**Security in Web Applications**

**Security**

* **Authentication and Authorization**

**Authentication**

* Checking the identity of users.

**Authorization**

* Verifying the access permission of a user.

**Example Banking App**

* To get into banking app every user should be authenticated
* Based on the role and designation of the logged in user the access permission on the resources will be given or will be denied.
  + Customer role user can operate only Daily Transactions.
  + Clerk role user –Daily ,Loan operation
  + Manager-all operation of app.
* In security application where username and password and roles are maintained is called security realm or Authentication Info Provider.
* While performing authorization do not do by username.
  + Always prefer doing role bcoz if user is deleted or promoted that would not give problems.

**Process**

* As part of user registration process ,insert username and password and roles.
* **Login activity**
  + Collect username and password to perform authentication.
  + Collect roles of a user logged-in to perform authorization.
    - i.e to allow or deny access to user on different services/operations offered by the app

**Two Important Components of Security Implementations**

* Authentication Provider/Authentication Info Provider/Security Realm
* Authentication Manager
* **Authentication Provider/Authentication Info Provider/Security Realm**
* It is the place where username ,passwords and roles will store and managed and will be used for authentication and authorization activities
  + Ex-Xml file-
    - if we forget the password we can get
    - Here we need to encrypt password manually ,we can password though forget it.
  + DB Software
  + LDAP Server(best)—Gmail
    - automatically password encrypted, if we forget we cannot get the password,we can only reset the password
    - Light Weight- Directory Access Protocol.
  + Tomcat server is giving support only for xml file by default
  + Spring/Spring boot security supports all types authentication info provider.
* **Authentication Manager**
* It is the component that perform both authentication and authorization activities by taking security realm/Auth info Provider
* More over,it give error page with error number accordingly
  + Error Code 401 -🡪Authentication failed
  + Error code 403-🡪Authorization failed
* **Two Types of Authentication manages**
  + **Programmatic Auth Manager**
    - We need to develop authentication Manager manually to perform authentication and authorization activities.
  + **Declarative Auth Manager**
    - By adding entries in web.xml file.
    - We can use the underlying server/container supplied ready made Authentication manager to perform authentication and Authorization activities.
    - **Realm-small storage unit**
  + Declarative Auth manager can perform Authentication activities in four modes
    - BASIC
    - DIGEST
    - FORM
    - CLIENT-CERT
* **BASIC**
  + Internally uses base64 algorithm to encode username and password in travelling
    - They don’t give username and password as it is,
    - They will add some extra character ,delete some character. Somehow doing encode.
  + Makes the browser to generate the dialog box for collecting username ,password from end-users.
  + Works with all browsers
* **DIGEST**
  + Same as BASIC , but MD5 algorithm for encoding and decoding.
    - MD5 ->Message Digestive 5
* **FORM**
  + Same as basic ,but allow the programmer to design his choice form page for gathering credentials from end-user.
    - Username
    - Password
  + Also allow to configure error pages of programmer choice for authentication and authorization failure.t
* **CLIENT –CERT**
  + It is no way related Authentication and authorization
  + It is all about configuring digital certificates in the server by enabling “https” protocol and bring those digital certificates at client side and using them for encoding and decoding activities.
* **Procedure to place username password and roles in tomcat server by taking xml file as the Authentication info provider**
  + Go to servers folder of Eclipse project Explorer
    - Tomcat--🡪tomcat-users.xml file.
* **<?xml version='1.0' encoding='utf-8'?>**
* **<tomcat-users>**
* **<role rolename="tomcat"/>--**
* **<role rolename="role1"/>---defining roles**
* **<user username="tomcat" password="tomcat" roles="tomcat"/>**
* **<user username="both" password="tomcat" roles="tomcat,role1"/>**
* **<user username="role1" password="tomcat" roles="role1"/>**
* **</tomcat-users>**