第十节课

课程回顾

1. 课程详情页面的显示,信息的处理
2. 通过分类的处理课程预习处理好.

课后

3. 模板的继承,这个再讲一下
4. 前端传递到后面的数据 和view之后的返回数据.

今天的课程

1. 说一些关于web安全问题

   https://blog.csdn.net/toto1222/article/details/52780139

   csrf攻击

   http://www.cnblogs.com/hyddd/archive/2009/04/09/1432744.html

2. sql攻击 ,利用在输入的时候的漏洞,进行查询时候,django的form会对我们的表单提交的时候进行验证,authenticate也会对用户的名字和密码进行验证.

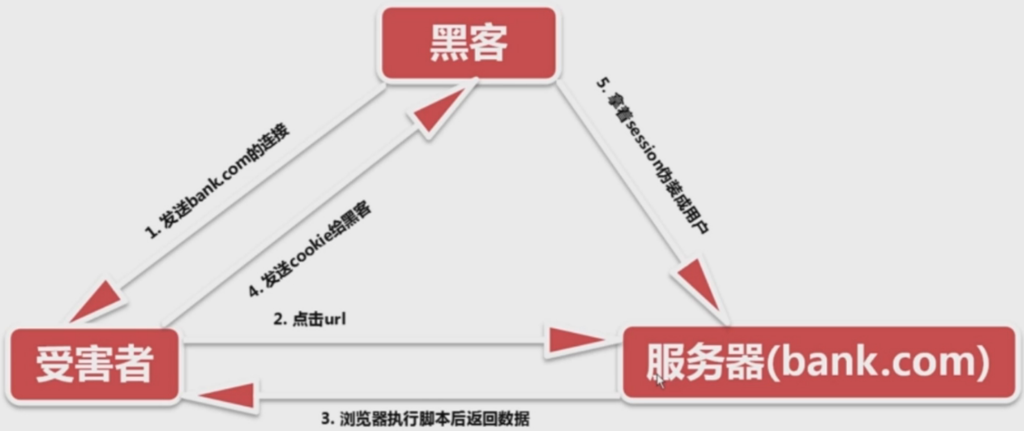
```
# 重写登录页面
class LoginUnsafeView(View):
    def get(self, request):
        return render(request, 'login.html', {})

    def post(self, request):
        user_name = request.POST.get("username", "")
        pass_word = request.POST.get("password", "")

        import pymysql
        conn = pymysql.connect(host='192.168.83.128', user='develop', passwd='QWEqwe123',
db='tanzhoudb', charset='utf8')
        cursor = conn.cursor()
        sql_select = "select * from users_userinfo WHERE  email='{0}' AND
password='{1}'".format(user_name, pass_word)

        result = cursor.execute(sql_select)
        for row in cursor.fetchall():
            pass
        print('123')
```

3. 这代码是对用户输入账号和密码的时候没有验证的,
4. 在前端使用 账号:' OR 1=1 # 密码:随便写一段
5. 使用authenticate的方法会对非法的字符转义.

6. xss攻击.---钓鱼

XSS攻击流程

http://www.bank.com/product/list/?name='iphone6'



http://www.bank.com/product/list/?name=<script>x=document.cookie;alert(x);</script>

xss攻击防护

xss攻击防护

首先代码里对用户输入的地方和变量都需要仔细检查长度和对
" <" ," >" ," ;" ," ' " 等字符做过滤；

避免直接在cookie 中泄露用户隐私，例如email、密码等等
通过使cookie 和系统ip 绑定来降低cookie 泄露后的危险

尽量采用POST 而非GET 提交表单

7. csrf攻击

csrf攻击

<img src=http://www.mybank.com/Transfer/toBankId=11&money=1000>

csrf跨站请求伪造(Cross-site request forgery)的危害

以你名义发送邮件

盗取你的账号

购买商品

虚拟货币转账

CSRF的防御

1. 服务端进行CSRF防御
    1. Cookie Hashing(所有表单都包含同一个伪随机值)：(csrf_token)
    2. 验证码.
    3. One-Time Tokens(不同的表单包含一个不同的伪随机值)