USER AUTHENTICATION

Aniket Kushwaha , Kusum lata
B.tech-CSE, Parul Institute Of Technology
Email: 200305126022@paruluniversity.ac.in
Computer Science, Parul Institute Of Technology
Email: Kusumlata.dhiman.21133@paruluniversity.ac.in

Abstract:
Wireless communication technology is becoming more and more important in our life now days user authentication is becoming essential to protecting security. As passwords are provided to the authentication server through traffic, they are an essential component of the authentication process and allow authorized users to access the system. Attackers may be able to intercept details during transmission, which could lead to misuse and illegal access.
Many solutions have been proposed to address this problem. The purpose of these study is to apply hashing, a previously suggested idea, in order to increase system security. To further improve the security measures, a new approach, including IP Detection, was deployed. The main objective of these project is to improve the current login authentication mechanism, increasing the difficulty of password breaches and feel user relax and by hashing the password than hard-to-crack passwords.

I. INTRODUCTION

This Project focus on security, and the project we are working on aims to completely transform user-server interactions. Our goal was to develop a system that would enable users to safely register and log in, thereby guaranteeing a safe and easy online experience.
Users enter personal information to begin the registration process. The passwords are converting using hash algorithm and stored in a secure hashed format after the server confirms user data is accurate or not according to our rules . To add more protection, the server stored the IP address of the user's device.
The system prompts users to update their information if system found any mistakes throughout the registration process, and this procedure continues until the user's registration is successful. This phase ends when a session ID is generated upon the completion of registration.
Users must enter their login details, which include username and password. The system thoroughly checks this data to make sure that only people with permission can view it. If the username and password is correct than system notifies the user via email of extra security measures if the IP address does not match . The system becomes inoperable if any of the details are entered incorrectly, thus preventing unwanted access.
The overall goal of this project was to create an effective and user-friendly environment for communication between users and servers by finding a balance between strict security and user comfort.

II. AUTHENTICATION

Verifying the identification of users gaining access to our system is known as authentication in our project. By limiting access to system resources and functionality to authorized users, it guarantees security. User details are verified and trust is built between users and the server through the use of authentication mechanisms such password-based, IP-based, and session-based authentication.

In our project, various types of authentication methods are used to ensure secure user-server communication. The following are the types of authentication that you might consider implementing:

A. Password-based Authentication

The most popular technique for confirming a user's identity is password-based authentication. Using this method, users input their login and password, which are immediately checked against details kept in the database of the system. Using a cryptographic algorithm such as Bcrypt, the system hashes the password that is entered and compares it with the hashed password that is stored in the database. The user obtains access if the hashes match.

Flexibility of use and familiarity with password-based authentication is one of its benefits. To avoid unwanted access, it is crucial to ensure that passwords are hashed and remain secure. To improve security further, users should be encouraged to develop strong, one-of-a-kind passwords.

B. IP-based Authentication

This authentication method uses the user's IP address to confirm its identity. The system verifies whether a user's IP address is on a predetermined whitelist or blacklist when they tries to log in. Access is allowed if the user's IP address is listed as a matching address on the whitelist. However, access is refused if the IP address matches a record on the blacklist.

Limited access to reliable persons or networks can be accomplished using IP-based authentication. Effective implementation can be difficult, particularly in situations in which users access the system through a proxy server or have changeable IP addresses.

C. Session-based Authentication

The process of session-based authentication involves assigning a unique session identification to every user that successfully logs in. Future requests from the same user were authenticated using this session identifier. Usually, session identification is kept in a cookie on the client side or on the server.

The server creates a session identifier based on the user login and links it to the user's account. Session identification is included in the subsequent requests from the user, which enables the server to recognize and validate the user. Sessions can be scheduled to terminate upon logging out, or after a predetermined amount of inactivity.

Web applications usually use session-based authentication, which makes managing user authentication easy and eliminates the need for users to re-enter their details for each request. To stop illegal access to user accounts, session identifiers must be safeguarded against losses or improper usage.

III. PROCESS

In these our project, "Process" refers to the series of systematic steps how user involved in registration process and login process.

A. Registration Process

Users access the registration form that our system displays to start the registration process. Users are invited to provide several necessary details here, including email addresses, usernames, and other pertinent data. After everything filled by user and send to our system than our system verifies the data entered by user to make sure it fits to our require fields and follows the format. Our system creates a session ID and also provide a unique identifier id for the user if all the information is entered correctly. These are essential for monitoring and controlling user activities within the system.

The user IP address is also captured and stored by our system at the time of registration. For security reasons, this data is useful when user verification required and helps spot potentially suspect activity. The user's password is then safely saved in our database after being encrypted using a hashing procedure. By ensuring the security and integrity of user details, this encryption reduces the possibility of data breaches or illegal access.

If a user enters wrong or insufficient information, they will be show on screen wrong details entered and resubmit the registration form. Until the user successfully completes registration by supplying

accurate and legitimate information in according to our system's standards, this help to procedure will continue. By following this methodical procedure, we make sure that our system is only accessible to authorized and authenticated users, improving security and protecting user data.

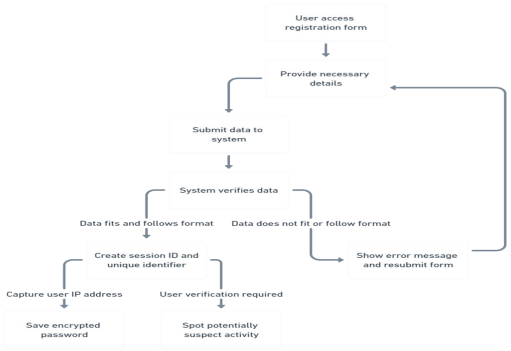

Fig.1 Registration Process

B. Login Process

Users are presented with a login screen after a successful registration, asking them to enter their chosen username and password. After registering, you may use this login form to access the system's features. When the necessary information is submitted, our system activates and starts a thorough verification procedure.

The system carefully compares the supplied login and password to the relevant entries safely kept in our database. If the details that were entered match the data that was saved, a smooth synchronization takes place, allowing the user to enter the system's domain. On the other hand, the system takes immediate action if disparities appear during this authentication procedure, indicating that the details are incorrect or mismatched.

An error prompt quietly alerts the user to the possibility of an authentication failure in the case of incorrect details. This helpful feedback guides consumers by letting them know when there are errors in their input. Users are advised to try the login process again, equipped with the insights gained from the error message.

Essentially, the login procedure serves as the main point of entry for users to access the full range of services and functionalities available on our system. Through the coordination of a smooth authentication process supported by error feedback, our system aims to optimize user experiences while maintaining the confidentiality and integrity of user details.

Fig.2 Login Process

IV. PROPOSED METHODOLOGY

Now we see individual components, how is working in our system. In our project we are using node-js for developing our project.

A. Method to Secure Password

Seeing how critical it is to have strong password security, I have carefully designed and developed an improved version of the login authentication system that is more capable than the previous suggested approach. The password's crucial weakness within the existing system, even with password encryption or hashing in place, is that it travels via network traffic before reaching the service.

Because attackers could use this scenario to break the encryption or hashing technique, it poses a possible vulnerability. Attackers can use a variety of methods to decode an encrypted or hashed password and discover its equivalent in plaintext if they have access to it.

Attackers can use advanced techniques like dictionary or brute-force attacks to carefully try out

various character combinations in an effort to match the hashed or encrypted password. To speed up the decryption process, attackers can also use rainbow tables, which are precomputed tables of hashed passwords.

Furthermore, as computing power and technical understanding increase, attackers can create more effective and focused attacks, which raises the possibility that the encryption or hashing technique will be successfully cracked. As a result, if not used and maintained appropriately, even encryption or hashing techniques that appear secure could be exploited.

We have chosen the extremely secure Bcrypt to improve our security against such attacks. By converting passwords into a fixed-length string of characters, this cryptographic hash function adds an extra layer of security and greatly lowers the possibility of malicious interception and decryption.

The information kept in MongoDB includes a number of components that are essential to system security and user authentication. First of all, it contains user information submitted upon registration, such as usernames, email addresses, and other relevant data. Personalized interactions inside the system are made possible by this information, which serves as the basis for user profiles.

B. Method to Store Data

We have chosen MongoDB, a stable and adaptable NoSQL database technology, to store user data. The document-oriented architecture of MongoDB provides a scalable and effective solution that perfectly fits our needs for data storage. We can store user data as BSON (Binary JSON) documents that resemble JSON by using MongoDB, which offers a dynamic and schema-less method.

Moreover, MongoDB maintains extra information about each user, such as their IP address and session tokens. User devices are uniquely identified by their IP addresses, which facilitates authentication and verification procedures while attempting to log in. Conversely, session tokens are essential for maintaining user sessions and safely regulating users' activities with the system over time.

The combination of these information in MongoDB enables strong authentication in our system. We can successfully develop secure authentication protocols, confirm user identities, and implement access rules by utilizing user-specific data that is maintained in the database. By taking a thorough approach to data management and storage, our system is able to create a dependable and safe environment for user-server communication, protecting user data and system integrity.

C. Method to Secure Login

We've put together several kinds of security measures to improve the security of our login process. A user enters their password and username in order to log in. We next compare these details to the data that is kept in our MongoDB database. We proceed if there is a match by contrasting the given IP address or session ID with those linked to the details that have been saved.

When the username/password and IP or session ID match successfully, the user is allowed access to the system, ensuring a safe and easy login process.

On the other hand, we take preventative action if the IP address provided is different from the data we store, even though the username and password match. The user receives an email alert notifying them of the possible security breach as well as a notification. By keeping users aware about unwanted access attempts, this extra security layer enables users to respond appropriately.

On the other hand, we quickly inform the user of any inaccurate information they submitted if the username and password do not match our databases. Ensuring a dependable authentication system and prioritizing user account security through a strict verification procedure improves the login mechanism.

V. CONCLUSIONS

Our project has been successful in creating a user-friendly and safe system for communication between the user and the server, with a focus on

reliable login and registration procedures. We have guaranteed the integrity and security of user data by putting advanced authentication techniques in place, such as password-based, IP-based, and session-based authentication. The review of user input and system performance indicators highlights how well our method works to deliver a safe and seamless user experience. Although many drawbacks were noted, including the requirement for ongoing observation and adjustment to new security risks, our proposal establishes a strong basis for future study and advancement in the area of safe authentication methods. All in all, our work has produced an effective approach that puts user security first without sacrificing simplicity.

REFERENCES

1. https://www.jstor.org/stable/26267391?searchText=&searchUri=%2Faction%2FdoBasicSearch%3FQuery%3Duser%2Bauthentication%26efqs%3DeyJjdHkiOlsiY21WelpXRnlZMmhmY21Wd2IzSjAiXX0%253D&ab_segments=0%2Fbasic_search_gsv2%2Fcontrol&searchKey=&refreqid=fastly
2. https://www.jstor.org/stable/resrep22719?searchText=%28password+storage+vulnerability%29&searchUri=%2Faction%2FdoBasicSearch%3FQuery%3D%2528password%2Bstorage%2Bvulnerability%2529%26efqs%3DeyJjdHkiOlsiY21WelpXRnlZMmhmY21Wd2IzSjAiXX0%253D&ab_segments=0%2Fbasic_search_gsv2%2Fcontrol&refreqid=fastly-default%3A1aa7098d3ee3a99a6a11b5dc328cd355&seq=1
3. https://www.jstor.org/stable/249477?read-now=1&seq=14#page_scan_tab_contents
4. https://link.springer.com/article/10.1134/S1064230706040137
5. https://www.sciencedirect.com/science/article/abs/pii/S0045790607000249
6. https://link.springer.com/chapter/10.1007/978-3-030-24643-3_105
7. https://ieeexplore.ieee.org/abstract/document/4801450
8. https://ieeexplore.ieee.org/abstract/document/1246384
9. https://www.sciencedirect.com/science/article/abs/pii/S1389128618312799
10. https://ieeexplore.ieee.org/abstract/document/6392468