Web Services Security Assessment: An Authentication-Focused Approach

Yannis Soupionis, Miltiadis Kandias

Web Services Security Assessment: An Authentication-focused Approach

Yannis Soupionis, Miltiadis Kandias

Information Security and Critical Infrastructure Protection Research Laboratory
Dept. of Informatics, Athens University of Economics & Business (AUEB)
76 Patission Ave., Athens, GR-10434 Greece
{jsoup, kandiasm}@aueb.gr

Abstract. Web services may be able to publish easily their functions to the rest of the web world. At the same time they suffer by several security pitfalls. Currently, there is limited research on how the proposed web-services security countermeasures affect performance and applicability. In this paper, we introduce the threats/attacks vs. web-services authentication, present the most widely used security method for protecting it, and identify the threats/attacks tackled by those methods. Moreover, we evaluate the web service authentication mechanism proposed in these implementations, not only on a theoretical level (by taking into consideration all the security issues of the implementing authentication sub-mechanisms), but also in a laboratory environment (by conducting extensive experiments). Finally we demonstrate the trade-offs between sophisticated web-service security methods and their performance.

Keywords: Security Assessment, Web Services, Authentication, Performance.

1 Introduction and Related Work

The explosive growth of the World Wide Web has introduced a wide array of new technological advances and several sophisticated end-user services. Development in data networks facilitated the introduction of web services [1] [2] [3], which have been increasingly penetrating the web market. Web services are a realization of a more abstract architectural style, called Service-Oriented Architecture. While web services introduce a variety of new functionalities, they also increase the complexity of the system and can be subject to a significant variety of attacks. Currently, there is limited evaluation on how web services security are practically enforced. Existing work focuses on high level aspects of security, namely business-oriented ones, while it has paid limited attention to technical aspects. A relevant publication shows an extension of the WS-Security standard [6]. Another publication [7] defines three security policy assertions provided in SOAP Message Security, WS-Trust and WS-Secure Conversation. The paper defines a basic set of assertions, which describes how messages are to

be secured, without providing enough information on development or performance trade-offs. Both the above papers propose mechanisms to enforce specific security aspects, but they provide no metric, capable of assessing a balance between performance and security.

In our paper, we provide a model capable of measuring the effectiveness and performance of security mechanisms and put in place with web services. In particular, we focus on the authentication process. A web service cannot function properly if the authentication mechanism is either compromised, or complex enough to inadequately downgrade the performance of the web service. Our approach is based on (a) identifying all possible attacks/threats of web services authentication, (b) assessing the key methods of securing web authentication, and (c) evaluating the system performance.

In Section 2, we present authentication in web services and we illustrate the web services authentication threats. We also show whether they are tackled by the authentication methods. In Section 3 we provide the results of a series of experiments on how different implementations of authentication mechanisms affect the system performance. Finally, we describe our conclusions, together with plans for future work.

2 Web Services Threats and Countermeasures

In addition to software development best practices that are proposed mainly by the various WS standards, numerous countermeasures exist. An authentication process consists of: a) the authentication data of the requesting entity, b) the channel through which this data is transmitted, and c) the data validation according to an authentication database.

The security countermeasures, which were chosen to protect the service, are: a) a username and a password (credentials), b) encryption over the channel, and c) password hashing [7]. Each of the above countermeasures may be implemented by different technologies, which are:

1. Password: a) One-Time-Password, b) good practice based password, c) based on no good practice (i.e., chosen once and valid forever, no rules about using special characters, etc.).
2. Channel: a) VPN, b) Asymmetric SSL, c) Symmetric SSL, d) unprotected channel (HTTP).
3. Password storage: a) MD5, b) SHA1, c) Plaintext.

The vulnerabilities introduced by the software implementing these methods, which can introduce new attacks to the service, are out-of- scope of this paper.

In order to study the effectiveness of the above security countermeasures, we produced a list of known web-service attacks. The list is based on WS standards [9-12] and research output. From the produced list of 24 attacks we excluded 7, because they do not affect authentication mechanism. In Table 1, if a countermeasure/ mechanism deals with an attack, then the corresponding cell is checked. This table is a useful tool for those in charge of either the security or the development of a web service. Fur-

thermore, based on the results of a risk assessment review, the analyst can choose a combination or a single countermeasure to tackle attacks assessed as important.

Table 1. Countermeasures vs. Threats

Mechanism / Threats	Password			Channel				Password Storage		
	OTP	Good Practice Based	No Good Practice	VPN	Symmetric SSL	Asymmet-ric SSL	HTTP	SHA1	MD5	Plaintext
Replay attack	✔			✔	✔	✔				
Session hijack				✔	✔	✔				
Security tokens altera-tion/stealing	✔			✔	✔	✔				
System misconfigura-tion attacks										
Man-in-the-middle				✔	✔	✔				
Eavesdropping	✔			✔	✔	✔				
Password guessing	✔	✔		✔	✔	✔				
Cryptanalytic pass-word attacks				✔	✔	✔		✔	✔	
Brute force attacks	✔	✔						✔	✔	
Denial of Service							✔			✔
Spoofing attacks	✔			✔	✔	✔				
Information gathering				✔	✔	✔				
Unauthorized access	✔	✔	✔	✔	✔	✔		✔	✔	
Insecure audit/ log - repudiation attacks	✔	✔	✔	✔	✔	✔				
Retrieval of clear text configuration				✔	✔	✔		✔	✔	
Encryption keys' theft				✔		✔				
SQL injection										

3 Experimentation results

In this section we identify the performance issues raised when a secure mechanism is used for a web-service authentication. The performance issue depends on the resources consumed, as well as on the time needed for establishing a web-service session. In order to determine and evaluate how the authentication method affects the system performance, we had to record the corresponding values of the monitoring subjects (time, computational resources) during significant overload. The tests were performed so as to evaluate a web service authentication mechanism, strengthened by the countermeasures mentioned in Section 2. The password policy was not tested through experimentation, as it is based solely on human factor parameters and do not differentiate, in terms of computational cost. The purpose of the first two tests is to evaluate the performance of each countermeasure, separately. In the final test, we implemented combinations of countermeasures in order to get a more holistic view of the authentication web service's performance.

Mechanism scenarios and results. In order to implement the scenarios, we used ten external clients, which initiated web-service requests. The scenarios consisted of five

phases. During the first phase, each external client made only one request. Thus, the web-service received a total of 10 requests. During the second phase, every external client made 100 requests. Thus, the proxy received a total of 100 requests, while in the last one, with proportional way, the web-service received 105 requests.

Table 2. Channel type experimental performance

Requests		10	100	1.000	10.000	100.000
Average time per session (msec)	**VPN**	29,3	28,9	29,1	171,1	298,0
	Asymmetric SSL	27,4	27,6	27,4	188,2	282,3
	Symmetric SSL	20,4	21,9	20,7	112,1	204,4
	HTTP	12,0	12,0	12,0	28,0	125,0

Only one mechanism was evaluated for each scenario. The other mechanism was implemented so as to produce the minimum overload to the test. If the channel type was the one tested, then the password storage parameter was set permanently to plaintext. The results of the evaluation of the channel type are summarized in Table 2.

The average processing time, for a request to be properly handled, was <300 msec. The maximum time for handling a request was 2.5 sec. This appears to be excessively high. It happened when many requests were to be handled and thus a very high CPU and memory overload occurred. Therefore, if a web service expects many simultaneous requests, then a VPN implementation is not a good solution. The results produced by the evaluation of the password storage policy are summarized in Table 3.

Table 3. Password storage experimental performance

Requests		10	100	1.000	10.000	100.000
Average time per session (msec)	**MD5**	9,5	9,2	12,6	75,4	102,8
	SHA1	9,8	10,0	16,8	107,8	143,2
	Plaintext	7,1	6,9	6,7	8,1	12,4

The average processing time, for a request to be properly handled, is <150 msec. The maximum time for retrieving a password was 1.9 sec for the SHA1 and 1.3 sec for MD5. These time periods may not be always acceptable, as several applications that cooperate with the web-services have lower timeouts. The long time is needed in case there were many requests to be handled. In this case, there was a high I/O to the hard drives, while both CPU and memory were overloaded.

Extended test scenarios and results. In order to implement these scenarios, we used ten external clients, which initiated web-service requests. The extended test scenario consisted of three phases (1000, 10000, 100000 requests). During these tests, all authentication sub-mechanisms were implemented, in all combinations. In Figure 1, we illustrate the most important fraction of these implementations, as well as the average time needed to serve an incoming request. Our main conclusions are: a) the symmetric SSL implementation is from 15-20% quicker than asymmetric, b) the performance

is slightly affected by the use of SHA1 or MD5 (5%), and c) the non-encrypted channel improves significantly the web service performance.

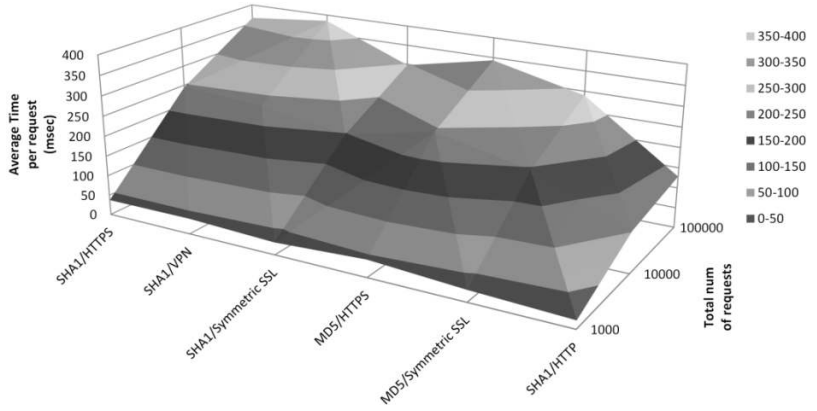

Fig. 1. The average time needed for serving a web-service request

Taking into consideration the above mentioned conclusions, as well as the maximum time period to serve a request, which is ~3 sec (except of SHA1/HTTP), it turns out that the web-service providers should use these results when selecting which countermeasure to use for strengthening a web-service authentication mechanism.

Concluding, the main questions a web-service provider should answer, before hardening the a web service security, are: a) how many requests are going to be received?, b) is it possible for the web-service and the participating machines to handle the key management for a symmetric SSL implementation?, c) should all the transmitted data between the corresponding entities, apart from authentication process, be encrypted in order to justify the VPN overload?, and d) is the processed web-service data useful enough to worth the additional overload of the encrypted channel?

4 Conclusions and further research

In this paper, we described a security assessment capable of evaluating web service authentication mechanisms. Based on security mechanisms, such as communication channel type, password storage, and password alteration policy, we managed to identify how these mechanisms deal with the corresponding attacks. Then, we evaluated the mechanisms by implementing a web-service. Our experimentation aimed at determining the performance in various combinations of the authentication implementations. We showed that security countermeasures/mechanisms selection should be based not only on their ability to tackle attacks, but also under the prism of their computational cost. Thus, web services, which are expected to serve requests under heavy load, must be developed with this parameter in mind. The correlation between security and computational cost can be used to explain to security unaware stakeholders the requirements put and the decisions made. Our assessment proved to be a useful tool

not only for those implementing web-services, but also for those who make management decisions. Further research could extend the proposed security assessment with additional goals, such as data integrity, non-repudiation, or authorization. As the additional security-related goals may add complexity, we may adopt different methods for security assessment, e.g., entropy algorithms [8], or take into consideration information for detected authentication vulnerabilities through formal analysis [14]. Finally, we intend to study the aforementioned characteristics in composed web-service environments (mashups) [5],[13].

Acknowledgments

This work was performed in the framework of the SPHINX (09SYN-72-419) Project, which is partly funded by the Hellenic General Secretariat for Research and Technology (http://sphinx.vtrip.net).

References

1. Papazoglou M., "Web Services and Business Transactions", *World Wide Web*, Vol. 6, No. 1, pp. 49-91, 2003.
2. Erl T., *Service-Oriented Architecture:Concepts,Technology and Design*,Prentice Hall, 2005
3. Erl T., *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, Prentice Hall, 2004.
4. Alonso G., Casati F., Kuno H., Machiraju V., *Web Services: Concepts, Architecture and Applications*, Springer-Verlag, 2004.
5. Zeng L., et al.., "QoS-Aware Middleware for Web Services Composition", *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, pp. 311-327, 2004.
6. Vedamuthu A. S., et al., "Web Services Policy 1.5 - Framework", W3C Recommendation, 2007. Available at: http://www.w3.org/TR/ws-policy/ (last visit Jan. 5, 2012).
7. OASIS Standard, "WS-SecurityPolicy 1.3", 2009. Available at: http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.3/os/ws-securitypolicy-1.3-spec-os.html (last visit Jan 5, 2012).
8. Serjantov A., Danezis G., "Towards an Information Theoretic Metric for Anonymity", in *Privacy Enhancing Technologies (PETS)*, pp. 41-53, 2002.
9. Nadalin, A., et al., "OASIS WS-Trust 1.4", OASIS, 2008.
10. Nadalin A., Kaler C., Monzillo R., Hallam-Baker P., "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)", OASIS Standard, 2006.
11. Vedamuthu A., Orchard D., Hirsch F., Hondo M., Yendluri P., Boubez T., Yalcinalp U., "Web Services Policy 1.5 - Attachment", W3C Recommendation, 2007.
12. Nadalin A., et al., "WS-SecureConversation 1.3", OASIS Standard, 2007.
13. Rosenberg F., Khalaf R., Duftler M., Curbera F., Austel P., "End-to-end security for enterprise mashups", in *Proc. of the 9th International Conference on Service Oriented Computing* (ICSOC '09), pp. 389-403, Stockholm, Sweden, November 2009.
14. Basagiannis, S., Katsaros, P., Pombortsis, A., "Intrusion Attack Tactics for the model checking of e-commerce security guarantees", In *Proc. of the 26th Int. Conference on Computer Safety, Reliability and Security* (SAFECOMP), Nuremberg, Germany, pp. 238-251, Springer Verlag, 2007.