Modified Authentication using One Time Password to Support Web Services Security

Adang Suhendra
Gunadarma University
Department of Industrial Technology, Software Engineering
Jalan Margonda Raya no. 100
adang@staff.gunadarma.ac.id

Anne Yulianti, Bismar Junatas, and Vega Valentine
Gunadarma University
Department of Industrial Technology, Software Engineering
Jalan Margonda Raya no. 100
anneyulianti, lepen_72, slaved_jepun@student.gunadarma.ac.id

Abstract

Freshness in accessing a web service is a challenge identified by the security of the website itself. It usually including advantages and disadvantages on architecture security and the machine language used by the site. With the recent technology available completed by sufficient information about risks thread in web services, there are a lot of things to be concerned. In this paper, we try to inform what actually a web service is, what are the risks we will face, and several techniques in handling web services security. This paper also propose a new design of authentication model added with one-time password (OTP) generator-source code written in Java, as a technique to enhance its security.

1. Introduction

The term web service is used to describe the ability to easily link programs and data from various sources in a way that creates a new look at the data or even a new application [9].

Because a Web service relies on some of the same underlying HTTP and web-based architecture as common web applications, it is susceptible to similar threats and vulnerabilities [2].

Web services security is based on several important concepts [6], including:

- **Identification and Authentication**. Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.
- **Authorization**. The permission to use a computer resource, granted, directly or indirectly, by an application or system owner.
- **Integrity**. The property that data has not been altered in an unauthorized manner while in storage, during processing, or in transit.
- **Non-repudiation**. Assurance that the sender of information is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later deny having processed the information.
- **Confidentiality**. Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.
- **Privacy**. Restricting access to subscriber or relying party information in accordance with Federal law and organization policy.

We only focus in point 1, which is the authentication part. Authentication is the process by which a system can determine whether or not a given user is who he/she claims to be. In the context of Java runtime environment, it is the process of identifying the user of an executing Java program [8]. Authentication is the key for information security since if the authentication mechanism is compromised, the rest of the security measures are bypassed as well [4].

One-time password (OTP) schemes, where each password is used only once, offer a viable alternative or a supplement to traditional password schemes [4]. Variations of OTP schemes include ”sequentially updated OTPs” where there is initially only a single secret password that is shared and the user creates and transmits the new password while he is being authenticated with the previous one, and ”shared lists of OTPs” where each user uses a set of passwords each valid for a single authentication and distributed as a pre-shared list [1].

Modifying an authentication technique has already done by Karl Ackerman, Piers Bowness, John Brainard, and Bill Duane from RSA Security Bedford, USA. They were proposed a technique called Disconnected Authentication (DAUTH) that allows OTPs to be verified without access to a server. The DAUTH scheme allows OTPs to be used for local authentication without exposing the long-term secret used to generate the OTP values [5].

Kemal Bicakci and Nazife Baykal from Middle East Technical University also done some modifications in OTP method. They are improving the security and the flexibility of OTP by using public key techniques. The method they have made is called Signature Chain Alternative to Lamport’s Hash Chain [4].

In this paper, we propose a method that combines authentication and one-time password which covers each other disadvantages. We know that authentication is vulnerable on sniffing or a middle man stealing user password, so that he can use user’s id to do some actions that usually adverse others. By adding the procedure with a method called ’key checking’ that generates one-time password for the user, we hope that it would improve the security of the whole authentication process.

This design hopefully can be implemented in some organization or institution (such as an office, for the employees to access certain confidential data, or a university, for the student or lecturers to access their personal information in university’s database through the web services) to give trust for the user, so that they can access the web services freely without any anxiousness on sniffing or another irresponsible third party.

2. Model Design, Analysis, and Implementation

2.1. Comparison of Original and OTP-Authentication Principles

Authentication is the process by which a user proves their identity to a system. This proof can take several forms:

- A password known only to the user.
- A digital certificate (a secret key contained in a normal file).
- A physical Smart-Card (similar to a Mondex card) inserted into a special reader.
- A biometric measurement such as a fingerprint.

OTP schemes, where each password is used only once, offer a viable alternative or a supplement to traditional password schemes. OTP schemes’ advantages depend on which mode of operation is used. The modes of operation and corresponding advantages are as follows [4]:

1. In the first mode of operation, to facilitate user-friendliness, each user has only (and should memorize) one password just as in traditional passwords. This password is used to authenticate the user to the client machine (workstation) and then this machine generates the OTP to be sent to the server. On the network-connecting the client and the server machines, only the OTP is transmitted.

2. For applications that require stronger security, it is possible to have the user enter the OTPs without getting any help from the system. Of course, we cannot expect someone to memorize all these OTPs. Now we are in a so-called human and paper environment where OTPs are listed on a piece of paper and carried in the pocket (alternatively, mobile devices such as PDAs and cellular phones can be used for storing the OTP list). This is an inconvenience for the user, but in most applications demanding a high level of security this inconvenience is tolerable.

2.2. Model Design: Original vs. OTP-Authentication

In this section, we will show the original authentication model with the one we have made called ’OTP-Authentication’.

2.2.1 Original Model: Plain Text Passwords

Figure 1 shows the simplest basic model of authentication named ’Plain Text Password Technique’. This model explained by Paul Anderson in his paper ”Authentication Models Explained: A Background to Single-Sign-On Issues for the University of Edinburgh”.

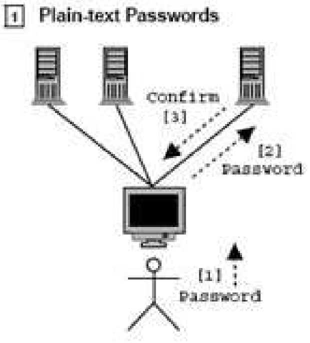

Figure 1. Model design of original authentication technique

Note: the numbers on the diagram define the sequence of operations.

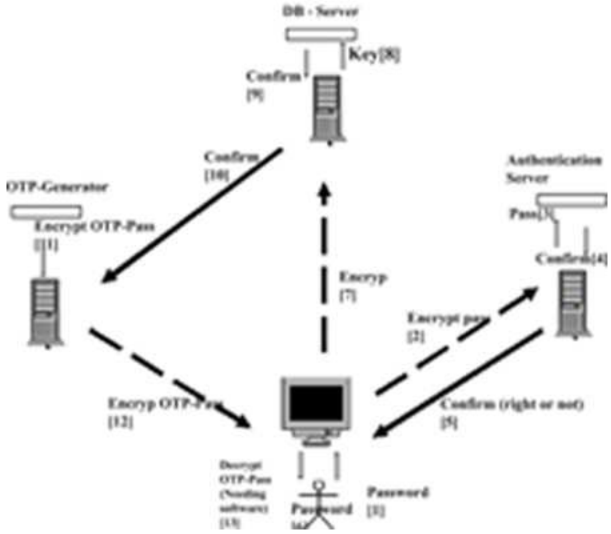

Figure 2. Model Design of OTP-Authentication

2.2.2 New Model: OTP-Authentication

Figure 2 shows the modified model of the original authentication model. Modifications done by adding two procedures after the authentication process confirmed. Several addition techniques and mechanisms are:

- Encrypted password and key.
- Encrypted channel, between services to secure key passing through the system.
- Synchronized key techniques, to do key checking procedure. The key entered by user and the one stored in the database is synchronized, then database server pass it to OTP generator as shown in figure 2 procedure.
- OTP generator, generates OTP for users.
- Delegation or multiple tiers. For example, the user may connect to a web server, and the web server may then connect to a separate database server. Somehow, the web server needs to pass the user's authentication on to the database server [9].

Details on its mechanism explained in 2.3.2.

2.3. Analysis: The Original Model vs. OTP-Authentication

The analysis stressed on the sequence of operations and both advantages and disadvantages of each model to see the difference, and then see whether the new model brings more advantages or effectiveness.

2.3.1 Original Model: Plain Text Passwords

- **Procedure**

  In this simple case, each remote service as its own authentication mechanism and the user types the corresponding password when required by the service.

- **Advantages**

  Provided that the user has been sensible enough to choose separate passwords for the different services, then if one of the passwords is compromised, there is less chance of other services being endangered.

- **Disadvantages**

  - Remembering the various passwords, matching them with their corresponding services, and changing them, are a problem for the user. This often leads to a bad choice of passwords, or even the use of a single password for all services (see below).
  - The passwords can easily be sniffed by anyone with the ability to connect to the network between the client and server. This is becoming less likely inside the University itself (although it is still a problem), but more likely overall, due to increasing use of remote access (for example, from Internet Cafes).
  - Account management is difficult since new users must be added to every service individually.

2.3.2 New Model: OTP-Authentication

- **Procedure**
  - User enters username and password he/she has, then having the confirmation whether the information sent is true.
  - If the information were true, then the user has to enter supporting information called 'key' corresponding to the information recorded in the database server. The database server then check whether the key entered by user and the one in the database is equal. This procedure called 'key checking'.
  - If both of the key is equal, next step is the server role to send this confirmation to the OTP generator to generate OTP for the corresponding user.
  - User got the OTP, then finally could log on to the network using this OTP and accessing the web services available in the system.
- **Advantages**
  - Prevents password sniffing that could be happened in original model, by using encrypted text for every passwords or keys passes the network, or encrypted channel between services.
  - It is true that a copy of the password passed through client and server machine. But remember that user just uses it as a shield to get OTP, so we still can say that it is relatively secure.
  - Each service requires different password or key. This means that if any of these services is compromised, this system is not in serious danger.
  - Compromise of any single server will not fully affect all services, because not all services are related directly to each other.
  - Authentication delegation is supported.
- **Disadvantages**
  - For this to be effective, it is very important that user understands the correct procedures for management of the encryption protocol.
  - It might be quite complicated for the user to do first log on procedure (undergo two procedures to get OTP).
  - User has to memorize the password-e.g. write it on a paper, so the difficulties are still faced.
  - Special software is required on the client to decrypt the OTP(some free software available at: http://www.encryptionanddecryption.com/download.html#free).

2.4 Implementation of OTP-Authentication

This method could be implemented by any organizations or institutions need a secure data accessing through web service.

For example, Gunadarma University has a lot of hotspots spread over many locations in the campus. In order to enable only Gunadarma's staffs and students to use the hotspot, such as to upload or download subject materials or checking their personal information provided in staff or student mails, so the authentication method is needed. In this context, authentication is needed to ensure that the user is the real affiliate of Gunadarma.

So, to avoid password sniffing by irresponsible third party, we can use the proposed OTP-Authentication. This method could keep the user id and password securely while passing over the network and also could generate a trusty OTP which can only be used once-also tells us that the OTP is exclusive and secure.

3. Result

From the experiment and its analysis, we got the result that OTP-Authentication has more ways to protect the system from threads.

In this method, OTP-Authentication encrypt and decrypt passwords, keys, and password's confirmation from client to server or server to user. By encrypting and decrypting the passwords, it is quite secure to againts the thread.

Comparing between original authentication and OTP-Authenthification, OTP-Authentication is more superior than the original. It is because there is a combination between authentication and OTP. Then, having more ways to protect the system from threads which using enrcypt and decrypt each time it delivers the important message from one sever to another server. Then, every server receive and sending different information of passwords, keys, and password's confirmation which could flam sniffers.

Original authentication does not use OTP generator and in original authentication, the password is in plain text format or have not been hidden by encryption. Besides, the password is passed through all server which is the main weakness of original authentication. It means that if one of services are hacked then it is easier to stole all client's information.

4. Conclusion

A Web service security model should address security issue involved in a request from an end client to a target service, including the intermediary services that route the service requests.

In this paper, we have proposed a mechanism, which is a combination between original authentication and OTP application. This new model hopefully could help the client to provide authentication data based on the service definition and, at the same time, help the service provider to retrieve those data.

For final completion of the model, we can added a VPN (Virtual Private Network) to 'virtualize' the network while the server delivers a confirmation or keys, instead of encrypting all data sent.

This method can works optimally using several software support. For the OTP generator, we can use JOTP calculator written in Java by Harry Mantakos. We can get the generator at: http://cs.umd.edu/h̃arry/jotp. And for a software needed to be built in the client machine, we can use Encryption and Decryption Pro software, which is a freeware that you can get from: http://www.encryptionanddecryption.com/ download.html#pro.

This design hopefully can be implemented in some organization or institution (such as an office, for the employees to access certain confidential data, or a university, for the student or lecturers to access their personal information in university's database through the web services) to give trust for the user, so that they can access the web services freely without any anxiousness on sniffing or another irresponsible third party.

Unfortunately, every security have their weakness and surplus and we got it back to our thought that every architecture and every methode or way that we use have to developed and upgrade. By developing and upgrading could help us to less the threads.

References

[1] S. V. A. Menezes, P. Van Oorshot. *"Handbook of Applied Cryptography"*. CRC Press, 1996.

[2] T. W. A. Singhal and K. Scarfone. Guide to secure web services. *Gaithesburg: NIST Special Publication 800-95*, 2007.

[3] P. Anderson. *"Authentication Models Explained: A background to single-sign-on issues for the University of Edinburgh"*, volume 1-9. Revision 1.1 edition, 2003.

[4] K. Bicakci and N. Baykal. Improving the security and flexibility of one-time password by signature chains. *Turk J Elec Engin*, (3):1–3, 2003.

[5] J. B. B. D. K. Ackerman, P. Bowness. Dauth: Secure offline verification of one-time passwords. *RSA Security*, MA01730:1–2.

[6] R. Kissel. Glossary of key information security terms, information security handbook: A guide for managers. *National Institut of Standard and Technology, NIST IR 7298, NIST SP 800-100*, pages 1–87.

[7] A. N. M. Hondo, N. Nagaratnam. Securing web services. *IBM System Journal*, 41(2):1–2, 2002.

[8] Sun Mycrosystem Inc. *"Java Security Overview"*, 2005. 10-11.

[9] P. J. Windley. *"Enabling Web Services"*. http://www.windley.com, 2003.