Hu et al. (2014), Xia et al. (2015), Sezer et al (2013) and Azodolmolky et al. (2013) explored the concepts of software defined networking (SDN) which has attracted greater interest since it is used to acquire control on the network environment. SDN offers great opportunities to defeat the DDoS attack in the cloud platform.

3 DoS attack motivation

According to worldwide infrastructure security report drafted by Arbor Networks (2016), the frequency of DoS attack has increased by 60 times in 11 years. The tabulation shows the loss incurred due to the DDoS attack.

Table 1 DDoS attack motivation (see online version for colours)

Survey name/affected	*Average financial loss/target*
Kapersky Labs 2014	444,000 USD
Verisign iDefense Security Intelligence Services	66K USD/hour
Greatfire.org	30K USD/day
HSBCC 2016	Online services
Yahoo 2014	Customer information
BBC 2016	BBC server

4 DoS detection method

Detection of DoS attack can be incorporated by numerous mechanism and plenty of tools are available for the same. Few detection measures considered for comparison are wireshark and Netflow. Wireshark has the ability to capture the packets passing through any port (https://www.sans.org/reading-room/whitepapers/detection/denial-service-attacks-mitigation-techniques-real-time-implementation-detailed-analysi-33764). The time taken by the packet to reach destination and return is the round trip time (RTT) which could be monitored with the help of this tool. The attack detection could also be done by examining the packet structure and flags analysis. The malformed packet information will be retained in the frame structure of the incoming packets. The drawback in using the Wireshark tool is it requires access to the switch which causes problem as all the switches will not support full spanning. Another major drawback is that, the security related software installed in the computer systems conflicts with the packet capturing process, which makes only the incoming packets visible.

Netflow is capable of monitoring traffic on any direction, but to attain maximum efficiency, it should be configured to monitor incoming packets. This protocol could be turned on for all interfaces. The limitations of Netflow are

1 devices only export two flows

2 Netflow overhead can overtax infrastructure

3 visibility limited to routed traffic

4 non-Netflow capable devices are blind to local traffic.

SNMP is proposed which helps in overcoming drawbacks in the existing detection measures. SNMP is a network management protocol which allows the exchange of information on the management of a resource across a network. It enables resource monitoring if required and initiates action based on the data obtained from the monitored resource. Wide range of network devices such as servers, workstation, printers, router, bridges and hubs can be monitored through the SNMP protocol. The components of SNMP work collaboratively to detect an intrusion in the network infrastructure which is mentioned as below

- management Information Base (MIB)
- SNMP agent

- SNMP manager
- network management system (NMS).

4.1 Management information base (MIB)

The MIB is a data structure which stores a list of resources to be monitored in a hierarchical order. Each resource is stored with a unique object identifier (OID) and attributes. The OID value cannot be changed once it is assigned. Each OID consists of sequence of integers. The sequence will be helpful in locating the position of the object in the MIB tree. SNMP commands are used to retrieve management information on the resource.

4.2 SNMP agent

SNMP agent is an network management software module that resides in a device which needs to be managed. An agent has knowledge about the management information and translates it into a SNMP compatible form.

4.3 SNMP manager

SNMP manager is responsible for communication and information retrieval related to the network status. The key functions of it are listed below

- query the agent
- obtain responses from the agent
- set related variables in agent
- acknowledges the events in the agent.

4.4 Network management system

The NMS presents interface to the user. It retrieves required network status information using the SNMP manager and provides it to the user.

5 Web server threshold prediction

To predict the regular workload pattern on a web server, an experimental testbed setup is done in the Amazon AWS console. The server was a PC Intel® Xeon® 2.40GHZ at 1.00 GB of RAM. The two clients were both PC's of the same configuration. Apache 8.0 is installed in the web server. To estimate the request handling capacity of a web server under various loads, the files chosen are text files. Different loads on the web server are simulated by making the client machines to observe the request handling capability of a web server httperf tool is used. Httperf is a tool for measuring performance of a web server under various loads. The distinguishing characteristics of httperf are

- robustness
- ability to generate and sustain overload
- support for HTTP and SSL protocols
- extensible to new workloads.

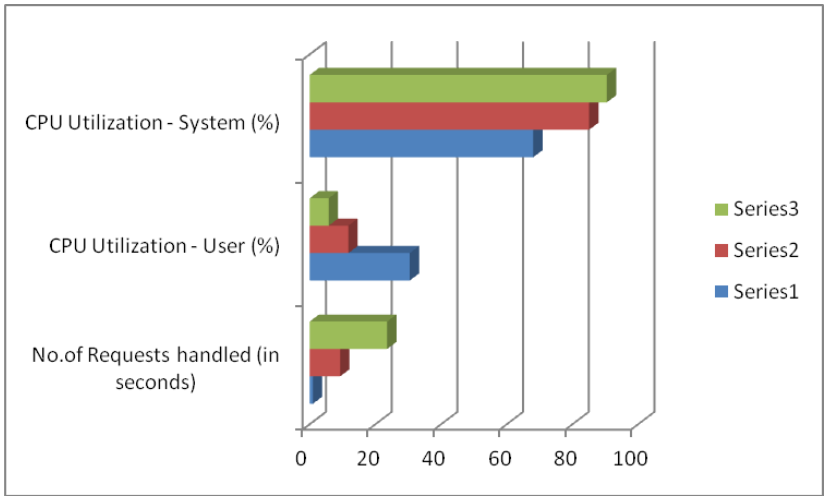
Table 2 Web server capability estimation using httperf

SL. no.	*Parameters observed*	*No. of connection attempts, no. of requests per attempt*			
		100, 1	*100, 10*	*100, 100*	*1,000, 1*
1	Connection request	100	100	100	1000
2	Test duration (seconds)	99.962	10.786	4.329	721.924
3	Concurrent connections	< 6	< 24	<= 100	<= 12
4	CPU utilisation (%)	98.6	96.9	96.2	99.01
5	Request handling capability	1	9.3	23.6	1
6	Connection size (KB)	65	65	65	65
7	Average connection time (ms)	1,260.1	1,306.9	1,576.6	1,375
8	Net I/O (kb/s)	0.3	3.0	7.7	0.3

Note: Text requests.

The graph is obtained by retrieving the requests corresponding to 100 from Table 1. The total request packets chosen for the scenario is 100 and varies in the rate of sending the packets to the target server. With increase in the request handling capability of the web server, the CPU utilisation of the system increases gradually and reaches a maximum of 90.5%. In order to arrive at the exact CPU utilisation %, where the web server starts to rejects the client request, the same experiment by sending 1,000 request to the web server by varying the rates as 1,100 and 1,000.

Figure 4 Request handling capability estimation for 100 requests (see online version for colours)

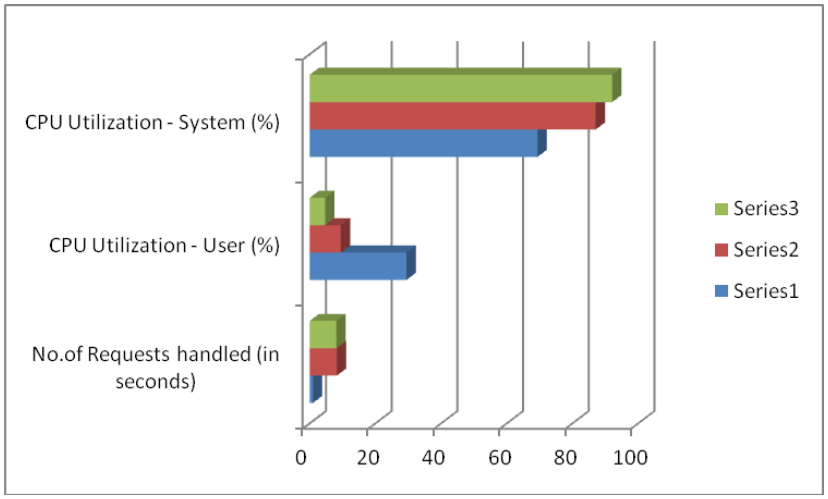


For generating Figure 5, the total request packets chosen for the scenario is 1,000 and varies in the rate of sending the packets to the target server. With increase in the request handling capability of the web server, the CPU utilisation of the system increases gradually and reaches a maximum of 92.1%. It is inferred that, beyond this point, the web server is incapable of handling further requests. It can be justified from the below obtained values.

The actual packets sent to the web server	1,000
Replies from the web server	499
CPU utilisation – System	92.1%

Hence, when the web server attains a CPU utilisation of 92.1%, it starts to reject connection attempts from the client. Based on the above inference, it is concluded that the maximum request handling capability of a web server of the specified configuration is 8.3 requests/ second for the text files of size 65 KB.

Figure 5 Request handling capability estimation for 1,000 requests (see online version for colours)



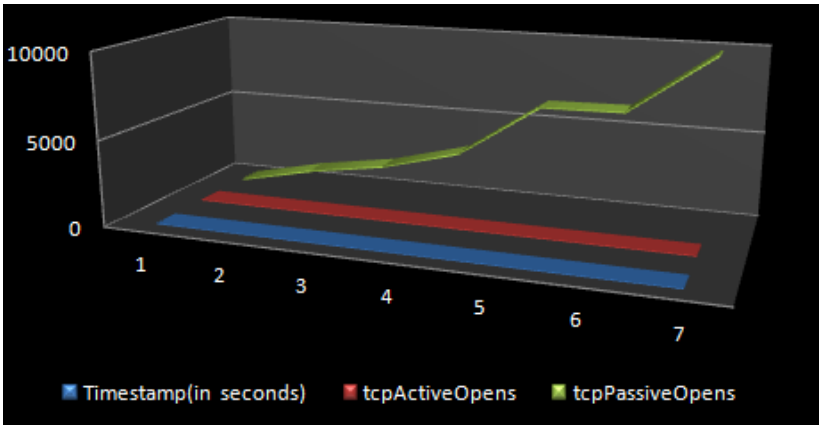
6 Results discussions

Hence, a new strategy of DoS simulation is carried out using the httperf tool in AWS platform. Two EC2 virtual instances are created in AWS. Windows server 2012 R2 is designated as a DoS attack target server. Another instance act as DDoS attack client machines. Httperf is installed in the EC2 virtual instances which sends TCP SYN packet to the destination server. SNMP is configured in the destination server to monitor the inbound traffic flow. Tcp variables tcpActiveOpens and tcpPassiveOpens are used to detect the DoS attack.

The state of the SNMP variables tcpActiveOpens and tcpPassiveOpens are observed before simulating the httperf tool which is assumed as a legitimate traffic flow. The state

of the variables is again noted once the httperf tool is initiated to send SYN packets. The change in the state of the variables are observed and represented as a graph.

Figure 6 DoS attack in AWS cloud (see online version for colours)



It is observed from the above graph that, if the attacker sends 100 packets at the rate of 100, within 7 seconds the CPU Utilisation reaches a maximum of 97.2%. The count of variables tcpPassiveOpens captures the half-open connections and tcpActiveOpens captures the established connections.

7 Conclusions

We have presented a DoS attack detection method using the SNMP variables and demonstrated its effectiveness in real time attack detection. The identified method is tested in a real time Amazon AWS console. System parameters were estimated to predict the regular workload pattern of a web server to test its performance under the legitimate network traffic. The same web server is loaded with DoS attack requests using the hping3 tool and the corresponding system parameters are tabulated. This paper discusses methods which serve as an effective measure in accurate identification and detection of DoS attack requests. The future enhancements include increasing estimating the capacity of a web server under various loads like video files, pdf files and other scripting files.

References

Arbor Networks (2016) https://www.arbornetworks.com/images/documents/WISR2016.Web.pdf.

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I. and Zaharia, M. (2009) *Above the Clouds: a Berkeley View of Cloud Computing*, February, EECS Dept., Univ. California, Berkeley, CA, USA, Tech. Rep.UCB/EECS-2009-28.

Azodolmolky, S., Wieder, P. and Yahyapour, R. (2013) 'SDN-based cloud computing networking', in *Proc. IEEE ICTON*, June, pp.1–4.

Bhadauria, R., Chaki, R., Chaki, N., and Sanyal, S. (2011) 'A Survey on Security Issues in Cloud Computing', *CoRR*, Vol. 12, No. 2, pp.10–17.

https://www.sans.org/reading-room/whitepapers/detection/denial-service-attacks-mitigation-techniques-real-time-implementation-detailed-analysi-33764

Hu, F., Hao, Q. and Bao, K. (2014) 'A survey on software-defined network (SDN) and openflow: from concept to implementation', *IEEE Commun. Surveys Tuts.*, Vol. 16, No. 4, pp.2181–2206, 4th Quart.

Lin, Y-D., Pitt, D., Hausheer, D., Johnson, E. and Lin, Y-B. (2014) 'Software defined networking: standardization for cloud computing's second wave', *Computer*, November, Vol. 47, No. 11, pp.19–21.

Sezer, S. et al. (2013) 'Are we ready for SDN? Implementation challenges for software-defined networks', *IEEE Commun. Mag.*, July, Vol. 51, No. 7, pp.36–43.

Somani, G., Gaur, M.S., Sanghi, D., Conti, M. and Buyya, R. (2015b) 'DDoS attacks in cloud computing: issues, taxonomy and future directions', *ACM Computing surveys*, December, Vol. 1, No. 1.

Somani, G., Johri, A., Taneja, M., Pyne, U., Gaur, M.S. and Sanghi, D. (2015a) 'DARAC: DDoS mitigation using DDoS aware resource allocation in cloud', in the *Proceedings of Springer*, December, pp.263–282.

Subashini, S. and Kavitha, V. (2011) 'A survey on security issues in service delivery models of cloud computing', *J. Netw. Comput. Appl.*, January, Vol. 34, No. 1, pp.1–11.

Xia, W., Wen, Y., Foh, C., Niyato, D. and Xie, H. (2015) 'A survey on software defined networking', *IEEE Commun. Surveys Tuts.*, Vol. 17, No. 1, pp.27–51, 1st Quart.

Yu, S., Tian, Y., Guo, S. and Wu, D.O. (2014) 'Can we beat DDoS attacks in clouds?', *IEEE Transactions on Parallel and Distributed Systems*, September, Vol. 25, No. 9, pp.2245–2254.