Volume 3, Issue 10, October 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

User Password Authentication Using Two-Server Architecture

[1]Kasula Tejaswini, [2]P. Akshita Rao, [3]Preethika Dorothy Shyamsundar, [4]Sudha S

[1,2,3] M.S (SE), SITE, VIT University, Vellore, India

[4]Asst. Professor Senior, SITE, VIT University, Vellore, India

Abstract: ***In today's computer era, password cracking and hacking is seen almost everywhere, which makes the user authentication for any kind of application very important. The concept of user id and password is one of the easiest ways of user authentication. Generally, most of the password based authentication systems use a single server which stores all the information which is necessary. This kind of architecture will make the system prone to various security issues like offline dictionary attacks and brute force attacks. To ensure higher security to the user's critical data, a number of schemes with multiple servers have been proposed in which the password is shared by the servers and these servers cooperate in a threshold manner in which the user wants to authenticate. In this paper, we are proposing a new and efficient two-server architecture with a one-way technique scheme. Compared to the other schemes in the literature review, our proposed scheme is much simpler and provides more security to the user's important information.***

Index terms: ***Problem, two-server, one way technique, fixed salt, variable salt, offline dictionary attacks, birthday attacks, brute force attacks, MD5(message digest) algorithm.***

1. Introduction:

Password based user authentication systems are low cost and easy to use. A user only needs to memorize a short password and can be authenticated anywhere, anytime, regardless of the types of access devices he/she employs. Password based authentication system is still gaining popularity even in the presence of several alternative strong authentication approaches, e.g., two factor authentication and biometrics. The reason for this is, it does not require any additional devices or tokens like in biometrics and two factor authentication systems respectively. In two factor authentication system the loss or theft of the token not only risks disclosing the secrets inside but also disables the authentication functionality. In recent years, much attention has focused on designing password based authenticated key exchange protocols which can resist any kind of intruder's attack. To solve this problem, a new kind of authentication structure called the multiple server authentication was proposed. In such schemes, the capability of verifying a password is split between two or more servers, and more than a certain threshold number of servers need to collude to recover the password. Till now, few multiple server schemes were proposed. In these multiple server authentication settings, the two-server authentication protocol is the simplest and the most acceptable to users. Here, we propose a two-server architecture which uses an encryption algorithm and no decryption algorithm, which is called the one-way technique. It is a comparatively less complex and more secure scheme than the other schemes which have been proposed previously.

2. Background And Related Works:

In the recent years, a lot of importance has been given to password authentication schemes which can provide security against threats like offline dictionary attacks. Many multi-server architectural schemes have been proposed to provide maximum security to the users of the organization. Two-server architectural schemes are the simplest and most accepted by the users. Some of the systems that are based on two-server architecture are discussed below. Mukesh et al [1] proposed a robust finger print based two-server authentication and key exchange system which is the first biometric two server authentication scheme. In this scheme, the user's password is replaced by the random string generated by fingerprint template, the user need not memorize. Brainard et al.'s [2] proposed a two-server password system in which one server (called Blue Server or Service Server, SS) exposes itself to users and the other (called Red Server or Control Server, CS) is hidden from the public. It is not a password-only system. Both the servers need to have public keys to protect the communication channel from users to servers. This setting makes it difficult to fully enjoy the benefits of a password system.

Katz et al's [3] proposed a system which is the first provably-secure two-server protocol for the important password only setting. The user needs to remember only a password, and not the server's public keys. It is the first two-server protocol (in any setting) with a proof of security in the standard model. The two server setting facilitates a balance between robustness and protocol complexities. Yang et al. [4] proposed a practical two server architecture and an efficient password-only two-server authenticated key exchange system. This scheme is a password-only variant of the one introduced by Brainard et al.'s [2]. Yanjiang Yang [5] proposed a new scheme which enables a user to use the same password over multiple service servers, which is an important feature of the two-server model. Dexin Yang , Bo Yang [6] proposed a scheme which includes three phases such as registration, authentication and key exchange. It proposes a

primitive technique called building block. This technique can be used to defend against adversary's off-line attack while carrying out key exchange.

3. Password Based Two Server Authentication System

In this paper, we are proposing to use an architecture with two-servers for the password authentication. Both the servers are equally important as they have equal part in the encryption of the password.
The overall architecture of our system is shown below in Fig. 1.

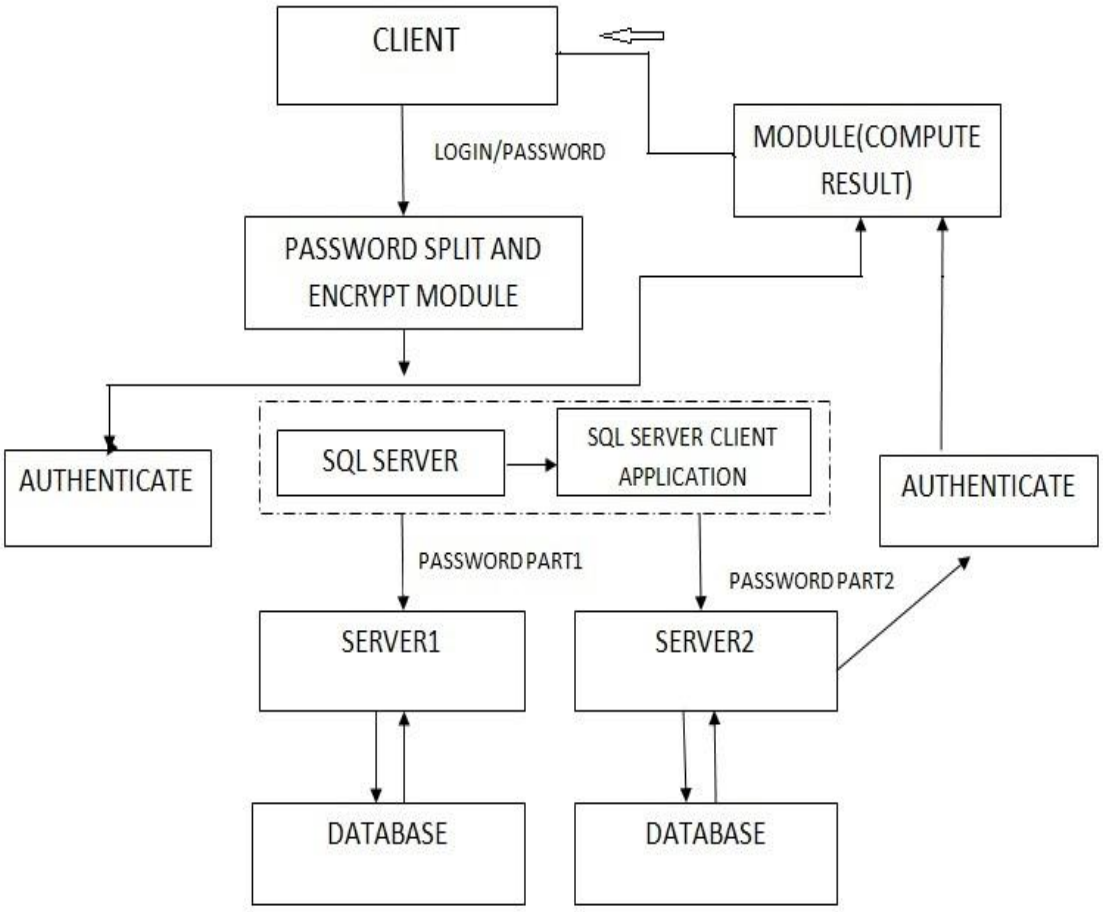


Fig 1: System Architecture

SIGN UP:

When the user signs up for an application, he/she enters the asked details in the form including the username and the password. The entered password has to be between 8 and 13 characters in length as the splitting function we have used is a modulo 7 function. The constraints for the username include that each user must have a unique user id. So a username once registered cannot be used again by another user. Once the details are submitted, the password is sent to the next module (next module) to start the encryption process.

$$8 < \text{length(password)} < 13$$

PASSWORD SPLIT AND ENCRYPTION MODULE:

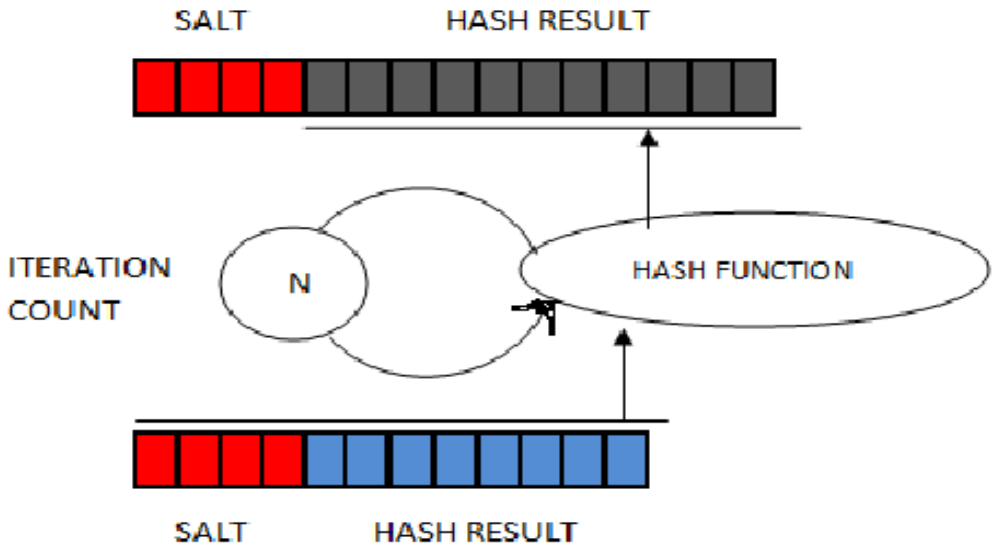
The password entered by user will be divided into two parts. Let us assume the password to be U and the two parts obtained after password division to be U` and U``.
Then, U`=U%7 and U``=U-U`.
Now, both U` and U`` will undergo the following process:

- Character string to byte sequence translation

Here, U` and U`` will undergo encoding process using UTF-8 encoding. Let us consider the encoded ones to be U1 and U2.

- Concatenating U1 and U2 with fixed salt

The salt is a sequence of bytes that is added to the password before being digested. This makes our digests different to what they would be if we encrypted the password alone, and as a result protects us against dictionary attacks.
Fixed salt is a sequence of bytes that we will use for digesting every password. We can keep this salt hidden and consider it an added security value, but it can make our system more vulnerable to birthday attacks and, in general, attacks driven against our database of passwords as a whole.
Let us consider the concatenated U1 and U2 to be U1` and U2``.
NOTE: The fixed salt that we use here is 8 bytes in length.



The iteration count refers to the number of times that the hash function with which we are digesting is applied to its own results. This means that, once we have selected a salt and concatenated the password to it, we will apply the hash function (MD5 algorithm), get the result, and then pass it again as input to the same hash function, then do the same several number of times.

The difference between applying the hash function once or many number of times is that, it won't be a problem for us. It might take only a couple hundreds of milliseconds... but attackers would have to generate an enormous amount of tentative password digests when brute-forcing. And, for an attacker, the difference between applying the hash function once and applying it a thousand times for each try would be a real computational problem.

CONCATENATING THE HASH RESULT WITH VARIABLE SALT:

Here, we use a variable salt, which is usually a safer option (better if it is random). This is generated or computed separately for each password being digested and it allows each stored password to be decoupled from the others, creating a stronger overall protection and highly improving safety against attacks driven on our database of passwords as a whole.

Let us consider the final result to be U1* and U2*.

NOTE: The variable salt we use here is 8 bytes in length.

SQL SERVER:

The front end of the application is connected to the back end using PHP connections. It helps us in storing U1* and U2* into their respective server's database.

AUTHENTICATION MODULE:

This module authenticates U1* and U2* with the passwords stored in the server's database.

RESULT COMPUTATION MODULE:

The results from the both the authentication modules will undergo AND operation to give final authentication result.

4. Results And Discussions

SECURITY AGAINST ATTACKS:

If an intruder wants to crack the password, then he is going to try all the possibilities but he will not succeed in his attempt, because the fixed salt is a sequence of bytes which is added to the password before being digested and as a result protects us against dictionary attacks. Because the hash function is applied a large number of times, the attacker would have to generate an enormous amount of tentative password digests when brute-forcing. And, for an attacker, the difference between applying the hash function once and applying it a thousand times for each try would be a real computational problem.

STRENGTHENING CONDITIONS:

In this scheme, we use a variable salt, which is usually a safer option (better if it is random). This is generated or computed separately for each password being digested and it allows each stored password to be decoupled from the others, creating a stronger overall protection and highly improving safety against attacks driven on our database of passwords as a whole.

5. Conclusion And Future Works:

In this paper, we have enhanced the security of user password authentication by using two-server architecture. In the previously existing projects the concept of a two-server architecture and two way techniques have been used. But the major drawback of this technique is that the hacker could decrypt the critical data, thus leading to loss of security.

In order to overcome this problem, we have combined the concepts of two server architecture and a hash function.

The hash function used here is MD5 (message digest) algorithm. This combination results in a one-way technique (which cannot be decrypted), thus avoiding brute-force attacks. The advantage of using a one-way technique is that even the

organisation will not know the actual password of the user, thus enhancing the security. Since two-server architecture is used, it helps in preventing offline dictionary attacks.

References:

1. Mukesh et al,"A robust fingerprint based two server authentication and key exchange system", 3rd International Conference on Communication Systems Software and Middleware and Workshops, 2008
2. Brainard et al,"A new two-server approach for authentication with short secrets", in Proceedings of the 12th USENIX Workshop on Security, pages 1-2.IEEE Computer Society 2003.
3. Katz et al," Two-server password-only authenticated key exchange",2004.
4. Yang et al. "A Novel Two-Server Password Authentication Scheme with Provable Security", IEEE Transaction 2010 10th IEEE International Conference on Computer and Information Technology (CIT)2010
5. Yanjiang Yang,"Enabling Use of Single Password over Multiple Servers in Two-Server Model ", Computer and Information Technology. (CIT), 2010 IEEE 10th International Conference.
6. Dexin Yang , Bo Yang,"A Novel Two-Server Password Authentication Scheme with Provable Security", IEEE Transaction 2010 10th IEEE International Conference on Computer and Information Technology (CIT 2010)"

