Journal of Computer Science, 9 (3): 308-313, 2013
ISSN 1549-3636

doi:10.3844/jcssp.2013.308.313 Published Online 9 (3) 2013 (http://www.thescipub.com/jcs.toc)

A Framework for Simple Object Access Protocol Messages to Detect Expansion Attacks for Secure Webservice

[1]Igni Sabasti Prabu Siluvai and [2]Veera Jawahar Senthil Kumar

[1]Department of Computer Science and Engineering, Sathyabama University, Chennai, India
[2]Department of ECE, Anna University, Chennai, India

Received 2012-12-22, Revised 2013-03-29; Accepted 2013-04-18

ABSTRACT

The world has shrunk in this internet era. The applications in the internet use XML and Web Services which are simple, but powerful standards that enable applications to more efficiently communicate with each other. Unfortunately this advantage is coupled with concerns of Web services security. All the services provided by the internet face security problem. The hackers find a loophole to attack the web service to eliminate the availability of service. One of the most severe threats is Denial of Service attacks which are intended to annihilate the availability of a service. In this study we propose a schema to detect a special type of Denial of Service attack where the hacker modifies the SOAP messages by expanding it. The message expanded thus, takes a huge amount of memory while parsing and thereby denies service to a legitimate request. To overcome this problem, in this study, we propose a new security scheme which adds a digital signature to the message and also limits the upper bound of the length of the SOAP message.

Keywords: DoS, SOAP, Web Services, WS-Security, XML

1. INTRODUCTION

As Web Services have become more popular, in every nook and corner of the enterprise communications, security is becoming crucial for operating Web Services. While the basic Web Service specifications (Gruschka and Luttenberger, 2006) do not address any security topics, a large number of additional specifications (WS-Security Ye (2008) WS-Security Policy Gruschka and Iacono (2009), WS-Trust, WS-Secure Conversation) for Web Services security exists. However all these standards focus on the aspects of message integrity and confidentiality and user authentication and authorization. The counter measure taken to secure the web service is very limited. Though we have packet filters, application level gateways and intrusion detection systems to prevent intrusion of hackers we still lack to secure a Web Service server's availability in an adequate manner.

In this article we present a framework for protecting Web Services from Denial-of-Service (DoS) attacks, where the SOAP messages are expanded by the hacker. The expanded code does not do any harm to the system but ceases the server's functioning. To prevent this kind of attack we limit the number of characters in a SOAP message, to be still more protective about the message from being hacked by the hacker we use the digital signature method to sign the message and encrypt the message.

1.1. Attacks on Services

Denial-of-service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include:

- Attempts to "flood" a network, thereby preventing legitimate network traffic
- Attempts to disrupt connections between two machines, thereby preventing access to a service
- Attempts to prevent a particular individual from accessing a service
- Attempts to disrupt service to a specific system or person

Corresponding Author: Igni Sabasti Prabu Siluvai, Department of Computer Science and Engineering, Sathyabama University, Chennai, India

Not all service outages, even those that result from malicious activity, are necessarily denial-of-service attacks. Other types of attack may include a denial of service as a component, but the denial of service may be part of a larger attack. Illegitimate use of resources may also result in denial of service. Denial-of-service attacks come in a variety of forms and aim at a variety of services. There are three basic types of attack:

- Consumption of scarce, limited, or non-renewable resources
- Destruction or alteration of requests
- Physical destruction or alteration of network components

In this study we concentrate on the second type of attack i.e., destruction or alteration of requests. There are many such attacks namely parameter tampering, replay attack, xml entity expansion attack, SQL injection attack. In this study we are going to concentrate on entity expansion attack. The architecture of the proposed study is shown below.

1.2. SOAP Message Format

1.2.1. Simple Object Access Protocol

Is a protocol specification for exchanging structured information in the implementation of Web Services in networks. A SOAP message is encoded as an XML document, consisting of an <Envelope> element, which contains an optional <Header> element and a mandatory <Body> element. The <Fault> element, contained within the <Body>, is used for reporting errors.

It relies on Extensible Markup Language (XML) for its message format. To make the XML standardize we use XML entities. Entities are variables used to define shortcuts to standard text or special characters. It is just like a macro which gets expanded when the document is processed. The xml entities are of different kinds. A normal XML entity code will look like the following:

```
<? xml version= “1.0”?>
<! DOCTYPE author [
<! ELEMENT author (#PCDATA)>
<! ENTITY email “igni@gmail.com”>
<!--the following use of a general entity is legal if it
is used in the XML document-->
<! ENTITY ig “Igni Prabu &email ;”>
]>
<author>&ig ;< /author>
```

The above piece of code is how we define an entity. The DoS attack which is detected in this study is study is Entity expansion.

1.3. Architecture

In this article we present a SCHEME for protecting Web Services from Denial-of-Service (DoS) attacks, where the hacker modifies the SOAP messages by expanding it .The proposed framework is shown in **Fig. 1**. The frameworks explains the encryption of the SOAP message and also the schema for detecting the entity expansion attack is shown in **Fig. 2**.

Digital signature is used for Providing the data confidentiality and data origin authentication for end user/Client. We create the certificate in visual studio for sender as well as receiver. Once we create the certificate they are stored in the address book. By providing the name of the certificate we can take the certificate and sign the message. A sample screenshot of creating the certificate is shown in **Fig. 3 and 4**.

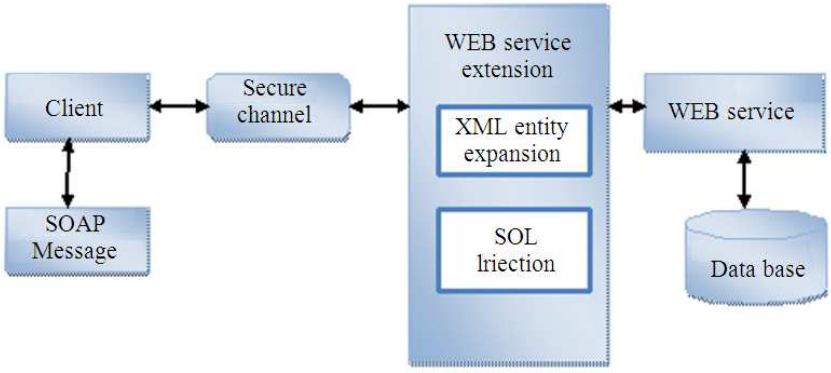

Fig. 1. Architecture of the proposed system SOAP Message

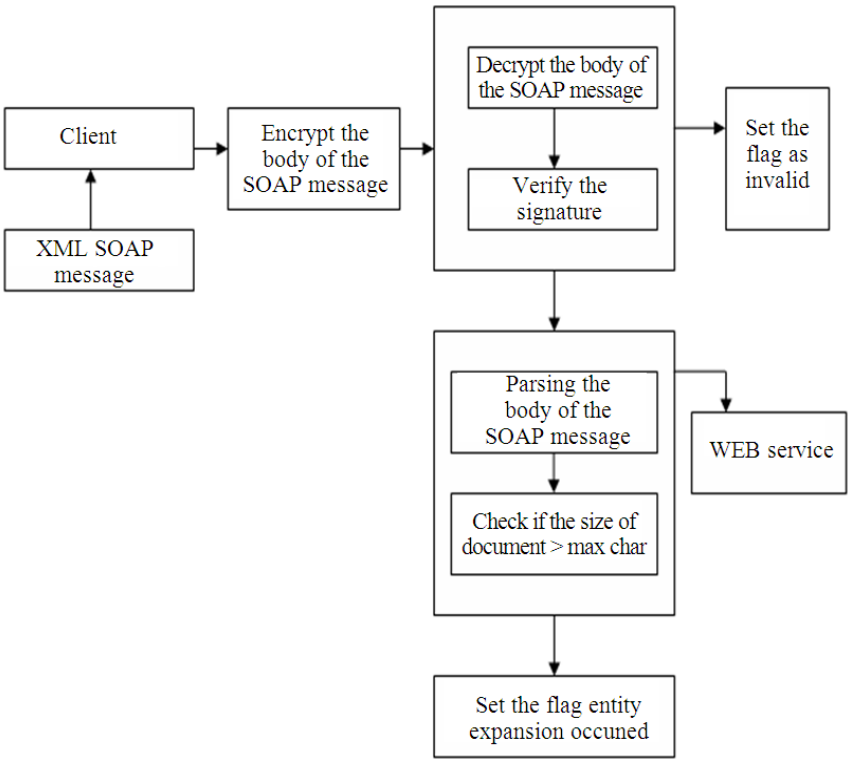

Fig. 2. Schema for detecting entity expansion attack

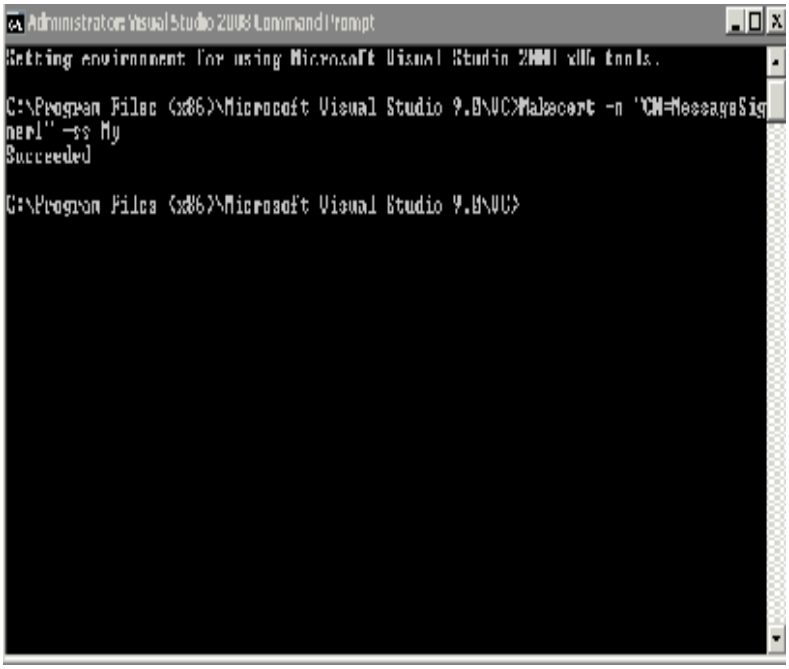

Fig. 3. Sample screen shot to create a digital certificate

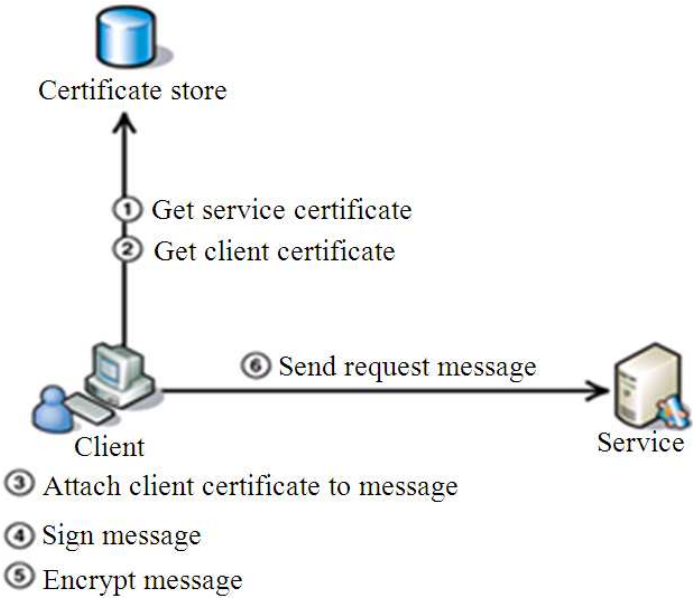

Fig. 4. Getting digital certificate by the client from a store

HTTP protocol is the stateless protocol, when the client and server communicate each other, SOAP message from client to server in a plain text format via stream. When the man in the middle reads the SOAP message we lose our data. For an example when we do online transaction, we may send our credit card number via SOAP message. In that case we are losing our very confidential information. To provide security between the client and server we implement digital signature. The Client's SOAP message will be hashed by using the Sender's Private Key. Then the hashed value will be encrypted using Receivers public key. So the return value is byte stream. Then the byte stream will be converted and is separated by comma. This comma separated value will be embedded into SOAP body then the message will be send. In the middle if any person want to decrypt the SOAP message it cannot be done. Because to decrypt the message we require Receiver's Private Key, this key is not shared between the client and server. Once the server receives this message it will be decrypted using Receiver's Private Key and then the message will be decoded into SOAP message by using the Clients public key. When the message is decrypted we verify the certificate. The figure below describes the process of the client getting the digital certificate from the store and attaching it to the encrypted message.

Now we get into the process of detecting expansion of SOAP message. This attack can be detected using the parsed data. A DTD is included in the input of the SOAP message. The DTD contains the entity reference. This entity reference will not affect the document until it is parsed from memory. In this case if the body contains the method call, it will invoke the web method. But if the body contains any entity expansion code, then while parsing the body the server will take long time to execute and also will take huge memory. To avoid this situation while parsing the body of the document we will be setting maximum chars in our document. If the document goes beyond the limit will be setting the status into Entity expansion attack. This attack isn't going to steal any data, but they could still cause a lot of damage through denial of service. Here the Client prepares the SOAP message by changing the SOAP body into XML Entity Expansion Code. When the server receives the SOAP message with attack code, it will take long time to execute and takes huge memory while executing, to avoid this maximum characters allowed is only 10000 while parsing the XML Document.

1.4. Experimental Results

Entity in XML is like a macro. An entity should be expanded to be used in the document. The entity expansion attack is a DoS attack where the entity when parsed to the memory will take a long time to be expanded and thereby denying the requests of the privileged users.

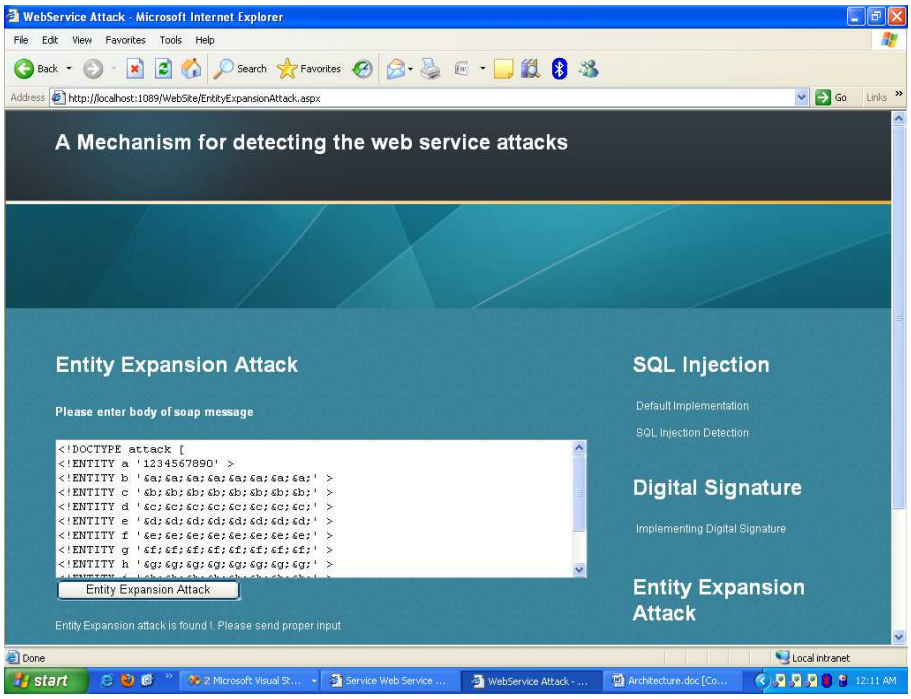

Fig. 5. Shows the error message detected by web service

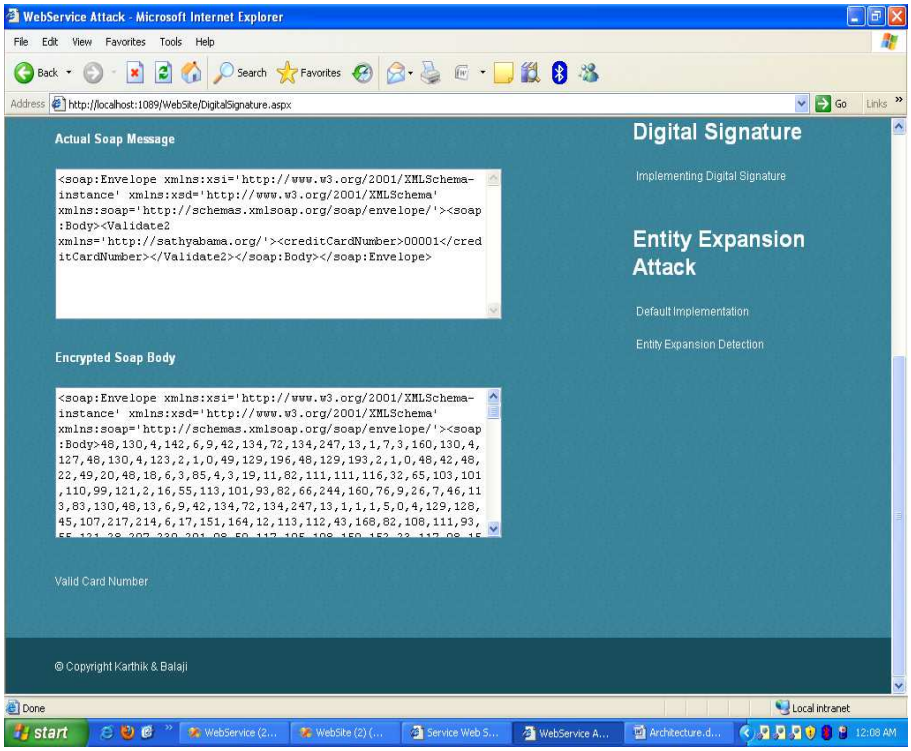

Fig. 6. Shows the encrypted SOAP message

The experiment which was carried out to prevent the expansion attack follows the below steps:

- The first step is to create the certificate for the sender as well as receiver
- The next step is to encrypt the actual SOAP message body
- We use RSA algorithm to encrypt the message
- This is done on the client side
- Moving on to the server side, first we decrypt the message
- Verify the signature, if it is valid we process the message further or send an error message
- Once the signature is verified and the message is decrypted we check if the entity has the limited number of characters
- The maximum limit of the characters set in our experiment is 10000
- If the message is going to exceed this number we display an error message
- If the message is a valid one the server will respond back to the client

This is the way in which the experiment was carried out and the **Fig. 5 and 6** are the result of the experiment carried out.

2. CONCLUSION

This study presents a new schema for detecting the DOS attacks. The request is also made secured by signing it using digital signature. In today's computerized world, web services have become in evitable. Security is the primary concern. The method proposed in this study detects and prevents one of the most vulnerable attacks, which makes the server to slow down its process, due to which the legitimate users are denied their usage.

3. REFERENCES

Gruschka, N. and L.L. Iacono, 2009. Vulnerable cloud: SOAP message security validation revisited. Proceedings of the International Conference on Web Services, Jul. 6-10, IEEE Xplore Press, Los Angeles, CA., pp: 625-631. DOI: 10.1109/ICWS.2009.70

Gruschka, N. and N. Luttenberger, 2006. Protecting web services from DoS Attacks by SOAP message validation. Proceedings of the IFIP TC-11 21st International Information Security Conference, May 22-24, Karlstad, Sweden, pp: 171-182. DOI: 10.1007/0-387-33406-8_15

Ye, X., 2008. Countering ddos and xdos attacks against web services. Proceedings of the IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Dec. 17-20, IEEE Xplore Press, Shanghai, pp: 346-352. DOI: 10.1109/EUC.2008.61