|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| VILNIAUS KOLEGIJA  UNIVERSITY OF APPLIED SCIENCES  FACULTY OF ELECTRONICS AND INFORMATICS  Image result for viko logo | | |  | | | VILNIUS COLLEGE  Image result for viko logoFACULTY OF ELECTRONICS AND INFORMATICS |
|  | | |  | | |  |
| Essay  Web Service security and malicious content filtering. | | |  | | | **INTRODUCTION TO INFORMATICS** |
| 6531BX028 PI18E | | |  | | | PRACTICAL ASSIGNMENT  SPOTIFY USER MANUAL  6531BX028 PI18E |
| STUDENT | LUKAŠ JUTKEVIČ | STUDENT | | DŽIUGAS PEČIULEVIČIUS |
| (SIGNATURE)  4/8/2020 | | |  | | | LECTURER |
| LECTURER | 1. SAVICKAITE | (SIGNATURE)  10/17/2018 | | VIRGILIJUS KUKLIERIUS |
| (SIGNATURE)  4/8/2020 | | |  | | | 2018 |

2020

# TABLE OF CONTENTS

[TABLE OF CONTENTS 2](#_Toc37197659)

[Introduction 3](#_Toc37197660)

[Web services 3](#_Toc37197661)

[Web service Protocol Stack 4](#_Toc37197662)

[Web service security 4](#_Toc37197663)

[Malicious content 5](#_Toc37197664)

[Sources 8](#_Toc37197665)

# Introduction

Nowadays a huge number of people have access to the internet, and this means that immense quality of information is uploaded to various databases every day which leads to more and more resources being allocated to information storage and processing. Before all this information, be it some artists blog post, celebrities’ tweet or your friends Instagram photo, is stored on some remote data server it needs to be processed. And here several issues arise: how securely send data from one network location to the other? And how filter out malicious or unwanted data? Information sending thru the computer networks is handled by web services.

Web services are the underlying software behind todays internet. They control the flow of information from database servers to end-user’s web browsers, mobile phones, various web applications etc. and back

# Web services

Web service (see Figure 1) is essentially the piece of software that is utilised by for example another web server, or mobile application that provides user interface to the end user to exchange information over the internet. They are built on top of open standards such as HTTP, HTML, SOAP, etc.

(TutorialsPoint, Web Services, 2020/05/21)

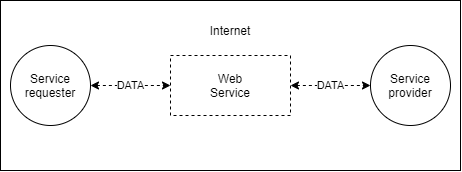


Figure 1. Example of a Web Service

In addition to that web services use standardized XML (Extensible Markup Language) messaging system to provide an interface where two-way communication occurs between service requester and service provider. This communication is made from messages that contain data structured using XML tags. Messages can be transferred using SOAP (Simple Object Transfer Protocol) which is a messaging protocol specification for exchanging structured information used in various implementations of web services.

This two-way communication is initiated when service requester (i.e. “Client”) sends XML message to web service and then waits for a corresponding XML response.



Figure 2. Example of how SOAP is used to request and receive data.

Major advantage of web services is that they are language independent, this means that they can be implemented in different programming languages and on different platforms and can communicate with different languages on different platforms since all communication is XML based. One, obvious limitation is that services requester and provider need to be able to create and parse SOAP XML message which similarly to HTML document has header and body that’s wrapped in an envelope (see Figure 3).

(Wikipedia, Web Service, 2020/05/21)

(TutorialsPoint, SOAP, 2020/05/21)

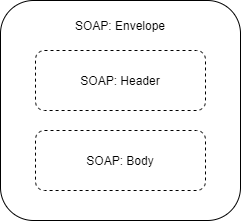


Figure 3. SAOP XML messages structure.

## Web service Protocol Stack

There are 4 major parts to web service all of them together are called “web service protocol stack”. These parts combined result in a web service

**Service Transport protocol:**

This Service part is responsible for transporting messages from service provider to service requesters. Service transport protocol includes such protocols as Hyper Text Transport Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), and newer protocols such as Blocks Extensible Exchange Protocol (BEEP).

**XML Messaging protocol:**

This protocol is responsible for encoding messages in a common XML format so that messages can be understood by service provider and requester. XML messaging is usually handled by SOAP or XML-RPC (XML remote Procedure Call Protocol)

**Service Description protocol:**

This service part is responsible for describing interface to a web service. Service description usually is handled via the Web Service Description Language (WSDL).

**Service Discovery protocol:**

This layer is responsible for centralizing services into a common registry and providing easy publish/find functionality. Service discovery is usually handled via Universal Description, Discovery, and Integration (UDDI).

## WSDL and UDDI

One of the main questions that arise when one starts to dive deep into webservice architecture is how service requester knows what methods service provider provides. Not only that, but when sending soap message what variables need to be sent and how they need to be named. WDSL (Web Services Description Language) helps to solve this problem.

It’s an XML based interface description language that’s used for describing the functionality offered by the SOAP web service. WSDL file provides machine readable description of how web service method can be called, what method returns and what parameters it expects in a request

WSDL uses network endpoints as an access point to the SOAP methods all SOAP requests are sent by service requester to those endpoints and then processes by webservice and they are described in WSDL file.

To be easily accessible web service providers publish WDSL files on “Universal Description, Discovery, and Integration” (UDDI) which is an XML based registry for web services that facilitates the discovery of web services for web service users that can browse UDDI manually or automatically.

UDDI consists of 3 parts that are called pages:

1. White pages – give information on business that’s supplying the web service, mainly name, description and contact information.
2. Yellow pages – provide a classification of the webservice, since one business can offer multiple web services there may be multiple yellow pages in relation to one white page.
3. Green pages – describe how to access the web service and provide its WDSL file.

(Wikipedia, Web Services Discovery, 2020/05/26)

(Tutorials Point, WSDL, 2020/05/26)

# WordPress Based SOAP Webservice Demonstration

To demonstrate how SOAP based webservice works I created very simple WordPress blog and installed a plugin that exposes some webservice functionality of my blog, for example I can log in using SOAP retrieve user and post information.

Web Page link: <http://calmdownlebek.000webhostapp.com/>



Figure 4 WSDL file example

This is a WDSL file describes SOAP APIs methods from my WordPress webpage:

Endpoint tag is named *<soap:address />*

Method tag is named *<operation name=’getPost’>,* it describes methods input and output with according XML tags, in this case they are:

*<input message='tns:getPostRequest'/>* - method request

*<output message='tns:getPostResponse'/>* - method response

In message tags you can see message parameter names and types for example getPost methods request has 2 parameters named *token* and *postid.*

## SOAP requests

To sent SOAP requests to my WordPress page ill use a tool called postman that lets user to manually create API requests and send them.

1. *authenticate* request message:

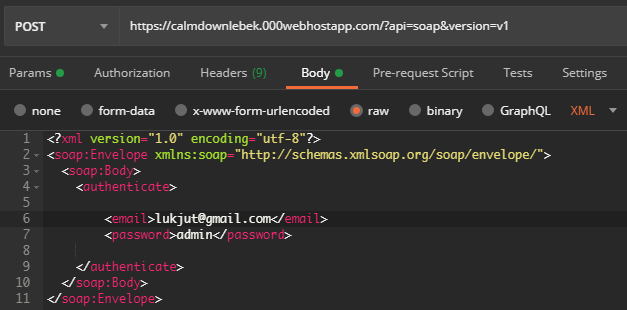


Figure 5 Authenticate request message.

As you can see this SOAP message has an envelope and body, method is described in body part and is called *authenticate* with parameters called *email* and *password.* This SOAP message is sent to the endpoint located at: [*https://calmdownlebek.000webhostapp.com/?api=soap&version=v1*](https://calmdownlebek.000webhostapp.com/?api=soap&version=v1)

Web service response:

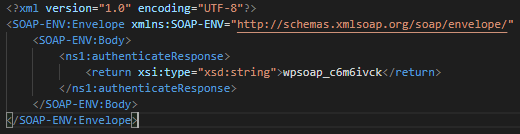


Figure 6 Authenticate response message.

Web service method returned a string in this case *“wpsoap\_c6m6ivck”* which is current sessions tokenand need to be used to save service requesters identity for future requests.

1. *getUser* request message:

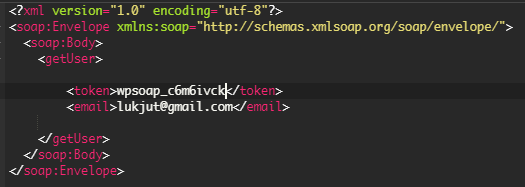


Figure 7 getUser request message.

Same as last request this one has envelope and body method name is *getUser* and it expects 2 parameters session token and requested users’ email.

Web service response

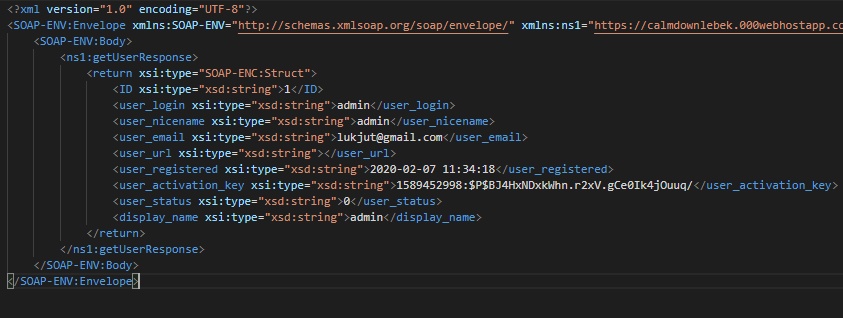


Figure 8 getUser response message.

This method returned information on requested user.

1. *getPost* request message:

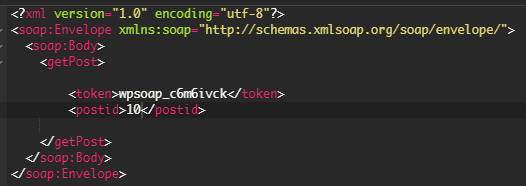
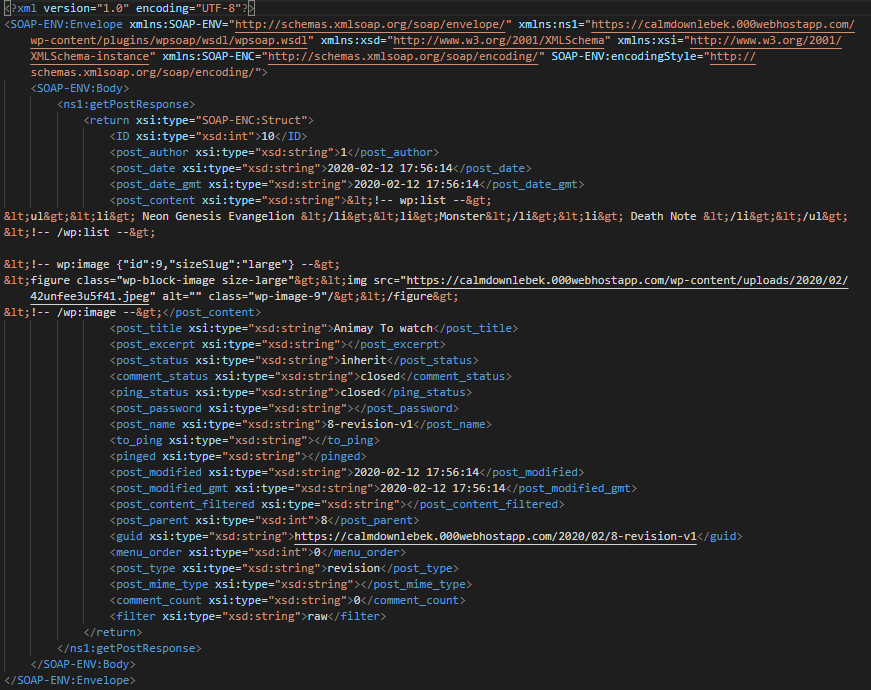


Figure 9 getPost request message.

*getPost* method requires postid and session token and return XML file with formatted information on the post:



Link to this particular post:

[*http://calmdownlebek.000webhostapp.com/2020/02/animay-to-watch*](http://calmdownlebek.000webhostapp.com/2020/02/animay-to-watch)

# Web service security

Security is a critical aspect of a web service, secure information transfer is needed to ensure that data won’t be hijacked by malicious actors that shouldn’t be able to access it. However, most commonly used XML messaging protocol – SOAP lacks any security or authentication requirements. There are three main security issues associated with web services:

Confidentiality – if client initiates communication with a server by sending an XML request to a server how to ensure that this communication remains confidential?

Authentication – if client connects to a web server how to identify him? How to ensure that client is authorized to use the server?

Malicious content – malicious content inside otherwise valid SOAP messages can cause immense damage to service provider and data servers behind the web service.

Confidentiality issue can be mainly solved by encrypting all web service traffic (i.e. XML messages) this can be easily done since SOAP and XML-RPC messaging protocols primarily use HTTP as a transport protocol to send messages and HTTP has support for Transport Layer Security (TLS). Therefore, all web service communication can be encrypted via HTTP over TLS (i.e. HTTPS, See Figure 4) which is widely used technology that ensures secure communication over the internet.

(Oracle, Understanding Web Services Security Concepts, 2020/05/21)

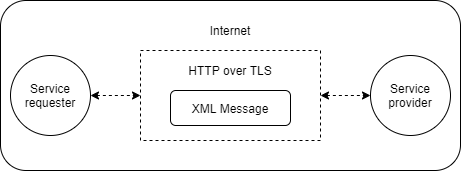


Figure 10. XML message transmitted over HTTPS.

Second issue, that is Authentication issue can easily be solved by requiring client to provide his credentials (i.e. password). HTTP and HTTPS already have built-in support for basic and digest authentication (basic authentication: credentials are sent not encrypted; digest authentication: credentials are being encrypted). Using HTTPs authentication will protect services similarly how HTML documents are protected.

## Malicious content

Malicious Content, Viruses, or Trojan horse programs, being transmitted inside otherwise valid XML messages can easily damage Web service applications. Binary data that can be stored and sent inside SOAP messages such as images or executables can be modified to cause exceptions inside web service application. In addition to that many tools like spreadsheets can be corrupted to include malicious code or otherwise illegal content that can crash applications or degrade webservice performance (See Figure 5). So called SQL (Structured Query Language) Injections are one more type of malicious content that can be hidden inside XML document. SQL statements are used to manipulate database records therefore if one is sent from the client to the web service it can be used to retrieve, update, delete or insert information in database. There statements should not be inside SOAP messages, they must be detected and dealt with before being received by the server.

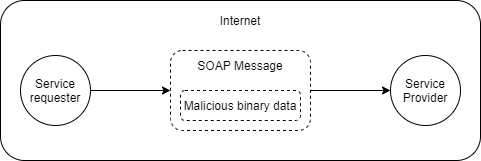


Figure 11 Diagram of how malicious code can be sent using SOAP.

To prevent malicious content related security threats essentially every SOAP message needs to be inspected to ensure that all attacks are caught before they can cause any real damage. Traditional solution to control data coming in from the internet to computer is a firewall, but a standard firewall can’t be used to inspect SOAP messages for malicious content. Network firewalls are designed to allow HTTP and HTTPS traffic to pass through by checking if HTTP/HTTPS requests are valid according to the syntax of the protocol and not by requests content. Therefore, network firewalls can’t be used to check whether SOAP message is malicious, corrupted or otherwise damaging.

One alternative to catch malicious SOAP messages is network traffic filtering by inspecting XML and SOAP syntax and content. filtering allows for intelligent decision to be made regarding which messages should be allowed to be parsed and interpreted. By capturing and scanning XML and SOAP messages before they are interpreted by the application relevant and safe data can be passed through and inappropriate and malicious data can be filtered out before any damage is done. In addition to that message content can be filtered for any suspicious text, commands and unsafe SQL queries to prevent SQL injection attacks from occurring.

The biggest drawback of filtering SOAP and XML messages is performance decrease that comes with it. Parsing big messages can degrade application performance

(Forum Systems, Anatomy of a Web Services Attack, 2020/05/21)

(NIST, Guide to Secure Web Services, 2020/05/21)

# Conclusions

Today when amount of information exchanged via internet grows with every hour and more and more people trying to profit by exploiting, changing or accessing this information, web service security is crucial. SOAP and XML messages need to be encrypted for information being sent to remain confidential. Authentication need to be employees to ensure that only authorized clients can access the data through web service. But most crucially the applications and servers behind web services need to be protected from malicious content and possible attacks. Due to the structure of web services any malicious content or attack being sent using XML or SOAP messages needs to be detected before message is interpreted which comes with considerable performance cost.

# References

*Anatomy of a Web Services Attack*. Retrieved from https://www.forumsys.com/wp-content/uploads/2014/01/Anatomy-of-a-Web-Services-Attack.pdf

Oracle. *Understanding Web Services Security Concepts*. Retrieved from https://docs.oracle.com/cd/E23943\_01/web.1111/b32511/intro\_security.htm#WSSEC2342

National Institute of Standarts and Technology, *Guide to Secure Web Services*. Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-95.pdf

TutorialsPoint. *Web Services*. Retrieved from https://www.tutorialspoint.com/webservices/index.htm

TutorialsPoint. *SOAP*. Retrieved from https://www.tutorialspoint.com/soap/index.htm

Wikipedia. *WebService*. Retrieved from https://en.wikipedia.org/wiki/Web\_service

TutorialsPoint *WDSL*. Retrieved from https://www.tutorialspoint.com/wsdl/wsdl\_introduction.htm

Wikipedia. *Web setvices discovery*. Retrieved from https://en.wikipedia.org/wiki/Web\_Services\_Discovery