Web based Graphical Password Authentication System

[1]Mrs. G. Sandhya(Assistant professor),
[2]AJAY BANDARI, [3]HARSHITH KYATHAM, [4]PRAVALIKA NIMMAKAYALA, [5]VAISHNAVI NAINI,
Department: CSE(IOT),
MALLA REDDY INSTITUTE OF TECHNOLOGY AND SCIENCE, Telangana, Hyderabad.

Abstract—

The majority of security and protection software use authentication that relies on passwords. However, "the most fragile connection" in the authentication chain is considered to be human acts, such as selecting weak passwords or supplying passwords in square proportions. Customers will likely choose brief or meaningful passwords that are easy to remember rather than random sequences of letters and numbers. People may use these applications anytime, anywhere with a variety of devices thanks to web apps and mobile app compilations. There is a higher chance of exposing passwords to bear riding assaults, however this innovation also delivers great accommodation. Intruders may observe clients directly or use external recording devices to get client credentials. We need an alternate confirmation mechanism to circumvent this kind of problem. We have the option to choose a graphical authentication technique here. A picture password is the most convenient way to log in as it is easier than trying to remember and type in a lengthy string of simple passwords. You may log in by touching certain areas or making certain movements over a pre-selected picture.

Key words: Graphical Passwords, Image Slicing, and Encryption • Authentication.

I. INTRODUCTION

User authentication is a process that devices use to verify the identity of people attempting to access their network resources. The majority of websites and apps still utilize text-based passwords. A string of characters, including spaces, alphabetic characters, numerals, and special characters, makes up a textual password. Most services allow customers to use the same login credentials for many accounts. However, their security is lacking. Strong passwords that include both capital and lowercase letters, numbers, and symbols should be maintained. After that point, these text passwords should be able to withstand brute force assaults. But it's not easy to remember and recite a robust text password. As a result, customers will often use shorter passwords or ones derived from words rather than random sequences of letters and numbers. The weakest link in the authentication chain is human error, which includes choosing weak passwords for new accounts and entering incorrect passwords in an unsafe method for subsequent logins. When someone sneaks a peek behind your back as you type in sensitive information like a credit card number, password, or ATM PIN, they are engaging in shoulder surfing. It is hard to remember and recall a strong textual password. To circumvent these issues, we provide a safe graphical web-based authentication method that safeguards users from being targeted by shoulder surfing assaults.

II. LITERATURE REVIEW

A technique called "Combined PWD" was suggested by Wantong Zheng and Chunfu Jia. By including separators (such as spaces) into passwords, this technique suggests an online component for secret phrase verification called combined PWD. This would strengthen the existing framework for secret word validation. This strategy takes the client's feedback into account. During this test, users may insert spaces into their secret word up to the point when they register a record, and the site's back-end will keep track of how many spaces are in each hole [1]. We found that the framework described in the paper [2]—a new kind of time-based unique password—could help users sidestep problems associated with using third parties, such as one-time password emails, tests, and token devices—by allowing them to specify an underlying secret word that would describe the evolution of the secret key over a specified period of time. After then, it was discovered that the system keeps the dynamic

password's strength and makes the system more usable in terms of availability [2].

Yang Jingbooo put out a robust password authentication technique. There are two main categories of one-time password authentication schemes: weak password authentication and strong-password. systems for verification. Within the scope of this article, we examine W.C. Ku's strategy and demonstrate an assault on his procedure. I have discovered that passwords that are both strong and difficult to guess have a greater strength. We will later provide a robust password authentication method. In order for the modification convention to counter the Stolen-verifier attack, this work develops W. C. Ku's original proposal. Without sacrificing efficacy, the revised convention is constructed [3]. Hua Wang and Yao Guo provide Desktop Password Authentication Center (DPAC), an alternative reuse-positioned secret phase authentication system, to reduce the cost of securing passwords from threats by reusing countermeasures across apps. With this setup, a lot of boring tasks can be eliminated, and costs can be cut down. Essentially, we show that DPAC may work by creating a prototype that uses two sample countermeasures and migrates the popular OpenSSH to it [4].

In several applications, such websites and database systems, etc., the password authentication code (PAC) is a crucial concern. A PAC-RMPN method is suggested by Salah Refish. This article introduces PAC as a means for two clients to confirm each other's authenticity. Password authentication at the incoming level has been a long-standing issue, but this study offers a new alternative. In order to protect this secret word from potential assailants, they need to come up with a plan. In order to authenticate another user, an authorized user just enters in their password and pushes enter [5].

An enhanced level of security is achieved via the proposal of a password authentication mechanism. Pattern, key, and fake numbers are all used in this approach. In order to do this, the client must understand and use design as network area numbers, record key attributes that relate value to secret password, and attach imposter attributes to trick the attacker. The next time the customer wants to log in, they'll need to follow the example and create a secret word using fake digits, based on the secret key from the design with the enrolled key attributes. The great difficulty of guessing passwords in multi-levels—first from the pattern, then from the key, and finally from dummy values—minimizes shoulder surfing, brute-force assaults, cross-site scripting, etc. [6]. Although the secret key is essential for approval, programmers often find success in cracking secret phrases using the weak secret key that the customer chooses. The suggested architecture uses Honey encryption in conjunction with the Honeyword process to fortify the secret key storage. To entice an attacker, you may use honeywords, which are fake passwords stored with a unique secret word. Honeyword is based on the notion of using fake passwords. These are meant to entice the aggressor. Although there are a variety of methods available, such as the Chaffing-with-tweaking and Chaffing-with password models, etc., for generating the Honeyword of the original password, the current methodology is flawed [7].

III. PROPOSED SYSTEM

In this section, we build a web app that employs graphical authentication. It employs a dual-layer security system. In this case, the second layer of security is provided by a picture password. Strong passwords made of text are therefore unnecessary. Any simple textual password will do for users. Three distinct modules make up the system. As shown in Figure 1, the Graphical Password System Modules Section A: Open-To-the-Public In other words, it's the last destination for any given website's viewers. This section is accessible to everyone who has the URL. Although anybody may see it, they are unable to edit or modify the data. Section B: The Human Interface A component of the user module are the users who have registered. Two features make up the user module: login and registration. The user's name, cellphone number, email address, and basic password information (both textual and graphical) are collected by the system throughout the registration process. All of these are securely kept in the database using encryption. In order to get access to the resource, the user must provide their username, a text password, and

an image password during the login step. It checks the entered values against the information the user provided on registration. If there was a match, then the user will be able to access the page. C. Profile and Preferences The third module houses the client's data as well as other settings for the electronic web platform. When a user registers, their account is automatically established in the database via a connection between the user and account modules. Users also have the option to change their passwords whenever they choose. Benefits include information about logins, settings for privacy and security, and similar features. Clients may also use this section to get warnings and submit help requests.

IV. SYSTEM ARCHITECTURE

How the framework should function is decided by the architecture. Web application design is defined by factors such as request response speed, page loading time, ability to handle varied requests, and more. Consequently, using the optimum design is essential for improved execution. In this case, the Model-View-Control Architecture (MVC) is used. One example of a programming project's architectural plan is the Model-View-Controller (MVC) architecture.

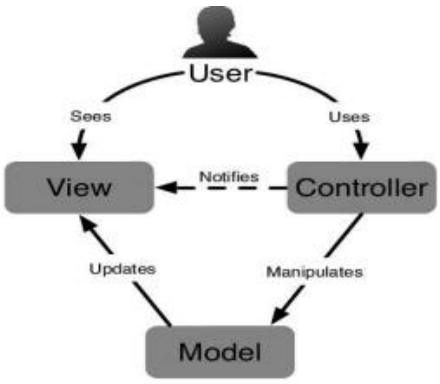

Fig. 2. MVC architecture

Model, View, and Controller are the three components that make up the design (Fig 3). The structure becomes more adaptable with these parts. The Model layer handles information and data set relationships at its main level. In a model-view-controller (MVC) architecture, the layer that displays the results is called the view layer. The Controller selects the data flow and acts as a go-between for the model and view components. It works like this: the client provides data, which is then cycled through the Model segments before being passed on to the View segment.

System Architecture

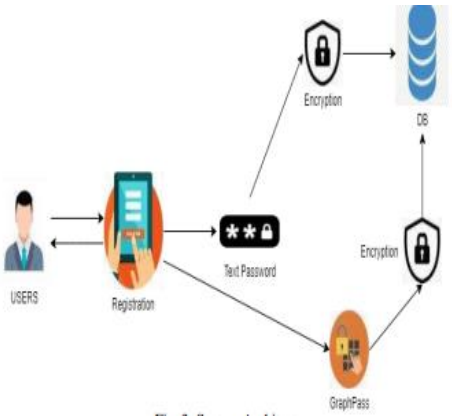

Fig. 3. System Architecture

At the client's boundary, the user initiates the registration process. Both of these encryptions are part of the registration procedure. Separate ones are provided for graphical passwords and text passwords. Graph Pass was cut into four sections. Each slice is responsible for encryption. Graphical user interfaces that are easy on the eyes simplify the process. So, the customer is spared the mental effort of researching and understanding programming languages and concepts. The framework is an ardent adherent of the Model-View-Controller (MVC) architectural principles. One example of a programming project's architectural plan is the Model-View-Controller (MVC) architecture. Not to mention the need for a more robust database capable of holding an enormous amount of data; in this case, we use the SQL worker to store all of the client information.

This program runs on the web and uses a client-server model. On the client side, many devices will be linked to the server over the internet or cloud computing. In

response to a request from a client, the server provides the relevant data. In the client-server paradigm of computing, the worker stores, transfers, and manages most of the resources and services that the consumer consumes. At least one client PC is linked to a server in this design via an organization's or web's affiliation. The framework's computational resources are shared. Every request and service is sent via an organization in a client/server architecture, which goes by many names such as systems administration processing model or customer/worker network. B. Architecture of the Framework Our client side makes use of CodeIgniter, a PHP framework. It follows the Model View Control (MVC) design pattern. The model session is responsible for managing database activities. The model session is where things like validations and comparisons to databases happen. The control session is where the macro functions are executed.

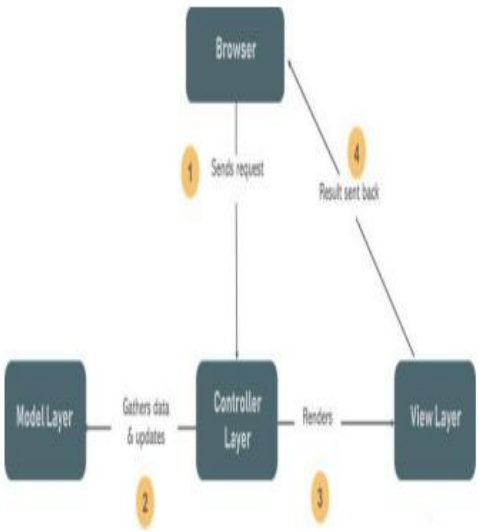

Fig. 4. Framework Architecture

Tools for software Sublime Text was the text editor used for this work. Shareware and accessible across platforms, Sublime Text is an API-enabled source code editor that uses Python. Users may connect functionalities via plugins, which are usually community-built and maintained under free-software licenses, and it supports several programming and markup languages natively. We use XAMPP to set up the server. Apache Friends' XAMPP is an open-source, free, and cross-platform web server solution stack that includes MariaDB, an HTTP server, and interpreters for Perl and PHP scripts. Moving from a local test server to a live server is made easy by most genuine web server installations using the same components as XAMPP. Database collaboration is made possible with the use of Structured Query Language, or SQL. Database storage, manipulation, and retrieval are all possible uses for it. Section B: Appliances Computer specifications for this build should include an i3+ CPU, 4GB+ RAM, and 2GB+ SSD space.

V. RESULT

Fig. 5. Home Page of Graphical Password Authenticator

You can see the graphical password authenticator in action in the picture up there. On this main page, users have the option to sign up for an account and access their profile. Two levels of protection are in place to protect the Registered user's information. You may choose between a written and a visual password. Simply clicking the login button will allow the user to access their profile. As indicated before, the login page also incorporates two levels of protection.

VI. CONCLUSION

Every time a user attempts to access their account or data, authentication is necessary to safeguard their digital property. There is a risk of shoulder surfing assaults if the authentication procedure is carried out in public. When people log in using PINs or standard text passwords, it's easy for them to have their credentials disclosed if someone watches them closely

or utilizes a video recording device, such a mobile phone. We suggested a graphical password-based authentication system that is resistant to shoulder surfing as a solution to this issue.

REFERENCES

[1]. Wantong zheng, Chunfu Jia, 'CombinedPWD: A New Password Authentication Mechanism Using Separators Between Keystrokes": 2017 13th International Conference on Computational Intelligence and Security (CIS)

[2]. Salisu Ibrahim Yusuf, Moussa Mahamat Boukar, 'User Define Time Based Change Pattern Dynamic Password Authentication Scheme', 2018 14th International Conference on Electronics Computer

[3]. Yang Jingbo, Shen Pingping,' A secure strong password authentication protocol", 2010 2nd International Conference on Software Technology and Engineering

[4]. Hua Wang, Yao Guo, Xiangqun Chen,' DPAC: A Reuse-Oriented Password Authentication Framework for Improving Password Security'', 2008 11th IEEE High Assurance Systems Engineering Symposium

[5]. Salah Refish, 'PAC-RMPN: Password Authentication Code Based RMPN', 2018 International Conference on Advanced Science and Engineering (ICOASE)

[6]. M Hamza Zaki, Adil Husain, M Sarosh 'Secure pattern-key based password authentication scheme'2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)

[7]. Vasundhara R Pagar, Rohini G Pise, 'Strengthening password security through honeyword and Honey encryption technique', 2017 International Conference on Trends in Electronics and Informatics (ICEI)

[8]. S. Sood, A. Sarje, and K. Singh, "Cryptanalysis of password authentication schemes: Current status and key issues," in Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on, Dec 2009, pp. 1–7..

[9]. S. Gurav, L. Gawade, P. Rane, and N. Khochare, "Graphical password authentication: Cloud securing scheme," in Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014International Conference on, Jan 2014, pp. 479–483

[10]. A. Bianchi, I. Oakley, and D.S. Kwon, "The secure haptic keypad: A tactile password system," in Proceedings of the SIGCHI Conference on Human Factors in Computing System. CHI '10. New York, NY, USA: ACM, 2010, 1089–1092.

[11]. E. von Zezschwitz, A. De Luca, and H. Hussmann, Honey, shrunk the keys: Influences of mobile devices on password composition and authentication performance," in Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational, ser. NordiCHI '14. New York, NY, USA: ACM, 2014, pp. 461–470.