Web Application for Secured Two Factor Authentication

Ravi Anand[1], Shaz Ahmad[2], Shubham Jaiswal[3], Shashank Gowda M K[4]

[1,2,3,4] Student, Dept. of Information Science and Engineering, the National Institute of Engineering, Mysuru, Karnataka, India

---

Abstract *- We are surrounded in an era where we are indulge in a lot of online banking transaction. As far as money is considered, we want to be sure about the security provided by banks. User authentication thus becomes a very crucial part in the security domain. Till now the smart-card based authentication schemes are becoming obsolete. To overcome this issue, we propose a security model for user authentication which will capture the capabilities of an adversary and will safeguard the data and money of authentic users.*

Index Terms **- Two-factor authentication, Three-tier Architecture, M-Pin, Hashed Transactional Password.**

I. INTRODUCTION

The paper lies under the domain of security and entity authentication. In this modern era there has been an exponential growth in the amount of online transactions. As entity or user authentication becomes a crucial part, we are proposing a security model for safeguarding user's data from adversarial attacks.

Our proposed system is based upon three-tier architecture. Three –tier is an architectural deployment style which describes the separation of functionality into layers with each segment being a tier. They evolved through the component-oriented approach, generally using platform specific method for communication instead of a message-based approach.

The architecture has different usages with different applications. It can be used in web applications. With the help of this architecture the software is divided into three different tiers which are: Presentation layer, Business layer and Data access layer. Each layer is independent of each other. This helps in reducing the overall load on to the system.

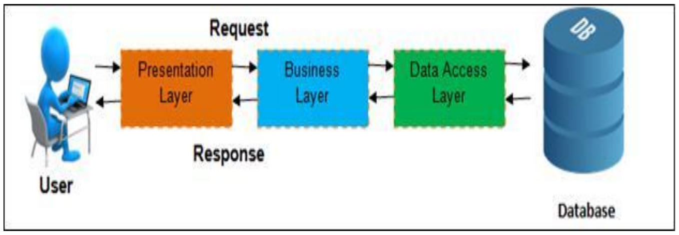

Figure 1: Three-tier architecture

Presentation tier: Occupies the top-most level of the architecture and displays the services available on a website for a user. This layer accumulates data from the user through a front end interface.

Business tier: Also known as Logic tier, it controls application functionality by performing detailed processing. All the logical and algorithmic computations are taken care in this tier.

Data access tier: It houses database servers where information is stored and retrieved. This layer is the actual database layer.

The main benefits of three-tier architecture are:

- As each tier is independent of other tier, any changes can be carried out without affecting the system.
- As tier is based on deployment of layers, scaling out system is easy.
- Independent architecture of the system improves flexibility.

In our proposed system there are two actors: User and Hacker. An authentic user will register in our system and will his bank credentials for doing online transactions. Hacker is presumed to have hacked the system and can use the users credentials to act as an imposter for doing any transaction from an authentic user's account.

But because of our two factor authentication security model, hacker will not be able to do any unauthorized transaction and as a result, account of the user will be blocked as a preventive measure and along with it, an email will be sent to the user's mail id regarding the unauthorized access by the hacker and will be asked to change their credentials.

II. LITERATURE SURVEY

Many works have been done in the field of user authentication and security mechanism in recent times.

Ding Wang *et al.* [1] took a one step ahead in two factor authentication scheme. Their new scheme is simpler and has strong notions of security. They proposed an adversary model which will create a new benchmark for processing of current user authentication schemes. Their new scheme is built to resolve various issues arising from the area of a user's security. They emphasized more on to protect user's data and privacy from malicious attackers.

SK Hafizul Islam [2] in his paper examined the security loopholes of the smart card based remote user password authentication scheme. To ensure efficient and robust online transaction, security of authentication protocol turns out to be a great concern nowadays. This paper constructs a more advanced version of authentication scheme which has more security features, more functionality and low cost thereby preserving and protecting user's privacy and important data's.

We try to implement a model which is better than a smart card authentication security model and provides two folds of entity authentication mechanism. We have emphasized more on protecting user's data and are blocking the hacker before he/she can do any fatal harm. Till now, the security models are only able to know that an user's account has been hacked only after an unauthorized transaction is carried out and the authentic user has claimed about this. In our system we want to save the user's account from any such unauthorized transaction.

III. SYSTEM DESIGN

This paper presents the idea to deal with entity authentication of an authorized user. There are two secure factors used for authentication checking. First is m-pin and second one is a hashed transactional password, which will be used for making every transaction.

In our system, the user first has to register itself. In the registration process the user need to give various information such as name, phone no., email address and the password that they are going to use in every subsequent login.

After successful registration process every user will get an email with a m-Pin. This m-Pin is the 1st factor that we have mentioned in our system. As soon as the user will login for the first time he/she will have to enter the m-Pin for self authentication.

After going through the first authentication, user will be asked to enter the various bank credentials for doing any transactions such as card no., account no., bank name which will be used for doing any transaction.

After registering all the bank details, a transactional password will be generated and will be sent to the user's registered mail id, which will be used for doing each transaction. This generated transactional password is stored in our database after going through a hash and salt process.

When user doesn't need to do any transaction, he/she can log out of the system at any point of time

A use case diagram is a type of behavioral diagram created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The use case diagram of any authentic user is represented in Figure 2.

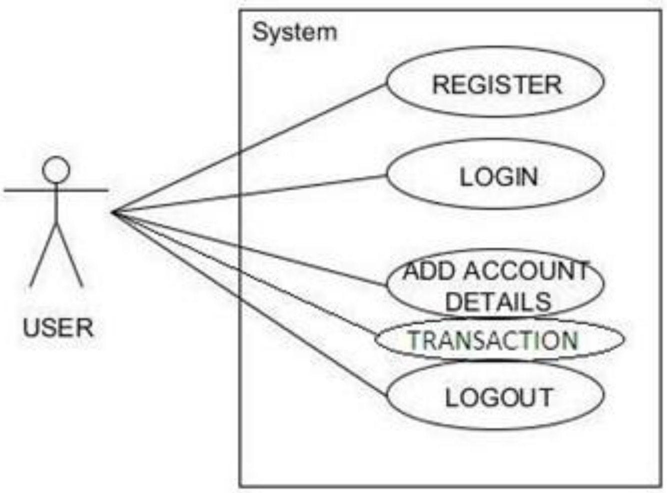

Figure 2: Use Case Diagram of an Authentic User

The second actor in our system is the hacker, we are presuming that the hacker has already hacked the system and thus, has the access to the stored data credentials about the user, such as login id, password and bank details.

But even if the hacker has got the access to the stored data, he will be barred from doing any transaction because for doing any
transaction, hacker will need to have the transactional password which is our 2nd secure factor.

When an user wants to do a transaction only then a transactional password is generated, which will be sent to the user's registered mail id but this same password is not saved in our system.

This generated transactional password is first salted (attached with a random string) and is hashed, after salting and hashing the resulted ciphered password is stored in our database. So even if the hacker has the access to the database, he/she can't use the stored transactional password as it is. He/she needs to decipher it to get the actual password.

What is hashing? Hashing is the conversion of a string of characters into completely different strings which is usually of fixed length value. A hash algorithm is a function which converts a data string into a fixed length of numeric string output. It is a function used to map data of any arbitrary length to a fixed size string. But it is difficult to reconstruct the same hash value if the input data is not known. This function is generally used for checking data integrity. Hash functions are generally perceived as a similar concept to: Checksums, Check digits, Lossy compression but each has its own uses. Although this hash function can be considered to overlap these concepts.

The size of input data is very important in any hashing function. If the input data is small the logarithmic attacks or the dictionary attack becomes an option for intruders for hacking the hash values. For increasing the safety of our hashed passwords saved in the database, we are attaching the actual password with salts (random string of characters).

Firstly, the generated transactional password will go through salting, which will increase the size of our input data for hashing. After the salting the increased size password will be converted into final hash values which will be stored in the database.

In the proposed system, if a user wants to do a transaction a transactional password will be sent to his registered mail id, since the hacker has the access to the database he/she can easily get the transactional password, even if it is hashed, it is of no use if we allow a hacker to take months of time to hack it. As a restriction to this scenario, we have come up with a solution which is to time every transaction and also the validity of each transactional password. If we allow a window of time (say 5 minutes) for the validity of a transactional password, it becomes nearly impossible for the hacker to retrieve the original password from the hashed value.

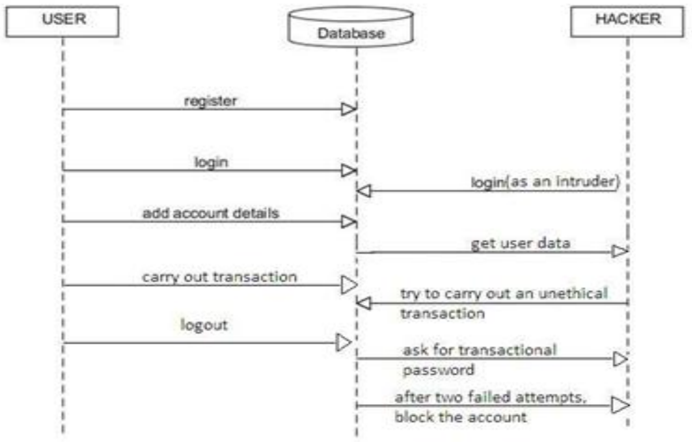
USER
Database
HACKER
register
login
login(as an intruder)
add account details
get user data
carry out transaction
try to carry out an unethical transaction
logout
ask for transactional password
after two failed attempts, block the account

Figure 3 Sequence diagram of proposed system

IV. FUTURE ENHANCEMENT

The above proposed system depicts an application level entity authentication mechanism. But in near future two factor authentications would not be enough and three factor authentication is already on its way. IP tracing can also be included two catch the intruder after two unsuccessful attempts in the proceeding of any transaction.

V. CONCLUSION

In the proposed system we have developed a web application prototype in which we will host several user accounts and critical information of the account will be hacked by the intruder, but as soon as the intruder tries to decrypt the transactional password the user account will be blocked as a preventive measure and proper notifications will be send so that no unauthorized transactions can be made.

ACKNOWLEDGEMENT

The authors would like to acknowledge lecturers, Ms. P Devaki Dept. of Information Science and Engineering, The National Institute of Engineering, Mysuru, India and Mr. Suhaas K P Dept. of Information Science and Engineering, The National Institute of Engineering, Mysuru, India for their valuable support in the study.

REFERENCES

1. Ding Wang; Ping Wang “Two Birds with One Stone: Two Factor Authentication with Security Beyond Conventional Bound”. 2016 IEEE Transactions on Dependable and Secure Computing

2. SK Hafizul Islam; “Design and Analysis of an improved Smart Card based remote user password authentication scheme” 2014, International Journal of Communication Systems

3. Chu-Hsing Lin; Yi-Shiung Yeh; Shih-Pei Chien; Chen-Yu Lee; Hung-Sheng Chien; “Generalized secure hash algorithm: SHA-X”. 2011 EUROCON –International Conference on Computer as a Tool