**以常见的web应用程序开发中的登录认证界面的设计为例，通常的做法如下1-3。**

**下面的作法有哪些安全问题？如何改进？**

**你的web开发中的登录界面通常还会使用哪种其他的方法？**

1. web页面上用户输入账号、密码，点击登录
2. 请求提交前，web端首先通过客户端脚本如javascript对密码原文进行md5处理
3. 请求提交到后台，验证账号与密码是否与数据库中的一致，一致则认为登录成功，反之失败

**安全问题：**如果使用第一种方式提交，密码是以明文的方式传到后台，在这个传输的过程中，是以明文方式传输的，容易被第三方直接窃取。如果web前端对密码进行了MD5加密，如果用户的密码过于简单，即使加密了，也会很快被破解出来。请求提交到了后台，如果别人窃取了这个用户的密码，那么就可以直接登录，不安全。

**改进：**1.限制用户输入一些非常容易被破解的密码。如什么123456, password之类，就像twitter限制用户一样做一个密码的黑名单。另外，可以限制用户密码的长度，是否有大小写，是否有数字，你可以用程序做一下校验。

2.不要让浏览器保存密码。很多用户可能会使用一些密码 管理工具来保存密码，浏览器只是其中一种。是否让浏览器保存这个需要你做决定，重点是看一下你的系统的安全级别是否要求比较高，如果是的话，则不要让浏览 器保存密码，并在网站明显的位置告诉用户――保存密码最安全的地方只有用户的大脑。

3、密码在网上的传输。因为HTTP是明文协议，所以，用户名和口令在网上也是明文发送的，这个很不安全。你可以看看这篇文章你 就明白了。要做到加密传输就必需使用HTTPS协议。

4.在设计网页时使用一些安全框架，比如shiro,SpringSecurity。使用这些安全框架后，在后端会加一个盐值对密码进一步加密，即使有相同的密码，经过随机生成的盐值后，存到数据库中的密码也不一样，这样减少了题目中第三种情况的产生的安全问题。

5.千万不要在cookie中存放用户的密码。加密的密码都不行。因为这个密码可以被人获取并尝试离线穷举。所以，一定不能把用户的密码保存在cookie中。虽然现在好多站点都在这么干。

6.正确设计“记住密码”。这个功能简直就是一个安全隐患。一般的设计是

——用户勾选了这个功能，系统会生成一个cookie，cookie包括用户名和一个固定的散列值，这个固定的散列值一直使用。这样，就可以在所有的设备和客户上都可以登录，而且可以有多个用户同时登录。这个并不是很安全。下面是一些更为安全的方法：简单方式：（登录的时候，若是有“记住我”，那么，生成一个唯一的加密token，可以是 用户名/过期时间/用户id/登录时间 等混合加密；在另外一张表维护；每次来，就判断用户名和对应的token，这个token就相当于一个临时的密码）

1）在cookie中，保存三个东西――用户名，登录序列，登录token。

用户名：明文存放。  
登录序列：一个被MD5散列过的随机数，仅当强制用户输入密码时更新（如：用户修改了密码）。  
登录token：一个被MD5散列过的随机数，仅一个登录session内有效，新的登录session会更新它。

**我的web开发中的登录界面通常还会使用哪种其他的方法？**

使用shiro 安全框架，在前端进行加密，不使用cookie。