Cumhuriyet Üniversitesi Fen Fakültesi
Fen Bilimleri Dergisi (CFD), Cilt:36, No: 3 Ozel Sayı (2015)
ISSN: 1300-1949

Cumhuriyet University Faculty of Science
Science Journal (CSJ), Vol. 36, No: 3 Special Issue (2015)
ISSN: 1300-1949

Providing a Model to Measure the Performance of the Security Objectives of Service-Oriented Architecture

Sohrab ASADOLLAHZADEH MEHENH[1,*], Reza EBRAHIMZADEH MOGHADAM[2]

[1] *Young Researchers and Elite Club, Torbat-e Heydarieh Branch, Islamic Azad University, Torbat-e Heydarieh, Iran*

[2] *Departments of Computer, Zahedan Branch, Islamic Azad University, Zahedan, Iran*

Received: 01.02.2015; Accepted: 06.06.2015

Abstract. While web services have been widely accepted as an independent platform of service-oriented technology, their performance remains as a concern because of durability of XML. However service-oriented computing has become very common, a number of developed programs will be developed further using existing software components with standard interfaces. Performance implications of these applications to support an effective business process are very important. The performance evaluation in this distributed environment is a difficult problem. This paper investigates a study of web services performance under the impact of security. Having an inherent competition in these two features will produce a Trade Off which makes managers to choose a better performance at the expense of reduced security or vice versa.

Keyword: Security, Performance, Service-Oriented Architecture, Web Services

1. INTRODUCTION

Services are accepted as an independent platform for service-oriented technology which make the interaction of heterogeneous systems possible by text-based transfer protocols. Cooperation and interaction between systems significantly reduces the cost of integration systems. A common approach is to consider Web services cooperation along with performance. It has been reported that the transfer protocol in Web service (SOAP) is significantly slower than Binary Based [4,5] and Domain Specific protocols [6]. This has caused concern in Web service performance among researchers and industry and there is a need to work more to improve the performance [7,8,9,10]. On the other hand SOAP can not be a safe protocol for message transfer and creates a high risk for the parties in message exchange, although added security technologies such as ssl or https solve a part of the problem by encrypting messages between two spots [11].

But these transit layer security technologies cannot provide end-to-end security between the client and Web service in multi-layer distributed systems. Furthermore, these point to point security technologies are based on a specific protocol or transit layer such as TCP / IP for SSL and HTTP for HTTPS, while SOAP is an independent messaging protocol for Web services.

OASES develop the definitions of WSS [12] to provide the secrecy and authentication of protection of the surface of the message between the two ends (client and Web service) through message integrity. WSS provides SOAP with extendable architecture through adding information that is dependent of security (security signs, signatures, etc.). In the process of SOAP, information is stored like Body, but they might be protected with a password / signature. This design enables the integration of WSS with SOAP. Nowadays most Web services products support standard WSS. While the security of Web services is being increased, there may be

* Corresponding author. *Email address: sohrab_mehneh@yahoo.com*

Special Issue: The Second National Conference on Applied Research in Science and Technology

concerns about the overhead for performance. Overheads may be due to: 1- extra CPU processing for elements related to WSS, 2: More long transfer time network by adding WSS additions [13,14,15]. In this article we intend to review and evaluate the impact of WSS on the performance. Based on this review and assessment we will provide a model through which we can calculate Web service performance by considering WSS additions. We will evaluate this developed model compared with other existing models.

2. PERFORMANCE MODEL FOR WSS

WSS is an added security implementation that is added to a web service through which the SOAP messages are encrypted / signed and are transmitted to the receiver and are decoded.

The performance of a Web service can be considered as a combination of WSS performance, plus the added cost of time through SOAP message transmission and added cost of time to process the security contents of the SOAP message [2]. Obviously, security negatively affects Web service performance. Now we will develop the presented model in [1] and [2] by examining the effects of security on parts affected in web service.

According to Tanenbaum et al [3] security in distributed systems can be divided into two parts. One part is related to the relationship between users and processes that are may be in different machines. The main strategy to ensure the security of the communication is secure channel.

The second part is related to the license, which ensures access to resources in the distributed system which is intended for. Safe channels and access control needs solutions for distribution of encryption keys, authentication, signatures, etc., which is called security management. [3]

The model which is presented in [1] is in the following:

(2-1) Latency = Tmsgproc + Tmsgtran + Tsync + Tapp

Tmsgproc: shows the total message processing costs, including Coding / Encoding, Enc. / Dec., Sign / Verifying, Security Checking and sorting data type.

Tmsgtran: Shows the total cost of transferring a certain amount of messages on the network.

Tsync: shows overhead associated with synchronization protocols.

Tapp: time spent on logic business in the application layer.

Full details of the above model have come in reference [1]. In Figure 1 we illustrate a called Web service with a security implementation.

Figure 1, shows how to call a Web service through security strategies. Four major processes on demanding machine, four processes on the answering machine, and four major processes on security management machine.

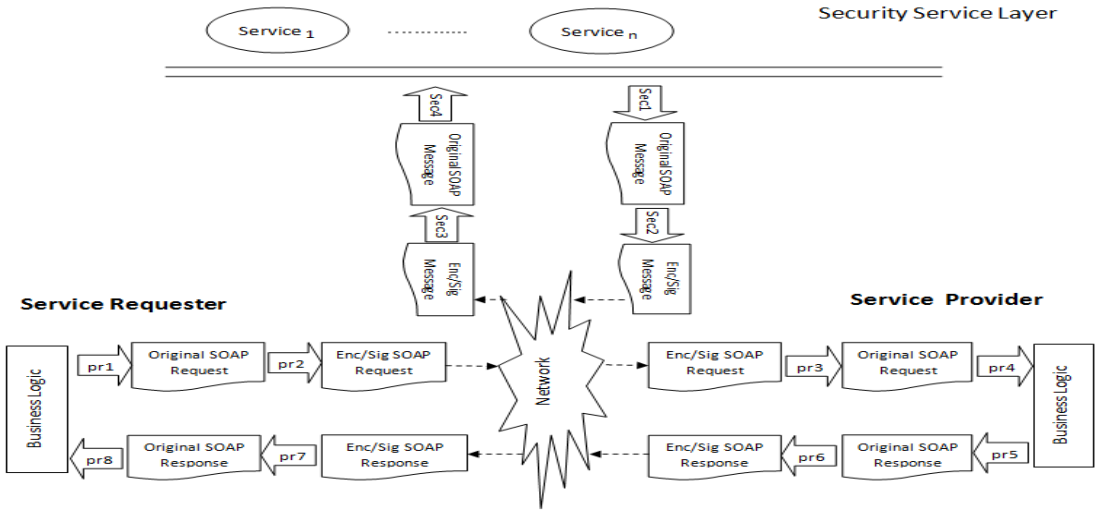

Figure 1: Analysis of a complete process of SOAP message with WSS

Pr1: computational procedures for coding data to generate a simple SOAP request.
Pr2: computational procedures to encrypt and sign a simple SOAP request.
Pr3: computational procedures to decode and trial SOAP request.
Pr4: computational procedures to decrypt SOAP request.
Pr5: computational procedures for coding data to generate a simple SOAP response
Pr6: Computational procedures for encryption and signing a SOAP response.
Pr7: computational procedures to decrypt the trial of a SOAP response.
Pr8: computational procedures to decrypt a SOAP response.
Sec1: computational procedures for data encryption to generate a SOAP message.
Sec2: computational procedures to encrypt and sign a SOAP message.
Sec3: computational procedures to decrypt a message and trial a SOAP message.
Sec4: computational procedures to decrypt the data to generate a simple SOAP message.

pr1, pr4, pr5 and pr8 are processes of a web service call without WSS. Pr2, pr3, pr6, pr7 and sec1 to sec4 are additional processes that are necessary to implement security in Web services.

We can model the performance of a Web Services under the effect of security implementations according to model (1) as follows:

W is the number of transits between the client and the server. For example, w = 1 represent one time send and w = 2 a request call / response.

Refpj shows CPU capacity of the platform reference.

pj represents cpu capacity of a machine which is based as a client / server.

αj represents the same substantial overhead of processing / analyzing a message for client / made server on a specific firmware running on a reference platform.
βj represents an overhead due to processing / analyze of a unit amount of message for j firmware often running on a reference platform.
n is the total number of devices involved in the network.
Mow is the actual size of the message conveyed by the media.
N is the network devices bandwidth.
τ is the delay of switching / routing messages in all network devices. D is the distance between client and server.

L is the speed of light in glass, for example, L = 2 * 105 km / s

W is the delay on WAN kernel

S is the number of occurred synchronization during messaging according to the following definition:

$$S=\delta(http)+\left\lfloor \frac{Mow}{Ws} \right\rfloor \qquad \delta(http)=\begin{cases} 1 & wait \\ 0 & no\_wait \end{cases} \tag{1}$$

m is the message size for each synchronization.
Ws is the size range of TCP from 16k to 64k.
αenc, αdec, αsig and αveri shows the same inherent overheads of encryption, decryption, signature and verification of a SOAP message.

βenc, βdec, βsig and βveri shows the same inherent overhead of encryption, decryption, signature, and verification of a value of a SOAP message.
X: The number of transits between the client and Z security service to establish a secure communication.
Y: number of transits between the service provider and Z Security Service to establish a secure communication.
Z: the number of security services involved in security objectives.
S1: number of synchronizations occurring between the applicant and the security service layer.
S2: number of synchronizations occurring between the service provider and security service layer.

$$Tmsgproc = Tmsgproc_{(service\ requester/\ service\ provider)} + Tmsgproc_{(security\ service\ layer)} =$$

$$\sum_{i=1}^{w}\left[\sum_{j=1}^{2}\left(\alpha_j + \alpha_{sec} + \left(\beta_j + \beta_{sec}\right)M_i\right)\frac{refpj}{pj}\right] + \sum_{i=1}^{Z}\left[\sum_{j=1}^{X_z}\left[\sum_{k=1}^{2}(\alpha_k + \alpha_{sec} + (\beta_k + \beta_{sec})M_j)\frac{refp_k}{p_k}\right]\right] +$$
$$\sum_{i=1}^{Z}\left[\sum_{j=1}^{Y_z}\left[\sum_{k=1}^{2}(\alpha_k + \alpha_{sec} + (\beta_k + \beta_{sec})M_j)\frac{refp_k}{p_k}\right]\right]$$

$$\alpha_{sec} = \alpha_{enc} + \alpha_{dec} + \alpha_{sig} + \alpha_{veri} \qquad \beta_{sec} = \beta_{enc} + \beta_{dec} + \beta_{sig} + \beta_{veri} \tag{2}$$

$$Tmsgtran = Tmsgtran_{(service\ requester/provider)} + Tmsgtran_{(security\ service\ layer)} =$$

$$\sum_{i=1}^{w}\left[\sum_{j=1}^{n}\left(\tau_j + \frac{M_o w_i}{N_j}\right) + \frac{D}{L} + W\right] + \sum_{i=1}^{Z}\left[\sum_{j=1}^{X_z}\left[\sum_{k=1}^{n}\left(\tau_k + \frac{M_o w_j}{N_k}\right) + \frac{D}{L} + W\right]\right] + \sum_{i=1}^{Z}\left[\sum_{j=1}^{Y_z}\left[\sum_{k=1}^{n}\left(\tau_k + \frac{M_o w_j}{N_k}\right) + \frac{D}{L} + W\right]\right] \tag{3}$$

$$Tsynch = Tsynch_{(service\ requester\ /\ provider)} + Tsynch_{(security\ service\ layer)} =$$

$$\sum_{i=1}^{s}\left[\sum_{j=1}^{n}\left(\tau_i + \frac{m_i}{N_j}\right) + \frac{D}{L} + W\right] + \sum_{i=1}^{Z}\left[\sum_{j=1}^{S_1}\left[\sum_{k=1}^{n}\left(\tau_j + \frac{m_j}{N_k}\right) + \frac{D}{L} + W\right]\right] + \sum_{i=1}^{Z}\left[\sum_{j=1}^{S_2}\left[\sum_{k=1}^{n}\left(\tau_j + \frac{m_j}{N_k}\right) + \frac{D}{L} + W\right]\right] \tag{4}$$

- Presentation of all items together: (presenting model in form of integration)

Through the above models for each called processing on the network, we can present a general model to measure the performance for secure distributed calls:

$$(5)\quad Latency = Latency_{(requester\ or\ provider\ /\ security\ service)} + Latency_{(requester/provider\ service)} = \left[\left[Tmsgproc_{(requester\ or\ provider\ /\ security\ service)} + Tmsgtran_{(requester\ or\ provider\ /\ security\ service)} + Tsynch_{(requester\ or\ provider\ /\ security\ service)}\right]\right] + \left[\left[Tmsgproc_{(requester\ service/provider\ service)} + Tmsgtran_{(requester\ service/provider\ service)} + Tsynch_{(requester\ service/provider\ service)}\right]\right] \quad (5)$$

In this section we modeled the delay of a SOAP message affected by security implementations in Web services, so we divided delay in two parts. The first part is related to the delay between the service provider or requester and security service (security management) and the second part is related to the delay between the requester and service provider which can be seen in Figure 1.

3. MODEL EVALUATION

We express our models capabilities in this model then we will compare our model with other existing models.

Table 1. Model capabilities.

Capability	Research factor	Research Scenario
considering communication security	Pr3,pr2 Pr7,pr6 Sec3,sec2	Pr2 encryption and signature, pr3 decryption and verification on the client. Pr6 encryption and signature, pr7 decoding and verification on the service provider. sec2 encryption and signature, sec3 decoding and verification in security services.
Considering security management	Security Service	A component called Security Service is responsible for security management tasks.
collectivity secrecy validity, authority	Pr3,pr2 Pr7,pr6, Security Service	Pr2, pr3, pr6 and pr7 are used to encrypt and sign decryption and verification, and service security for password management and distribution, sign distribution.....
considering security as a service	Security Service	Service Security as a security service is active.

In Table 1, we provided the capability of our model and described the factors that are affected by different security parts in the presented performance model. Obviously, if we want to propose performance model affected by security, regardless of the two important parts of security (communications security and security management) the model will not work correctly. In Table 2, we compare our model with other models in this area and we show the benefits of our model in this table

Table 2. Comparison of the model with other models available.

Name of the Model	Communications security Cost	security management costs	Security as a Service	Processing costs	Transfer fees	Synchronization costs
Model Presented in [2]				√	√	√
Model Presented in [3]	√			√	√	√
Model Presented	√	√	√	√	√	√

4. CONCLUSION

The obtained results are due to testing on the Kerberos system and placement in formula according to the following table that shows the amount of costs incurred by performance is strongly influenced by the security, and our measurement model that considers security in terms of its performance, shows performance delay more.

Table 3. Kerberos system test results.

The amount of the main message	The amount of security Message	Delay in Model [2]	Delay in Model [3]	Delay in presented model
2MB	512KB	87	121	232
3MB	512KB	127	177	288
8MB	512KB	327	457	568
2MB	1MB	87	121	316
3MB	1MB	127	177	372
8MB	1MB	327	457	652

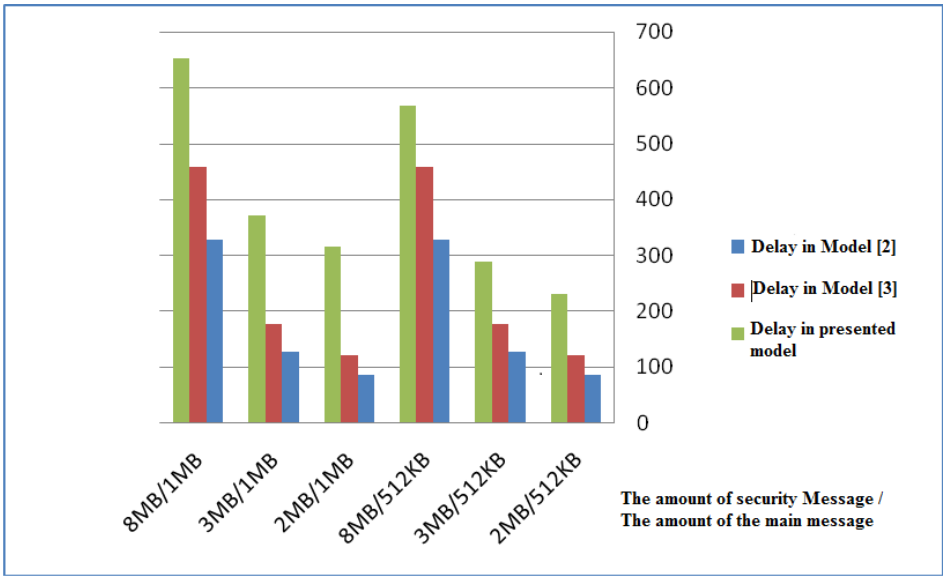

Figure 5-1. Kerberos system test results.

According to the results obtained in Table 3, it is clear that the model presented in [2] and [3] doesn't show correct results in the amount of delay due to security. Because as the size of the security message gets larger no growth in delay is observed. However the presented model fully shows the growth in delay because the original and secure message gets large. However, we note that this case has been investigated in Kerberos. A comprehensive evaluation model to assess the performance of web security services was presented in this article. We first investigated security in Web services, and then we provided a model to assess the performance affected by the security implementations. We examined security in our model in two parts, including security management and security implementation of communication. And we believe that when security is required in Web services the performance is debatable in the two parts.

In the presented model we examined the performance of the requester / service provider to security provider service and between requester to service provider, and it was clear that in order to measure the performance of a Web service we should pay special attention to the way in which we implement and manage security.

REFERENCES

[1] chen ,shaping , yan , bo , zic, john, liu, ren, ng, alex, 2006, Evaluation and modeling of web service performance,IEEE Icws'06

[2] chen,shaping, zic, john, tang kezhe, levy, david, 2007, performance Evaluation and modeling of web service security, IEEE Icws'2007

[3] Tanenbaum,Andrew s,van steen, maarten, 2006, distributed systems (principles and paredigms) 2nd Edition, Isbn:0-13-239227-5

[4] davis,d,parasher,m,2002, latency performance of SOAP Implementions,proc of IEEE cluster computing an the GRID (CCGRID'02) p407.

[5] hiu,k, et, 2002, Investigating the limits of SOAP performance for scientific computing, proc of 11th IEEE International symposium on high performance distributed computing (HPDC'02) PP.246-254

[6] C. Kohlhoff and R. Steele. Evaluating SOAP of High Performance Business Applications: Real-Trading Systems. Proc. of WWW2003, Available at www-taff.it.uts.edu.au/~rsteele/EvaluatingSOAP.pdf Time

[7] M. Cai, et al. A Comparison of Alternative Encoding Mechanisms for Web Services. Proc of DEXA2002 http://dblab.usc.edu/microsoft/ http://www.datapower.com/

[8] [9] P. Sandoz, et al. Fast Web Services (on-line) Accessed 27 August 2003,Available athttp://developer.java.sun.com/developer/technicalArticles/WebServices/fastWS/index.html

[9] K. Devaram and D. Andresen. SOAP optimization via parameterized client-side caching. Proc of the IASTED conf. on Parallel and Distributed Computing and Systems, pp. 785--790, 2003

[10] D. Booth, H. Haas, F. McCabe and etc., Web Services Architecture, W3C Working Group Note 11 February 2004, http://www.w3.org/TR/ws-arch/

[11] Web Services Security: SOAP Message Security 1.0 (WSSecurity 2004), OASIS Standard 200401 March 2004, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

[12] Web Services Security: X.509 Certificate Token Profile,OASIS Standard 200401, March 2004, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0. pdf XML Encryption Syntax and Processing http://www.w3.org/TR/xmlenc-core/

[13] C. Kohlhoff and R. Steele. Evaluating SOAP of High Performance Business Applications: Real-Time Trading Systems. Proc. of WWW2003

[14] H. Liu, X. Lin and M. Li. Modeling Response Time of SOAP over HTTP. Proc. of IEEE International Conference on Web Services (ICWS'05), pp. 673-679