Countering the DDoS Attacks for a Secured Web Service

S.Igni Sabasti Prabu
Research Scholar, Department of Computer Science and Engineering,
Sathyabama University,
Chennai, Tamil Nadu, India
igni.prabu@gmail.com

Dr. V.Jawahar Senthil Kumar
Associate Professor,
Anna University,
Chennai, India.

Abstract

Web services are the most widely used technology today. SOAP messages are the building blocks of web services. Since web services mostly deal with confidential data they must be protected from intruders. Intruders can be of any form. They can be a person or a program. One of the most adverse attacks is the distributed denial of service attack (DDoS). This kind of attacks is normally targeted at a particular service provider to exhaust the network and system resources of the provider. The proposed framework will restrict the connection attempts on a single user account so that it makes the job a little difficult for the hacker to hack information and will also monitor the size of the input soap message so that it does not consume too much of system resources.

***Keywords*:** DDoS; SOAP; WS-Security; XML.

1. Introduction

Web services are widely used in today's world due to its simplicity and modularity. Since it is simple it is vulnerable to all attacks. The distributed denial of service attack mostly targets the service providers. SOAP is a protocol specification for invoking methods on servers, services, components and objects, a way to create widely distributed complex computing environments that run over the Internet using existing Internet infrastructure. The soap messages can be easily attacked. Attacks on the soap protocol include parameter tampering, Recursive payloads attacks, oversized payload attacks, Simultaneous connections etc.

2. Proposed Architecture

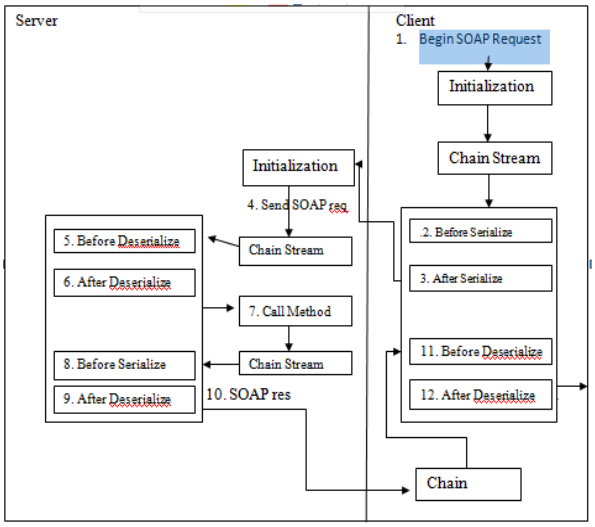

Fig.1Architecture of the proposed System

2.1 *Workflow of the proposed system*

The Client initiates a request to the server using a SOAP request. The request message is being deflated and then it is send to the server. Server side receives the deflated message from the client. The process

of converting deflated message to inflated message takes place in the server side. Based on the client's request, the requested service is being invoked from the server. The invoked service on the server side is being deflated and it has been sent as response for the client request in the form of SOAP response. The client receives the response from the server. Again in the client end, the process of inflating takes place so as to view the response. After the above said process, that accomplishes the task of request / response communication.

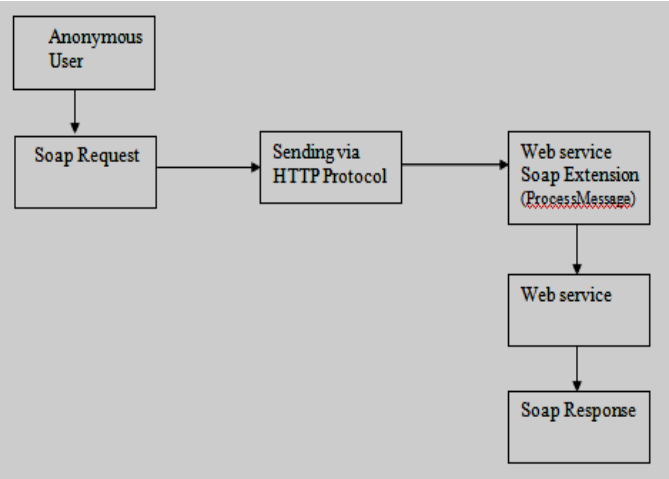

Fig.2 Process Flow Structure

The Web service soap extension checks for the various attacks. The various attacks countered in this paper are

- Unauthenticated Connections
- Over sized payload
- Simultaneous Connections
- Simultaneous Resources
- Connection Attempts

2.2 Threshold value selection

The attack detections are done based on certain values. These values are determined by the administrator and these threshold values are stored in the configuration file. A sample configuration file is shown below.

```
<appSettings>
  <add key="MaxSize" value="1024"/>
  <add key="SimCon" value="3"/>
  <add key="SimRes" value="10"/>
  <add key="ConnectionAttempts"  value="10" />
</appSettings>
```

- MaxSize – This parameter specifies the maximum possible size of the request.
- SimCon – This parameter denotes the most number of simultaneous connections possible for a single user at a given time.
- SimRes – this denotes the count of the the total number of resources that could be provided to a single user.
- ConnectionAttempts – This denotes the number of attempts the user can make for connecting to the server in the given time limit.

2.3 Oversize payload detection

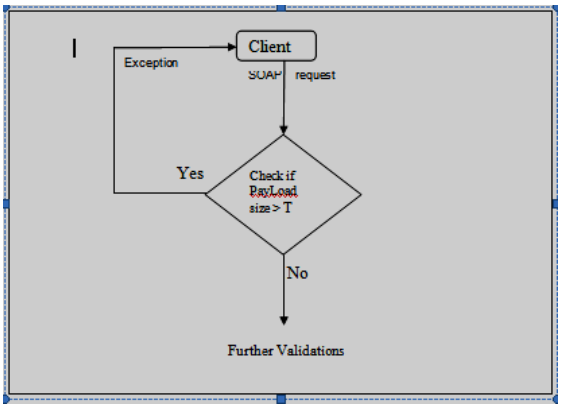

Fig.3. Detecting oversize payload

Client submits the username and password. The client data is being stored in the form of a table which contains Username, Password, Login Time, client IP address and status are stored in the server machine . If a size of SOAP Message exceeds the threshold limit during the given time limit an exception is being raised stating that permitted message size is exceeded. The threshold value is chosen from the configuration file. On the other hand if not, proceed to do further validations.

2.4 Simultaneous connections

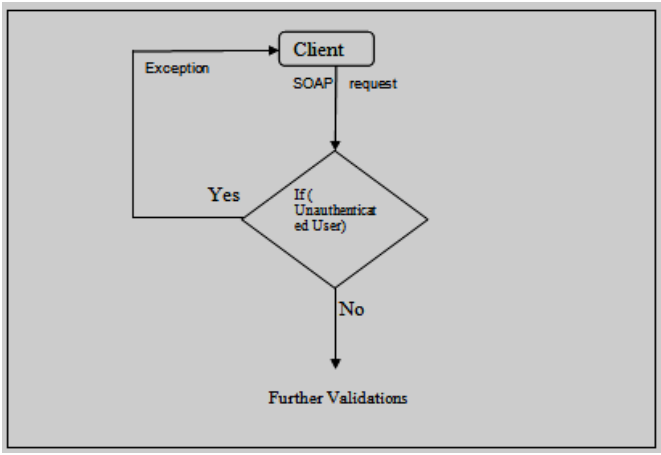

Fig.4. Flowchart for simultaneous connections.

Client submits the username and password. The server checks if the user is a valid user or not. If a client request is not from an Authenticated User, the client does not process the request. On the other hand if not, the system proceeds to do further validations. The threshold value is chosen from the configuration file.

2.5 Simultaneous resources

Client submits the username and password. The server checks how many connections from the same client are active at the given moment. If the concurrent active connections are less than the threshold value set the system proceeds to do further validations .The threshold value is chosen from the configuration file. If a client request exceeds the threshold limit during the given time limit an exception is being raised stating that Simultaneous requests limit exceeded.

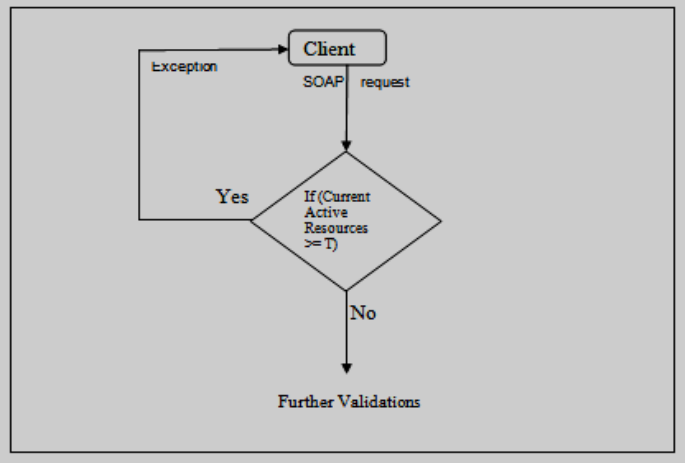

Fig.5. Flowchart for simultaneous resources.

2.6 Unauthenticated connections

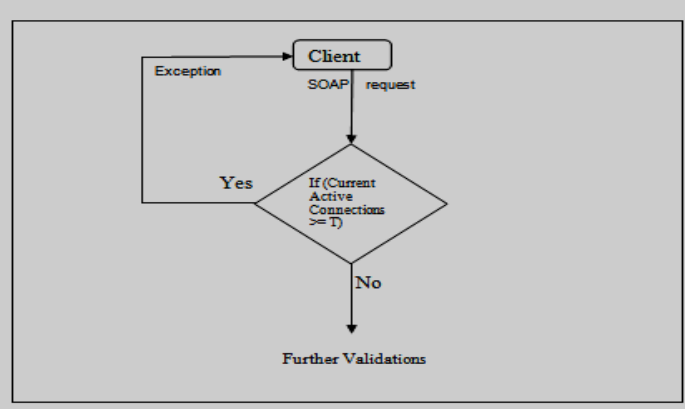

Fig.6. Flowchart for unauthenticated connections.

Client submits the username and password. The server checks how many resources are utilized by the same client that is active at the given moment. If the concurrent active connection's resource utilization count is less than the threshold value set the system proceeds to do further validations. If a client request exceeds the threshold limit during the given time limit an exception is being raised stating that Simultaneous Resources limit exceeded.

2.7 Connection attempts

Client submits the SOAP request. The server tries to authenticate the user and increases the connection attempt by one both in case of success and failure. If the connection attempt count is less than the threshold value set the system proceeds to do further validations. The threshold value is chosen from the configuration file. If a client request exceeds the threshold limit during the given time limit an exception is being raised stating that Connection Attempt limit exceeded.

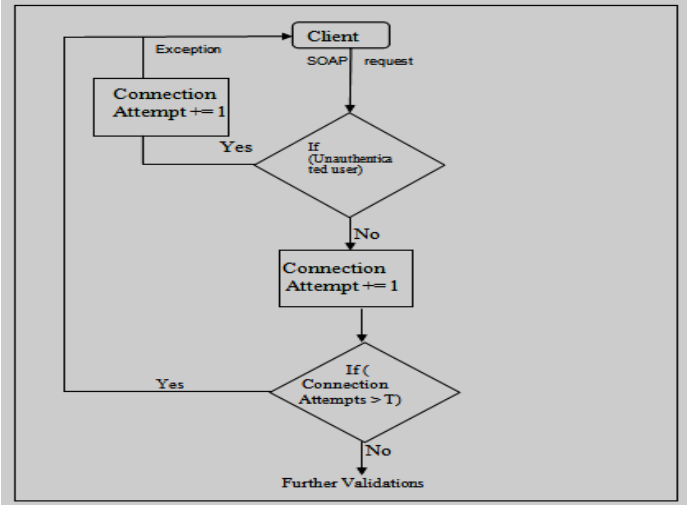

Fig.7. Flowchart for connection attempts.

2.8 Parameter optimization

The configuration parameters are dynamically altered based on the load. The server initially starts the operations with the best possible configuration values. It then starts to tweak the parameters in such a fashion that the load of the system is in a continuously uniform manner. The load of a single payload is initially tweaked without altering the remaining parameters. Then when the lower limit is reached it will start to alter the other parameters. It will restrict the number of simultaneous connections and other resources gradually till a lower limit is reached.

Load = (MAX / (Act_User / Param))
If (Load < MIN)
Load = MIN

Load = Current payload size to be permitted for the system.
MAX = Maximum possible payload.
MIN = Minimum possible payload.
Act_User = currently active users + 1.
Param = Max users permitted when using maximum payload.

3. Experimental Results

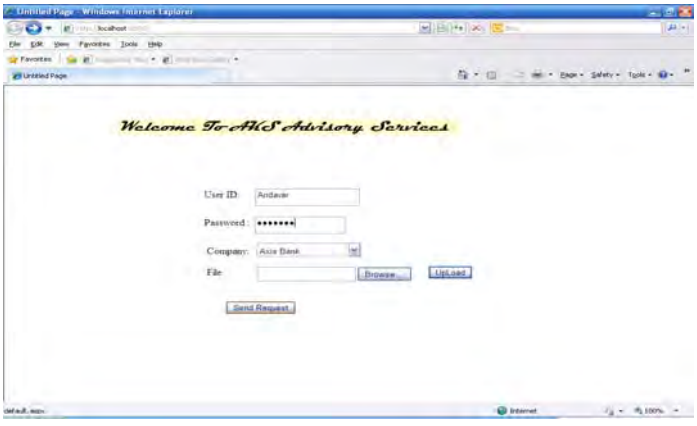

Fig.8. Login screen, the user gives the login details and also the XML file.

The Client provides the login detail along with

- Company Name for which he requires Details
- XML file to be uploaded with User Data

The server on receiving the SOAP message de-serializes the request and compares it with the threshold value specified in the configuration file. In the configuration file we specify the maximum number of simultaneous connections, the maximum size of the message, the maximum number of system resources and the number of connection attempts. The value in the configuration file can be changed or modified only by the administrator.

If the details provided by the user are valid and he hasn't overused the system in recent times then he will get the response from the server. The server restricts the enormous access of a single machine at a given time. This validation helps in preventing the server machine from flooded by zombie machine. It ensures that there can be only 'n' active connections at a given time. The n value is specified in the configuration file. When the limit is exceeding the server will respond with an "Error Message". Next it checks the message size, if the size is within the limit it will continue further validations or it responds back with an error message. If the request is within the threshold limit and the request is a valid request the server responds back with the response. The response is serialized and sent by the server and the client de serializes to get the response.

4. Conclusion

In this paper we have proposed different methods of countering the various DDoS attacks. The DDoS attacks is one of the severe threats to the web services which have adverse effects that may even cease the functioning of web server. Thereby denying the legitimate users their access to the web server.

5. References

[1] Bachor Alrouh & Gheorghita Ghinea A Performance Evaluation of Security Mechanisms for Web Services
[2] Christian Nagel, Bill Evjen, Jay Glynn, Morgan Skinner, Karli Watson. Professional C# (Wiley India Pvt. Ltd,2008)
[3] Ghossoon M. Waleed & R. Badlishah Ahmed, Security Protection using SOAP Messages Techniques,IEEE International Conference on Electronic Design,2008
[4] Kumar Sinha & Subrata Sinha, Limitations of Web Service Security on SOAP Messages in a Document Production Workflow Environment, IEEE,2008
[5] Nils Gruschka & Luigi Lo Iacono , SOAP Message Security Validation Revisited, IEEE International Conference on Web Services,2009
[6] Xinfeng Ye,Countering DDoS & XDoS Attacks against Web Services, IEEE International Conference on Embeded & Ubiquitous Computing,2008
[7] Xinfeng Ye & Santokh Singh,A SOA Approach to Counter DDoS Attacks,IEEE International Conference on Web Services ,2007
[8] www.wrox.com
[9] http://www.iss.net/security_center/reference/vuln/XML_EntityRefDoS.htm
[10] http://xmpp.org/extensions/xep-0205.html
[11] http://capec.mitre.org/data/definitions/82.html
[12] http://msdn.microsoft.com/en- us/library/7w06t139(v=vs.80).aspx
[13] http://msdn.microsoft.com/library/esw638yk.aspx