Implementation of Security System Using 3-Level Authentication

[1]Nagesh.D Kamble, [2]J.Dharani

[1]Department of Information Security and Computer Forensics, SRM University, Kattankulathur, Chennai, TN, INDIA
[2]Department of Information Technology, SRM University, Kattankulathur, Chennai, TN, INDIA
[1]nageshce@gmail.com , [2]dharani.j@ktr.srmuniv.ac.in

Abstract **- Increasing security has always been an issue since Internet and Web Development came into existence, text based passwords is not enough to counter such problems, which is also an anachronistic approach now. Therefore, this demands the need for something more secure along with being more user-friendly. Therefore, we are trying to increase the security by involving a 3-level security approach, involving text based password at Level 1, Pattern-Lock Based Authentication at Level 2, and automated generated one-time password (received through an automated SMS to the authentic user) at Level 3. And an assiduous effort has been done for thwarting Shoulder attack, Tempest attack, and Brute-force attack at client side, through the use of unique pattern set in the System Authentication plays a crucial role in protecting resources against unauthorized and illegal use. This unique user-friendly System named as 3 Level Security that can be employed in any organization for storing crucial and confidential documents, and ensures the security through its three levels– Firstly-through Text Password, Secondly-through Pattern-Lock based Authentication, and Thirdly-through One-Time Automated Password.**

Keywords **- Text-Based password, pattern-lock password, random code, authentication**

I. INTRODUCTION

Authentication processes may vary from simple password based authentication system to costly and computation intensified authentication systems. Passwords are more than just a key. They serve several purposes. They ensure our privacy, keeping our sensitive information secure. Passwords authenticate us to a machine to prove our identity-a secret key that only we should know. They also enforce non repudiation, preventing us from later rejecting the validity of transactions authenticated with our passwords. Our username identifies us and the password validates us. But passwords have some weaknesses: more than one person can possess its knowledge at one time. Moreover, there is a constant threat of losing your password to someone else with venomous intent.

Password thefts can and do happen on a daily basis, so we need to defend them. Now merely using some random alphabets grouped together with special characters does not assure safety. We need something esoteric, something different along with being user-friendly as our password, to make it secure.. This paper is a unique and an esoteric study of using pattern as password and implementation of an extremely secured system, employing 3 levels of security-(Text Password, Pattern-Lock, and One-Time automated generated password).

II. PROPOSED SECURE SYSTEM AUTHENTICATION PHASES

Registration

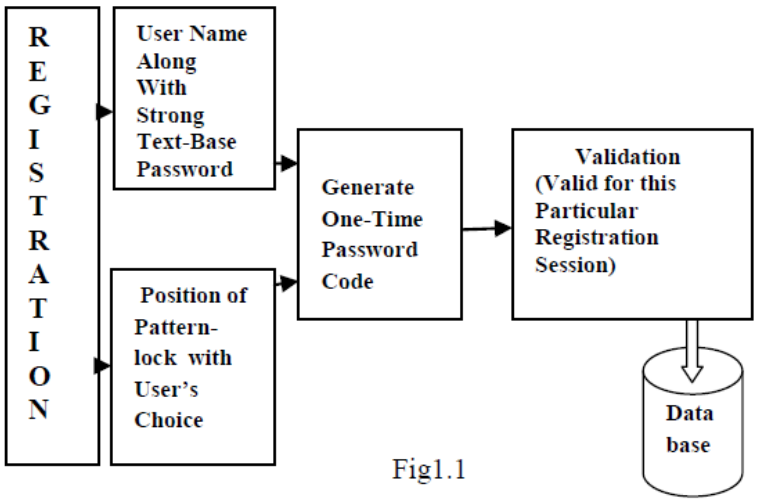

Fig 1.1 Registrations

Authentication

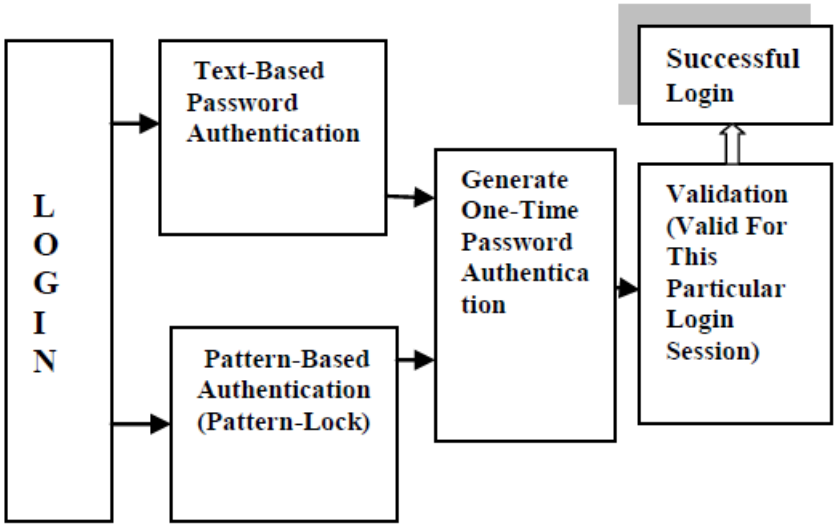

Fig 1.2 Authentication

In the registration phase in Fig1.1 , the user should provide user's details along with his/her user name and user conventional textual password which is as strong as much and difficult to guess. This will protect the system from Tempest attack, Brute-force attack at client side. User have to register with his/her mobile number along with one security question for validation phase of authentication and forget password recovery purpose simultaneously.

Above all, user has to select position of pattern according to his/her choice it's nothing but pattern-lock for that individual user, one advantage that selecting pattern is user can provide any kind of pattern he wanted while registration. Security at text-base level has been imposed by using Text based password (with special characters), which is a usual and now an anachronistic approach. At pattern-lock level the security has been imposed using patterns, where the user will be asked to select an patterns as difficulty level which is unique one for each and every individual user. After preceding above two levels in registration system will generate random-code which is used to provide one-time password authentication level that is next and uppermost third level of authentication. This generated random code is valid for that particular registration phase only. After the successful registration only all the related data about user for authorised/legal use of system (or) application will stored in database.

In the authentication phase in Fig1.2 ,the user should provide user name along with it's registered text-base password for textual password authentication which is level1,after preceding level1 user will ask for entering pattern in pattern-lock at level2, this pattern should match to the pattern in pattern-lock which is unique one and different for each and every user and has selected by user at the time of registration. At this stage pattern should be same as that of registered pattern in pattern-lock for individual user. If it's fails to match simply that user is unauthorised user to access that particular system (or) application.

After preceding above two levels, random-code which has generated by system, will send to registered user's mobile number (or) for application flexibility purpose it can be send to user's entered mobile number at level3 , it's a six digit code, and advantage of this code is that it's valid for current login session only.

If any one of above 3 levels of security get mismatched (or) compromised user will not authenticate to system (or) application simply that would be restricted user. This unique and user-friendly 3-Level Security System is involving three levels of security. Where the preceding level must be passed in order to proceed to next level.

- Security at level1 has been imposed by using Text based password (with special characters), which is a usual and now an anachronistic approach.
- At level2 the security has been imposed using Pattern-Lock authentication where the user will be asked to select pattern levels. For each and every user will have different levels with unique pattern-lock, from where the user has to select any kind of pattern he want.
- After the successful clearance of the above two levels, the Level3 Security System will then generate a one-time numeric password that would be valid just for that registration (or) login session only.

The authentic user will be informed of this one time password on his mobile number. Any hacker if in the extreme case, suppose (although difficult) will cross through the above two mentioned security levels, will definitely not be able to cross the third security level, unless he has access to the original user's mobile number along with mobile device. The user will be authenticated as an authentic user, and will be awarded access to the stored information, or redirect to any secure application, where we want to implement this security approach only after crossing the three security levels as shown in below Fig1.3 (Security level1-Text password, Security level2-Pattern-Lock password, and Security level3- One-Time Automated password).

III. PROPOSED SECURE SYSTEM ARCHITECTURE

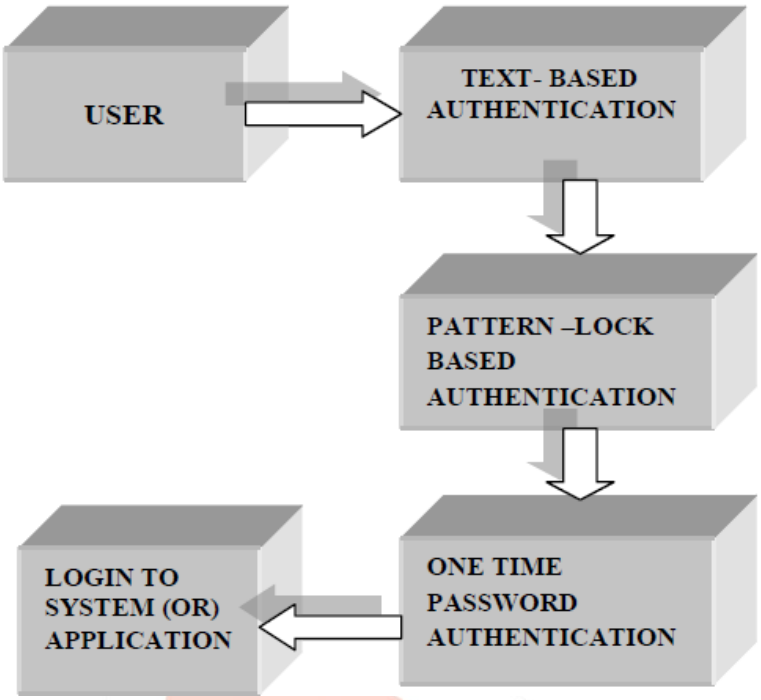

Fig1.3 Architecture Diagram

We proposed this security system as a building platform to access the system (or) any kind of secure application. In this present paper we are implementing this security mechanism to access the web application and redirect the authenticate user to SRM University homepage by using JSP. Java Server Pages (JSP) is a Java technology that allows software developers to dynamically generate HTML, XML or other types of documents in response to a Web client request. The technology allows Java code and certain pre-defined actions to be embedded into static content. The JSP syntax adds additional XML-like tags, called JSP actions, to be used to invoke built-in functionality. Additionally, the technology allows for the creation of JSP tag libraries that act as extensions to the standard HTML or XML tags. Tag libraries provide a platform independent way of extending the capabilities of a Web server.

JSPs are compiled into Java Servlets by a JSP compiler. A JSP compiler may generate a servlet in Java code that is then compiled by the Java compiler, or it may generate byte code for the servlet directly. JSPs can also be interpreted on-the-fly reducing the time taken to reload changes Java Server Pages (JSP) technology provides a simplified, fast way to create dynamic web content. JSP technology enables rapid development of web-based applications that are server and platform-independent. Active Server Pages (ASP). ASP is a similar technology from Microsoft. The advantages of JSP are twofold. First, the dynamic part is written in Java, or Visual Basic or other MS specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.

- Pure Servlets. JSP doesn't give you anything that you couldn't in principle do with a servlet. But it is more convenient to write (and to modify!) regular HTML than to have a zillion println statements that generate the HTML. Plus, by separating the look from the content you can put different people on different tasks: your Web page design experts can build the HTML, leaving places for your servlet programmers to insert the dynamic content.
- Server-Side Includes (SSI). SSI is a widely-supported technology for including externally-defined pieces into a static Web page. JSP is better because it lets you use servlets instead of a separate program to generate that dynamic part. Besides, SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.
- JavaScript. JavaScript can generate HTML dynamically on the client. This is a useful capability, but only handles situations where the dynamic information is based on the client's environment. With the exception of cookies, HTTP and form submission data is not available to JavaScript. And, since it runs on the client, JavaScript can't access server side resources like databases, catalogs, pricing information, and the like.
- Static HTML. Regular HTML, of course, cannot contain dynamic information. JSP is so easy and convenient that it is quite feasible to augment HTML pages that only benefit marginally by the insertion of small amounts of dynamic data. Previously, the cost of using dynamic data would preclude its use in all but the most valuable instances.

Architecture of JSP

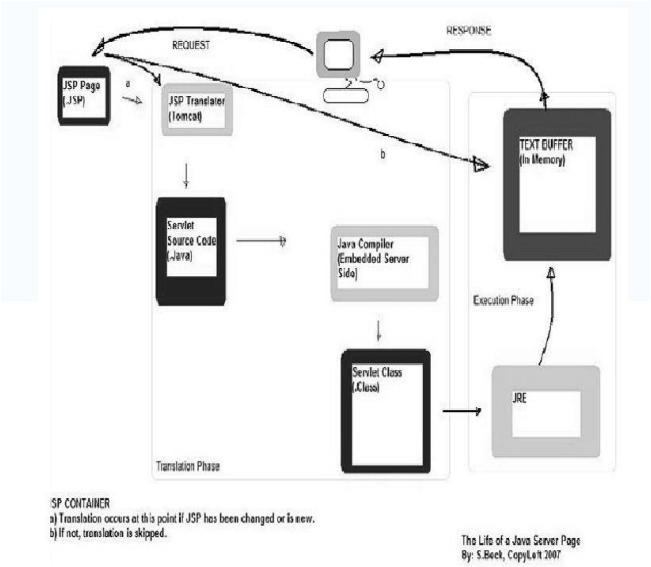

Fig1.4

Java Servlet technology provides Web developers with a simple, consistent mechanism for extending the functionality of a Web server and for accessing existing business systems. Servlets are server-side Java EE components that generate responses (typically HTML pages) to requests (typically HTTP requests) from clients. A servlet can almost be thought of as an applet that runs on the server side—without a face.

```
// Hello.java
importjava.io.*;
importjavax.servlet.*;
public class Hello extends GenericServlet {
public void service(ServletRequest request, ServletResponse response) throws ServletException, IOException{
response.setContentType("text/html");
finalPrintWriter pw = response.getWriter();
pw.println("Hello, world!");
pw.close();
}
}
```

- The import statements direct the Java compiler to include all of the public classes and interfaces from the **java.io** and **javax.servlet** packages in the compilation.
- The Hello class extends the **GenericServlet** class; the GenericServlet class provides the interface for the server to forward requests to the servlet and control the servlet's lifecycle.
- The Hello class overrides the **service(ServletRequest, ServletResponse)** method defined by the Servlet interface to provide the code for the service request handler. The service() method is passed a **ServletRequest** object that contains the request from the client and a**ServletResponse** object used to create the response returned to the client. The service() method declares that it throws the exceptions ServletException and IOException if a problem prevents it from responding to the request.
- The **setContentType(String)** method in the response object is called to set the MIME content type of the returned data to **"text/html"**. The **getWriter()** method in the response returns a **PrintWriter** object that is used to write the data that is sent to the client. The **println(String)** method is called to write the **"Hello, world!"** string to the response and then the **close()** method is called to close the print writer, which causes the data that has been written to the stream to be returned to the client.

By using latest technology mechanism as above mentioned we are proposing web secure application by active server pages to redirect secure and authorise access to web application or secure application, we can implement this proposed system to make application more secure and user-friendly.

IV. CONCLUSION

In this paper, The three level security approach applied on the above system, makes it highly secure along with being more user friendly. This system will definitely help thwarting Shoulder attack, Tempest attack and brute-force attack at the client side. 3-Level Security system is definitely a time consuming approach, as the user has to traverse through the three levels of security, and will need to refer to his mobile number for the one-time automated generated password. Therefore, this system cannot be a suitable solution for general security purposes, where time complexity will be an issue. But will definitely be a boon in areas where high security is the main issue, and time complexity is secondary, as an example we can take the case of a firm where this

system will be accessible only to some higher designation holding people, who need to store and maintain their crucial and confidential data secure. In near future not only we will add more features but also make our system customizable.

REFERENCES

[1] Security Analysis and Implementation of 3-Level Security System Using Image Based Authentication, Author: Surabhi Anand, Priya Jain, Nitin and Ravi Rastogi
[2] S3PAS:A Scalable Shoulder-Surfing Resistant Textual Graphical Password Authentication Scheme, Author: Huanyu Zhao and Xiaolin Li
[3] http://en.wikipedia.org/wiki/Hue
[4] http://en.wikipedia.org/wiki/Color_vision

.