Securing Web Services: Existing Solutions and Their Limitations

Aziz Nasridinov*, Jeong-Yong Byun**, Young-Ho Park*
*Dept. of Multimedia Science, Sookmyung Women's University
**Dept. of Computer Engineering, Dongguk University
aziz_nasridinov@yahoo.com, byunjy@dongguk.ac.kr, yhpark@sookmyung.ac.kr

Abstract

Web Services have been used in a variety of applications and have become a key technology in developing business operations on the Web. However, despite the growing use of Web Services by large corporations, security is still a major concern that is slowing their deployment. In this paper, we give an overview of security vulnerabilities, review the existing solutions, and show their limitations.

1. Introduction

Web Services have been used in a variety of applications and have become a key technology in developing business operations on the Web. However, despite the growing use of Web Services by large corporations, security is still a major concern that is slowing their deployment.

Since making Web Services secure means making SOAP messages secure and keeping them secure wherever they go, the group of security standards in WS-Security is used to secure exchanges of SOAP messages in Web Service environment. However, authors in [7] illustrated that the content of a SOAP message protected by an XML Digital Signature as specified in WS-Security can be forged without invalidating the signature. These attacks are called XML Signature Wrapping Attacks or XML Rewriting Attacks. They can happen because XML Digital Signature refers to a signed object of an XML Document in a way that does not take care of the location of that object.

In this paper, we first explore the different form of XML Rewriting Attacks that can take place in Web Service communication. Then we investigate the previous works that have been done for the detection of XML Rewriting attacks and describe their limitations.

The rest of the paper is proceeds as follows. Chapter 2 explains XML Rewriting Attacks. Chapter 3 discusses solutions and demonstrates their limitations. Chapter 4 highlights conclusions.

2. XML Rewriting Attacks

The XML Rewriting attack is an exploitation of the XML tree by reordering the existing nodes or by wrapping nodes without altering the signature. To illustrate this attack we give following example. Suppose a client wants to see the list of airline tickets.

As the client is charged for each request it makes, the request contains a <MessageID>, which is used by the server to keep track of client requests. An attacker, sitting in between the client and the server intercepts the message. He moves <MessageID> element into new fake header element and creates its own <MessageID> element. With this action, the attacker went through the attack detection checks, and by replaying this message, we may cause the same request to be processed several times, making the client pay several times for the same query and forcing the server to do redundant work. This is called Replay Attack which is typical scenario of XML Rewriting attack. Figure 1 demonstrates this attack.

```
<Envelope>
  <Header>
    ...
    <Bogus>
      <MessageID Id="Id-1">uuid:21c81...</MessageID>
    </Bogus>
    <Security mustUnderstand="1">
      <BinarySecurityToken ValueType="...#X509v3"
                           Id="Id-2">
        MIIBxDCCAW6g...</BinarySecurityToken>
      Signature
    </Security>
  </Header>
  <Body Id="Id-3">
    <StockQuoteRequest>...</StockQuoteRequest>
  </Body>
</Envelope>
```

(Figure 1) SOAP message after Replay Attack [3]

Let's consider the scenario where attacker with the slight variation as above method, redirects a SOAP request. Specifically, it wraps the signed <ReplyTo> element, which is used to specify URI of the server, under a bogus header and introduces different <ReplyTo> element with different URI. As we saw in the previous example, with this modification the integrity of the message will not suffer and the application logic will consider the <ReplyTo> element introduced by the attacker instead of the <ReplyTo> element that was wrapped by the bogus header. As a result the message will be redirected to a different location instead of the location it was sent for.

3. Related Studies

There have been numerous researchers who have targeted the problem of protecting SOAP messages from wrapping-attacks. In this section we will describe some of them (Table 1).

Authors in [3] propose an advisor for Web Services security policies. This is a tool that generates a security report by running queries that check for over thirty syntactic conditions. This tool helps to prevent wrapping-attacks in general, however, is not efficient. It is important to note that a simple variant of wrapping-attacks is the deletion of elements. In order to detect this form of attack, every element should be declared as mandatory. This reduces the flexibility

<Table 1> Comparison of selected approaches

Approaches/ Requirements	Policy-Advisor [3]	SOAP Account [8, 5]	Rewriting Healer [2]	Positioning based approach [1]	Bit-Stream [5]	XaT-SOAP [6]
Detection Ability	Yes	Yes	Yes	Yes	Yes	Yes
Warning Analyzing Ability	Yes	No	No	No	Yes	Yes
Decision Making Ability	No	No	No	No	Yes	Yes
Recovery Ability	No	No	No	No	Partial	Yes
SOAP Flexibility	No	Yes	Yes	Yes	No	Yes

of the XML technology and decreases the performance in the validation phase.

In [8] the authors proposed an inline approach that takes into account information about the structure of the SOAP message by adding new header element called SOAP Account. The SOAP Account header contains the number of child element of envelope; the number of header elements; the number of references for signing element; predecessor, successor, and sibling relationship of the signed object. While this approach can prevent certain forms of wrapping-attacks, it is demonstrated in [4] that it cannot address all forms of attacks.

In [4] the authors demonstrated that the state of the art solutions are not addressing all forms of wrapping-attacks and presented some ideas to fix the issue. However, they do not present a complete working solution. A similar work, called RewritingHealer, is conducted in [2] where authors extend the inline approach by proposing to take into account new characteristics of SOAP message such as the depth of information and parent elements of the signed node as well as a way uniquely identify the parents elements. While this extension provides a significant improvement, it is still vulnerable. This is because one can move the signed element together with its parent to somewhere so that the depth of signed element is not changed. In [1] authors proposed to use a new header in SOAP message containing the signed elements positions in the message. While it looks an appropriate way to detect XML Rewriting attacks, it still does not have a solution for recovering task.

In [5], we proposed an approach called BitStream. It works based on the importance of SOAP elements and detects the vulnerabilities and risks, while offering advice for higher security. However, as one of the common types of wrapping attacks is injecting false data into SOAP messages, it is important not only to detect the attacks, but also to recover from them. In [6], we proposed detection and log-based recovery mechanism.

4. Conclusion

In this paper, first, we explored the different form of XML Rewriting Attacks that can take place in Web Service communication. Then we investigated all the previous works that have been done for the detection of XML Rewriting attacks and showed their limitations. Among discussed solutions, we believe that XaT-SOAP approach looks promising because it can detect, analyze warnings, make a decision about recovery and at the same time keep the flexibility of SOAP message.

Acknowledgement

This work was supported by the IT R&D program of MKE/KEIT. [10041854, Development of a smart home service platform with real-time danger prediction and revention for safety residential environments].

Reference

[1] T. S. Barhoom, R. S. K. Rasheed, “Position of Signed Element for SOAP Message Integrity,” In International Journal of Computer Information Systems, Vol.2, No.4, pp. 21-28, 2011.

[2] A. Benameur, F. A. Kadir, and S. Fenet, “XML Rewriting Attacks: Existing Solutions and their Limitations,” In IADIS Applied Computing, April 2008.

[3] K. Bhargavan, C. Fournet, and A. D. Gordon, “An Advisor for Web Services Security Policies,” In Proceeding of the ACM Workshop on Secure Web Services, pp.1-9, 2005.

[4] S. Gajek, L. Liao, and J. Schwenk, "Breaking and Fixing the Inline Approach," In Proceeding of the ACM WorkSupplier on Secure Web Services, pp.37-43, 2007.

[5] P. P. Hung, A. Nasridinov, L. Qing, J.Y. Byun, "A Solution for Injection and Rewriting Attacks on SOAP messages in Web Services Security," In Journal of KIISE: Computing Practices and Letters, Vol.18. No.3, pp. 244-248, 2012.

[6] Nasridinov, A., Hung, P. P., Qing, L., Byun, J.Y.: XaT-SOAP: XML-based Attacks Tolerant SOAP Messages. In: Journal of KIISE: Computing Practices and Letters, Vol.18, No.6 (2012)

[7] M. McIntosh and P. Austel, “XML Signature Element Wrapping Attacks and Countermeasures,” In Proceeding of the ACM Workshop on Secure Web Services, 2005.

[8] M. A. Rahaman, M.Rits and A. Schaad, “An Inline Approach for Secure SOAP Requests and Early Validation,” OWASP Europe Conference, May 2006.