* 1. 口令暴力猜解
     1. 前置知识

在当今很多地方以用户名和口令作为鉴权的世界，口令的重要性就可想而知了。口令就相当于进入家门的钥匙。系统如果没有防暴力猜解的机制，很容易被他人使用工具暴力破解账号密码。

**威胁描述：**

如果口令被他人猜到或者破解，他人即可登录系统，进行非授权操作，造成不必要的损失。

**涉及功能点：**

登录。

* + 1. 修复方案

（1）涉及身份认证的的功能场景中，对身份认证机制进行有效的控制，提示信息模糊化（例如:用户名或密码错误）；

（2）对尝试次数阈值进行控制，超过阈值进行限制；

（3）使用图片验证码防止暴力猜解，图片验证码成功使用后应立即自动重置或失效。

参考业务流程图如下：



* + 1. 代码参考

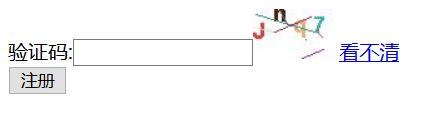
**安全开发实例：**

（1）生成图形验证码参考代码。

代码功能：随机生成4位的图形验证码，包括字大小写字母、数字；对图形验证码进行底纹干扰、 文本位置调整、文本颜色变换、旋转图片字体。

|  |
| --- |
| Public void code**(**HttpServletRequest req**,** HttpServletResponse resp**)throws** ServletException**,** IOException **{**  resp**.**setContentType**(**"image/jpeg"**);**//设置响应内容的类型为jpeg的图片  int width**=**64**;**  int height**=**40**;**  BufferedImage bImg**=new** BufferedImage**(**width**,** height**,** BufferedImage**.**TYPE\_INT\_RGB**);**  Graphics g**=**bImg**.**getGraphics**();**  //背景  g**.**setColor**(**Color**.**white**);**  g**.**fillRect**(**0**,** 0**,** width**,** height**);**  //字体颜色  g**.**setFont**(new** Font**(**"aa"**,** Font**.**BOLD**,**18**));**  // 验证码中所使用到的字符  char**[]** codeChar **=** "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"**.**toCharArray**();**  String captcha **=** ""**;** // 存放生成的验证码  Random r**=new** Random**();**  **for(**int i**=**0**;** i**<**4**;** i**++)** **{** //设置长度单位，随机取的个数  int t**=**r**.**nextInt**(**codeChar**.**length**);**// 循环将每个验证码字符绘制到图片上  int y**=**10**+**r**.**nextInt**(**20**);**//上下位置:10~30  Color c**=new** Color**(**r**.**nextInt**(**255**),** r**.**nextInt**(**255**),** r**.**nextInt**(**255**));**  g**.**setColor**(**c**);**  g**.**drawString**(**codeChar**[**t**]+**""**,** i**\***16**,** y**);**  captcha **+=** codeChar**[**t**];**  **}**  //画干扰线  **for(**int i**=**1**;** i**<**8**;** i**++)** **{**  Color c**=new** Color**(**r**.**nextInt**(**255**),** r**.**nextInt**(**255**),** r**.**nextInt**(**255**));**  g**.**setColor**(**c**);**  g**.**drawLine**(**r**.**nextInt**(**width**),** r**.**nextInt**(**height**),** r**.**nextInt**(**width**),** r**.**nextInt**(**height**));**  **}**  // 将生成的验证码code放入sessoin中  req**.**getSession**().**setAttribute**(**"code"**,** captcha**);**  //把图形刷到bImg对象中  g**.**dispose**();**//相当于IO中的close()方法带自动flush();  ImageIO**.**write**(**bImg**,**"JPEG"**,** resp**.**getOutputStream**());**//通过resp获取req的outputStream对象,发向客户端的socket的封装,即写到客户端  **}** |

效果图如下：



（2）验证图形验证码参考代码：

代码功能：验证图形验证码内容，验证成功后，重置图形验证码。

|  |
| --- |
| protected void doPost**(**HttpServletRequest req**,** HttpServletResponse resp**)**  **throws** ServletException**,** IOException **{**  // 获取存放在session中的验证码  String code **=** **(**String**)** req**.**getSession**().**getAttribute**(**"code"**);**  // 获取页面提交的验证码  String inputCode **=** req**.**getParameter**(**"code"**);**  **if(**code**.**toLowerCase**().**equals**(**inputCode**.**toLowerCase**()))** **{** // 验证码转换为小写，验证图片验证码内容  **new** ImageServlect**().**code**(**req**,**resp**);** //验证成功后重新调用生成验证码方法，将验证过的验证码重置  checkCode**();** //此处根据实际业务，做相应的操作，如验证账号密码、发送短信等  **}** **else** **{** // 验证失败  req**.**getRequestDispatcher**(**"/fail.jsp"**).**forward**(**req**,** resp**);**  **}**  **}** |