Infiltrate Testing Tool for Web Services Security

S. Kabeer  
M.Tech Student (CSE)  
GATES Institute of Tech, Gooty  
shaik.kabeer@gmail.com

S. Anjini Prasad  
Assistant Professor  
GATES Institute of Tech, Gooty  
prsdanjini@gmail.com

D. Venkatesh  
Dean of CSE & IT  
GATES Institute of Tech, Gooty  
dvvenkatesh@yahoo.co.in

Abstract
For distributed computing solutions Web Services are widely used. Web Services technology is used to integrate existing homogenous or heterogeneous enterprise applications. It can also be used to build inter-operable components that can be reused by many applications irrespective of the platforms in which they are built. Service Oriented Architecture (SOA) is being used for such distributed applications. This architecture enables integration of many services and allows access through a single interface.

As this technology is widely used many extension specifications came into existences which were developed by W3C. This has caused the rise in attacks on web services applications. The attacks include denial of service attacks to various other attacks that break security of the systems. Web application developers generally test their applications for security using penetration testing tools. However, for applications built using Web Services technology no such penetration testing tools are available. Mainka et al. developed a penetration testing tool by name WSAttacker which is plug-in based. They have implemented only two plugins namely SOAPAction Spoofing and WS-Address Spoofing. In this paper we improve the tool by implementing plugins for two more attacks namely Oversize Payload Attack, Oversized Encryption Attack. The WSAttacker is meant for testing web services applications for security. The empirical results revealed that the proposed plugins are effective and they could enhance the use of the tool.

Index Terms – Web Services, security attacks, SOAP, UDDI, WSDL, penetration testing tool

I. INTRODUCTION
Traditional applications serve businesses as they were intended. Systems were already built for many businesses. However, they needed integration. For integration of heterogeneous applications that enable business collaborations such as supply chain management a Web Services technology is used. This technology is based on XML (eXtensible Markup Language) and other W3C standards like WSDL (Web Services Description Language) [1], SOAP (Simple Object Access Protocol) [2], UDDI (Universal Discovery Description Interface), and XML Schema [3]. SOAP is an application level communication protocol which provides standards to render request and responses in XML format. It is useful as the Web Services requests and responses are based on XML. WSDL is another standard which is based on XML. It provides details of web service such as its name, URL to reach it, its operations, arguments and return types. It is essential for every web service as it can allow people to develop client programs that can access web services. UDDI is a standard which is again XML based. It is used to discover deployed web services. In fact web service is a program written in such a way that it can be called from a program written in programming language. Thus it is inter-operable and capable of integrating heterogeneous applications. The web services applications are generally based on an architecture known as SOA which provides standards for such distributed applications. As said already the key standards that are used in SOA include SOAP, WSDL, UDDI, and XML.

With the success of the Web Services technology and SOA architecture, it is quickly understood that these standards are to be used with complete security. For this reason other standards came up as extensions to Web Services technology. The OASIS consortium developed some standards known as WS-Security [4], WS-Security Policy [5], SAML [6] and XACML [7]. Due to the complexity of these specifications,
they are not being implemented properly. This has led to the emergence of attacks specific to web services. Some of the well-known attacks pertaining to web services [8] are discussed in this paper. Among the attacks, the important attacks are the ones that break cryptographic primitives. Signed messages can also be broken through XML signature wrapping attacks as explored in [9]. The impact of this kind of attacks is demonstrated on web services interfaces of Eucalyptus cloud and also Amazon EC2 SOAP [10], [11]. It is also proved in other Single – Sign on frameworks such as SAML [12]. Another attack of that kind in this area is explored in [13] which are related to decrypting XML cipher texts. The server side implementations which are counter measures to this attack are having drawbacks. [14]. A series of Denial of Service (DoS) attacks are also possible on web services applications.

The attacks pertaining to web services include middleware hijacking, indirect flooding, WS-Addressing spoofing, BPEL state deviation, instantiation flooding, attack obfuscation, oversized cryptography, WSDL scanning, metadata XML injection, coercive parsing and XML signature wrapping attack.

Usually the developers of applications that are based on web services are not able to identify security vulnerabilities of web services interface implemented. When compared with attacks such as SQL Injection and XSS (Cross Site Scripting) that have tools to conduct penetration testing, the XML – specific vulnerabilities can’t be automatically detected. In [15] that effort has been made by implementing a penetration testing tool known as WSAttacker. This tool’s architecture is very flexible and modular in nature.

This tool’s design supports extension. Especially its Plugins Architecture supports adding new plugins for supporting other attacks for penetration testing. The WSAttacker is believed to be the first automatic penetration testing tool for testing application made up of web services. It supports testing two web services related attacks known as SOAPAction spoofing attack and WS-Address Spoofing attack. It does means that they developed only two plugins in the WSAttacker.

In this paper, our main contribution is as follows.

- Implementation of Oversize Payload Attack plug-in. This plug-in tests web services to find vulnerabilities that may lead to oversized payload attack.
- Implementation of Oversized Cryptography attack plug-in. This plug-in tests web services to find vulnerabilities that may lead to oversized cryptography attack.

The remainder of this paper is organized into some sections. Section II describes WSAttacker tool. Section III provides information about the proposed plugins. Section IV evaluates the plugins while section V concludes the paper.

II. WSAttacker TOOL

Mainka et al. [15] have implemented a penetration testing tool for web services security. This tool is plugin based and can launch two kinds of attacks on applications based on web services. The attacks include SOAPAction Spoofing Attack and WS-Addressing Spoofing attack.

**SOAPAction Spoofing**

This is one of the possible attacks on web services [8]. This attack misuses one of the parameters of HTTP known as SOAPAction. This attack is described here. Imagine there are two operations in a web services such as Op1 and Op2. The WSDL file for this file contains details of the web service including SOAPAction for the operations of the web service. Imaging Action1 and Action2 are the actions pertaining to the web service. These actions correspond to the operations of web service. Figure 1 visualizes the attack.

Fig. 1 – Illustrates SOAPAction spoofing attack (excerpt from [15])

As can be seen in fig. 1, the logic of web service is executed as present in the body without checking authentication. Moreover it believes that the authentication is performed by firewall. If operation B is something like deleteAllCustomers(), the SOAPAction Spoofing attack may cause the problem of deleting all customers from the database.

**WS-Addressing Spoofing**
It is also a web service – specific attack. In this case, an attacker sends SOAP request to server where the server has access to WS-Addressing header which allows the server to send response in SOAP standard to different endpoint. This attack is illustrated in fig. 2.

As shown in fig. 2, the WS-Address Spoofing attack is able to deceive the server to send the response to different endpoint instead of the intended endpoint.

**Implementation of WSAttacker**

Manika et al. [15] implemented WSAttacker which is a penetration testing tool for web services security. They made it modular and plug-in based. It has a flexible design. Its design has two important parts namely the WSAttacker framework and Plug-in Architecture.

The framework has a set of activities that allow the attacks and present discovered vulnerabilities. The plug-in architecture holds one or more plugins that can be used to launch attacks on web services. Fig. 3 shows the implementation details.

Fig. 2 – Illustrates SOAPAction spoofing attack (excerpt from [15])

As illustrated in fig. 3 the framework enables loading WSDL file. Then it allows the user to select an operation supported by web service. Afterwards the user generates request content and submits a test request. User can also configure attacks and start specific attacks on the chosen operations of web service. Then the results of the attack are presented. Attack configuration includes specifying plugins required by the attacks.

**III. PROPOSED IMPLEMENTATION OF ATTACK PLUGINS**

In this paper we implemented two plugins that are compatible with WSAttacker. The plugins are used to launch two more web service related attacks. They are known as oversize payload attack and oversized cryptography attack. Before going to implementation details we first describe these attacks here.

**Oversize Payload Attack**

XML documents are verbose in nature. They are meant for sharing data across many applications. They are also human readable. They are an improvement over binary format and they are very essential in web service applications. Such documents are validated by XML parser. Only valid XML documents that conform to DTD (Document Type Definition) or Schema are allowed for further processing. If not the buffer overflow attack is possible on such XML file. When web services have vulnerability, an attacker can change the XML
content to be very huge content (many MBs) so as to disturb the normal processing flow. Sometimes, multimedia data might have large MBs of data. But the application should have mechanisms to know it and process piece by piece accordingly. Some applications might load an XML document into RAM before validating them. Such applications vulnerable to oversize payload attack [16].

**Oversized Encryption Attack**

Security features can be flexibly used in web services. The features such as signature algorithms, cryptography can be applied to different parts of SOAP messages with signed and encrypted possibilities. Thus it is made flexible. This flexibility is used by hackers to make recursive cryptography attacks. These attacks are also known as oversized encryption attacks. The attacks are aimed at causing exhaustion of system resources with respect the web service under attack [17].

**Implementation of Oversize Payload Attack**

The abstract plug-in interface given by Manika et al. [15] is implemented for this attack. This interface is reused to be consistent with WSAttacker tool specifications. The implementation steps are as given below.

1. Finding endpoints of web service.
2. Finding the payload verification details.
3. If there is no payload verification mechanism, inject huge amount of XML content into the SOAP body.
4. Ensure that the injected XML content disrupts the normal execution flow or cause the server to use huge amount of system resources unnecessarily and give unintended results.

**Implementation of Oversized Cryptography Attack**

Here also the abstract plug-in interface given by Manika et al. [15] is implemented for this attack. This interface is reused to be consistent with WSAttacker tool specifications. The implementation steps are as given below.

1. Finding endpoints of web service.
2. Finding whether encrypted element or signature element is processed based on the information provided in the header element.
3. If there is no encryption part in the chosen web service, the plug-in does nothing.
4. Else it performs this attack which recursively encrypts the content until the system resources are exhausted.

**IV. EXPERIMENTAL RESULTS**

Experiments are carried out on a PC with 4 GB RAM and Core 2 processor. Web services are running in Tomcat server where Apache Axis is deployed. The web services are pre-existing in an order processing system built by us earlier. There are five web services and the attacks are launched on all web services. The results revealed that the web services are vulnerable to the four kinds of attacks namely WS-Addressing Spoofing attack, SOAPAction Spoofing attack, Oversize Payload attack and Oversized Cryptography attack. Fig. 4 shows the number of vulnerabilities exhibited by web services.

![Fig. 4 – Number of Vulnerabilities Exhibited by Web Services](image)

As can be seen in fig. 4, there are web service names presented in horizontal axis and the number of attacks is presented in vertical axis. Out of them all the transaction service has encryption option. Thus it has all vulnerabilities including Oversized Encryption Attack. This attack is not possible with other web services as they do not have encryption process. All web services that are vulnerable to three attacks indicate that they are vulnerable to all attacks except Oversized Encryption attack. The verification service web service is vulnerable to two attacks namely WS-Address Spoofing attack and SOAPAction Spoofing attack.

Oversized encryption attack is the attack which causes the resources of the system to get exhausted. Only one web service by name TransactionService is vulnerable to oversized encryption attack. The impact of this attack is presented in fig. 5 and fig. 6.
Fig. 5 – Impact of Oversized Encryption attack
As can be seen in fig. 5, the impact of the attack is evident. After attack the RAM usage is increased. It does mean the RAM is exhausted.

CONCLUSION
In this paper we proposed two plugins to enhance the WSAttacker tool proposed by Mainka et al. The WSAttacker tool is the first of its kind meant for penetration testing of web services. This tool’s architecture is flexible and modular. The tool has two plugins already implemented for testing two types of web services attacks namely SOAPAction Spoofing attack and WS-Addressing Spoofing attack. These two plugins are used to discover the related vulnerabilities in web services applications. In this paper we enhanced the tool in order to implement two more plugins for Oversize Payload attack, and oversized encryption attack. These plugins help in discovering vulnerabilities in web services applications that might be exploited by hackers to perform oversize payload and oversized encryption attacks. The plugins implemented in this paper are tested. The empirical results revealed that the plugins are capable of discovering vulnerabilities.

REFERENCES:


