ISSN:0975-9646

DDoS Prevention on Rest Based Web Services

Neha Lad [1] Asst. Prof. Jwalant Baria [2]

[1]*Department of Information Technology, PIET,*
Gujarat Technological University, Gujarat, India
[2]*Department of Computer Science & Engineering, PIET Limda,*
Vadodara, Gujarat, India

***Abstract*— Web Service is the new paradigm for Internet Communication. They are becoming very useful technology today for business integration or inter enterprise communication. REST (REpresentational State Transfer) is an architectural style for designing web services, which provides a framework in web body components to save together and provide tremendous functionality for web applications. But they are often invoked by intruders. Intruders can be of any form. It can be a critical software, application bug or malicious user. One of the adverse attacks is DDoS (Distributed Denial of Sevice) attack. These attacks are rapidly mounted by cyber criminals to exhaust the network and system resources of the providers. REST proposes a lightweight approach to consume resources with no specific encapsulation, thus lacking of meta-data descriptions for security requirements. The proposed work provides prevention against DDoS attack on REST based web services. In the REST style, every resource is signified by a unique URL which may be operated on by a subset of the core set of HTTP commands: Get, Post, Put, and Delete. So DDoS can be easily performed on REST based web services. The proposed work will check the behavior of the IP address using number of requested URI and Time Interval Analysis based on the threshold.**

***Keywords*—**DDoS, REST, Web Services, SOAP

I. INTRODUCTION

Web services are designed to provide rich functionality for organizations and support interoperable interactions over a network. Web services are built based on Web service standards by using which Web service providers and Web service clients agree on a common Web service interface. Web services are language and platform independent: clients developed by using various languages and running on top of different platforms can communicate with the same Web service. Web services are said to provide loose coupling. A Web service application may include different services and that are each service is independent of each other. Modifying one of the services will not affect the other services. Each Web service is built based on Web service standard and therefore Web services are easy to integrate into a system. Web services are mainly realized in two ways: 1) SOAP-based services and 2) RESTful services [9].

Securing RESTful Web services is a dense endeavor: it involves securing the data, as well as the entire communication. One must prevent the confidentiality and integrity of data. The data in transit should be filtered for malicious payload. The communication should support authentication and access control, and ensure that the privacy of the communicating parties is not compromised.

Distributed Denial of Service (DDoS) flooding attacks are one of the biggest concerns for security professionals. A denial-of-service (DoS) attack is an attempt to make a computer resource (e.g. the network bandwidth, CPU time, etc.) unavailable to its intended users. To obtain the necessary network and CPU resources, attackers tend to use a large number of machines to launch Distributed DoS (DDoS) attacks. DDoS attacks can be devastating to the victims. While the DDoS problem is not new, its connection with web services technologies, and especially service oriented architectures, is not well understood. Most of the business applications on the Internet are dependent on web services for their transactions. Distributed denial of service (DDoS) attacks either degrade or completely disrupt web services by sending flood of packets and requests towards the victim web servers. An array of defense schemes are proposed but still defending web service from DDoS attacks is largely an unsolvable problem so far.

Web services are very useful technology today. Most of the web services are using SOAP protocol to deal with data as well as for the security. But the REST based services are better than SOAP because of its simplicity, interface flexibility, interoperability, and scalability. The purpose of this work is prevention on REST based services from DDoS attack. With the security concern SOAP is usually a better fit within the enterprise, where as REST is usually a better fit within public web facing service scenarios where it needs a high degree of scalability and interoperability. [6]

II. LITERATURE SURVEY

From the literature found that, today is an era of Internet. There are various methods for interaction between user and Internet which is web application through web services. Web services are based on the concept of service-oriented architecture (SOA). Web Services offered not only data exchange, but also tried to accomplish interoperability between different programming languages basing the entire data definition and data exchange on the well-known technologies, XML and HTTP. In the years to follow many security related specifications for Web Services appeared, extending the Web Services and enabling advanced security mechanisms, like cryptography, trust negotiation and single sign-on. But all those extensions did not just solve the challenges that the industry posed, they also made the Web Services one of the most complex distributed systems of the modern time. DDoS attack is one security problem with web services. Nowadays most of the business applications on the Internet are dependent on web services for their transactions. DDoS attack affects the web services by sending flooding packets and request towards the targeted web servers. There are various defence schemes for this problem but still not solved. In SOAP-based service, extra protocols specified in the various WS-* specifications, does support end-to-end message security which is very complicated. So it is implemented for the REST based web services because of its simplicity, interface flexibility, interoperability, and scalability.

III. PROPOSED SCHEME

In the proposed work, REST based services instead of SOAP based services. For REST based services this work will propose a method to prevent them from DDoS attacks. In this ASP .Net MVC Web API will be used to implement the scenario for the defence mechanism for DDoS attack on REST web services. In this framework various parameters are used to check for attack and try to analyse the configured values for those parameters. I will also try to dynamically alter the parameters according to the load. In this server initially start the operations based on the configured values. For that we can set the lower limit and upper limit for the parameters values. In this, if system reaches to lower limit or upper limit it will also try to alter the other parameters values according to that. It will try to restrict the number of simultaneous connection from the same IP and other resources gradually till lower or upper limit reached.

The client sends a request to the server using the HTTP request. The request is sending via the HTTP protocol. The request is firstly processed by the DDoS Protection Module which detects the DDoS attack from the request behaviour. And from that it will go through REST based web services if it is safe. Based on the client request the REST service is being invoked from the server. The invoked service at the server side sends a response for the client request. In this way the request response process is carried out. It is shown in figure 1.

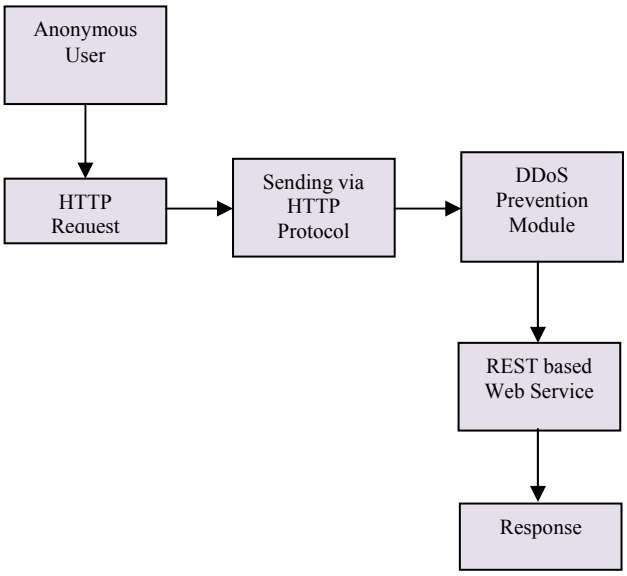

Fig 1 Process Flow Structure of Proposed System

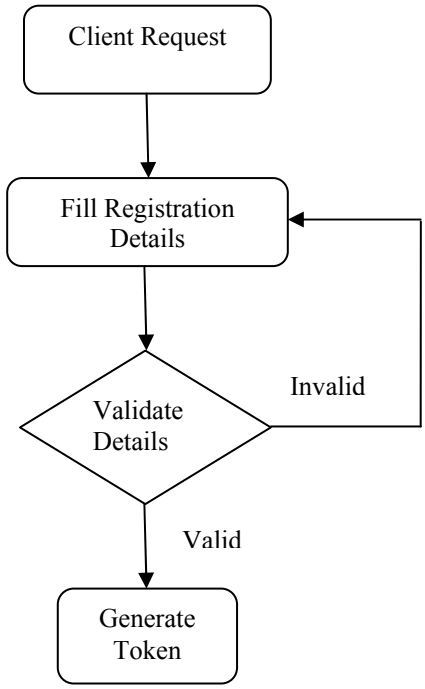

Fig 2 Flow Chart for Client Registration Module

The process flowchart for the client registration module is shown in figure 2. The client sends request and fill the registration details. The server checks if the details are valid or not. If the details are not valid then again go for the registration else it will generate a unique token for that client to access the web service. The token is stored in the database with all client details like login id, IP address, login time etc.

The flowchart for DDoS Prevention module is shown in figure 3. The client submits the request, and then the server tries to authenticate the user by its token. If the client does not have token then it goes to the registration module else it go for further validations. Then it checks the IP, if IP is black listed or blocked then it will drop the request. Else it will check the behaviour of that IP. If the behaviour is safe according to the check behaviour pseudo code then allows accessing the web service else it will block the IP.

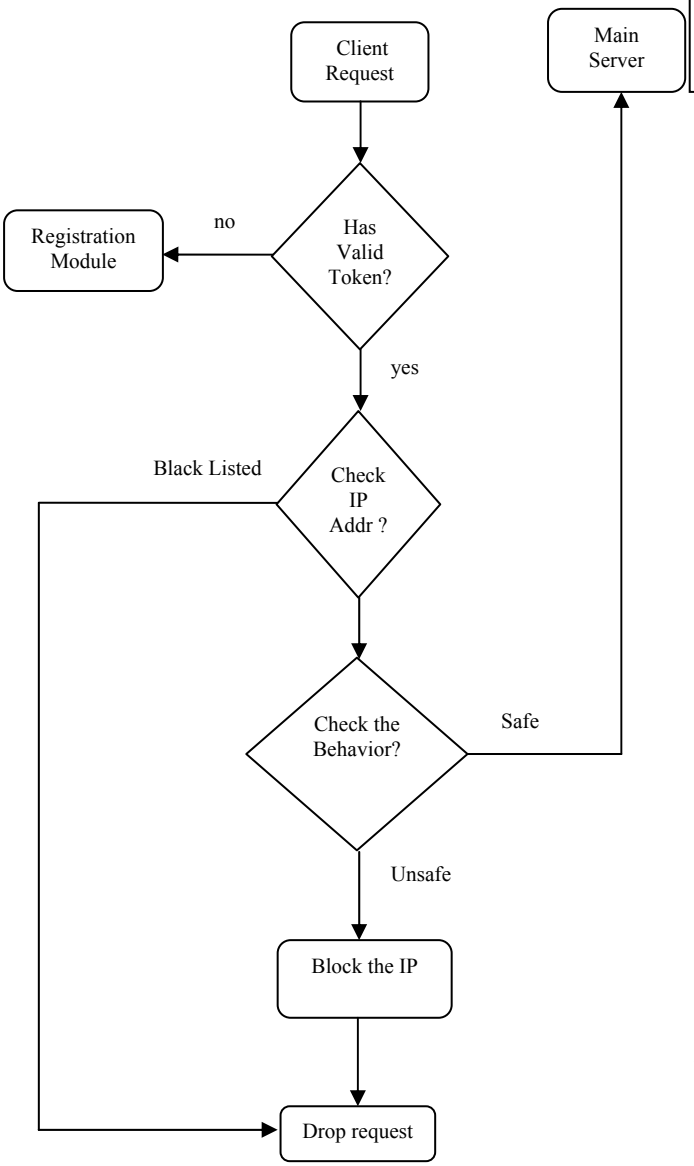

Fig 3 Flow Chart for DDoS Prevention Module

The Pseudo code for check behaviour is shown in figure 4 The client's request is received, then for that request current request time and previous request time for that IP is taken from the history. Then time interval between two requests is calculated. Then number of request per URI over the time period observed by IP is computed. Then the number of computed requests per URI is compared with the threshold defined. If it is greater than threshold then the IP will be blocked.

Pseudo Code for Check Behaviour

1. Get the request
2. Get Current Request Time (CT) and the Previous Request Time(PT)
3. Calculate the number of request per URI by IP
4. Calculate Time Interval ΔT=CT-PT
5. Compute the number of request per URI over the time period observed by IP
6. If number of requests per URI > Threshold
7. Then Block IP
8. Else End

Fig 4 Pseudo Code for Check Behaviour

Threshold

1. **Global Threshold**
   When Traffic is High, Set to be Constant
   $T_G < \sum_{t=1}^{T}(Req)t, x$
2. **Local Threshold**
   For Particular k URI from the same IP
   $T_L < \sum_{t=1}^{T}(Req)t, x\, k$,
3. **Average Threshold**
   For small Amount of Traffic, for limited URI
   $T_A < \sum_{t=1}^{T}(Req)t, x$ / n

Where,
n is the number of requested URI during the time period T
Req is the requests per second at a unit time from an IP x
$\sum_{t=1}^{T}(Req)t, x$ is Total No of Requests from the IP x during the Time period T, T is the Time period

Fig 5 Threshold Equations

The equations for the thresholds shown in figure 5 are considered based on the traffic captured. When traffic is high then it is considered as global threshold. This threshold is set to be constant. It is total no of Requests from the IP x during the Time period T. When the requests for particular k URI from the same IP is considered as local threshold. Which is total no of Requests from the IP x during the Time period T for particular k URI. When there is small amount of traffic for limited URI for that is considered as average threshold. Which is total no of Requests from the IP x during the Time period T divided by the number of requested URI during the time period T.

IV. RESULTS

The proposed work is implemented in Microsoft .Net and its technology ASP .Net MVC 4 Web Application and Microsoft SQL server.

Experimental results obtained in the form of CPU usage are compared here for the both existing and proposed approach for no of attacks for SOAP and REST based web services respectively. Also comparison of SOAP based web services and REST based web services are evaluated for the no of attacks.

Figure 6 shows the CPU usage for SOAP based web services with and without using existing algorithm for the 500, 1000, 1500 and 2000 number of requests. It shows that CPU Usage is reduced using the existing algorithm.

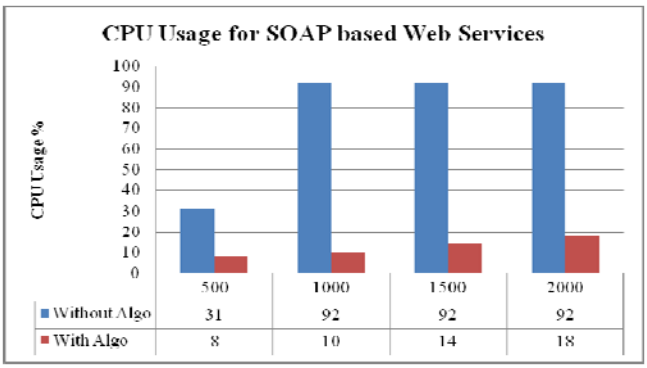

Fig 6 Comparison of CPU Usage for SOAP based Web Services with and without Existing Algorithm

Figure 7 shows the CPU usage for REST based web services with and without using proposed algorithm for the 500, 1000, 1500 and 2000 number of requests. It shows that CPU Usage is reduced using the proposed algorithm.

Figure 8 shows the CPU usage comparison of existing algorithm on SOAP based web services and proposed algorithm for REST based web services for the 500, 1000, 1500 and 2000 number of requests. It shows that CPU Usage is reduced using the proposed algorithm on REST based web services.

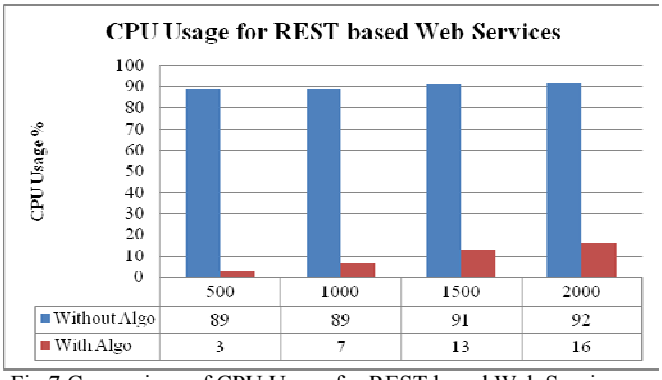

Fig 7 Comparison of CPU Usage for REST based Web Services with and without Proposed Algorithm

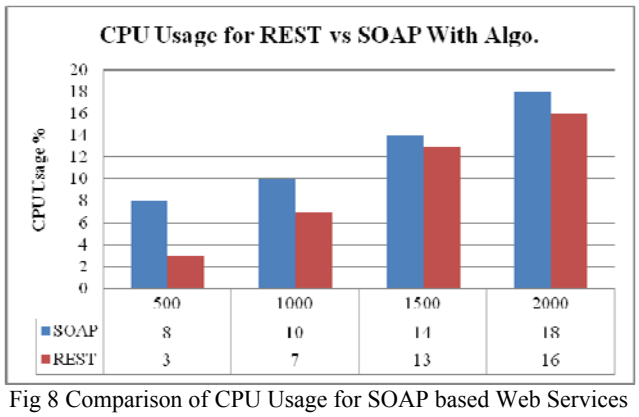

Fig 8 Comparison of CPU Usage for SOAP based Web Services using Existing Algorithm and REST Based Web Services using Proposed Algorithm

V. CONCLUSION

Results of proposed algorithm on REST based web service shows that there is a significant reduction in CPU usage. It has considered the results using total number of requests within the certain time interval and with and without proposed algorithm. The Comparison of CPU Usage for SOAP based Web Services using Existing Algorithm and REST Based Web Services using Proposed Algorithm indicates that REST based web service require less CPU usage than SOAP based web services. REST based web service can handle more number of requests than SOAP based web service.

REFERENCES

[1] "Countering the DDoS Attacks for a Secured Web Service", S.Igni Sabasti Prabu, Dr. V.Jawahar Senthil Kumar in Indian Journal of Computer Science and Engineering (IJCSE), 2013

[2] "A Survey of Attacks on Web Services - Classification and Countermeasures", Meiko Jensen, Nils Gruschka, Ralph Herkenhoner, in services," Computer Science - Research and Development (CSRD), Springer Berlin/Heidelberg, 2009.

[3] "A SOA Approach to Counter DDoS Attacks" Xinfeng Ye & Santokh Singh, IEEE International Conference on Web Services ,2007

[4] "Countering DDoS and XDoS Attacks against Web Services", Xinfeng Ye, IEEE International Conference on Embeded & Ubiquitous Computing, 2008

[5] "REST: An Alternative to RPC for Web Services Architecture," ,X. Feng, J. Shen, and Y. Fan, in Proceedings of the 1st International Conference on Future Information Networks, Beijing, China, October 2009, pp. 7–10.

[6] IBM, "Web Services Architecture Overview." [Online]. Available:http://www.ibm.com/developerworks/webservices/library/w-ovr/.

[7] "Service-Oriented Modeling: Service Analysis, Design, and Architecture". s.l. Jonhn Wiley & Sons, Bell, M. , 2008. 98-0-470-14111-3.

[8] "Validating Denial of Service Vulnerabilities in Web Services", S. Suriadi, A. Clark, and D. Schmidt, in Network and System Security, International Conference on Network and System Security. IEEE Computer Society, 2010, pp. 175–182.

[9] "Migration of SOAP-based Services to RESTful Services", B. Upadhyaya, Y. Zou, H. Xiao, J. Ng and A. Lau, in 13th IEEE International Symposium on Web Systems Evolution, pp. 105 – 114 (2011).

[10] "Exposing resources as Web services: A performance oriented approach", Kanagasundaram, R. Dept. of Syst. & Comput. Eng., Carleton Univ., Ottawa, ON, Canada Majumdar, S. ; Zaman, M. ; Srivastava, P. ; Goel, N, in IEEE Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2012 International Symposium on.

[11] "A Comparative Study of SOAP Vs REST Web Services Provisioning Techniques for Mobile Host", K.Wagh, R.Thool, Journal of Information Engineering and Applications, Vol. 2, No. 5, 2012, pp. 12-16

[13] "Threshold Based Kernel Level HTTP Filter (TBHF) for DDoS Mitigation", B Mohamed Ibrahim AK and Lijo George, in I. J. Computer Network and Information Security, 2012, 12, 31-39

[14] "Analysis of accounting models for the detection of duplicate requests in web services", B S. Venkatesan a, M.S. Saleem Basha, C. Chellappan, Anurika Vaish , Journal of King Saud University – Computer and Information Sciences (2013) 25, 7–24