案例：<https://blog.csdn.net/weixin_45139919/article/details/90633729>

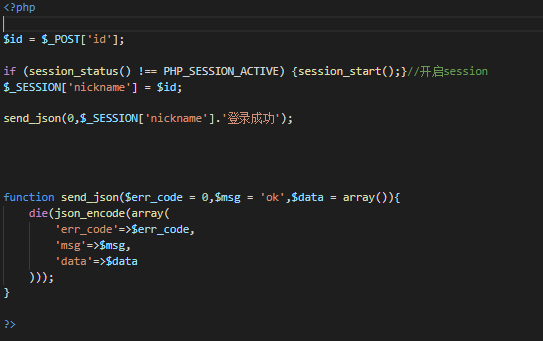
刚刚了解这个接口安全，内容未必正确，大神请指点，萌新别全信

以下内容纯属虚构，如有模仿，后果自负

假设这是一个简陋的转账接口



这是一个简陋的登录



小明想转账给小红，于是他登录之后点击页面上的转账按钮触发了

http://localhost.com/ajaxclass/test\_csfr\_php.php?money=10&who=小红

这样一个链接。

小奸发现了这个链接于是他就在浏览器上直接访问

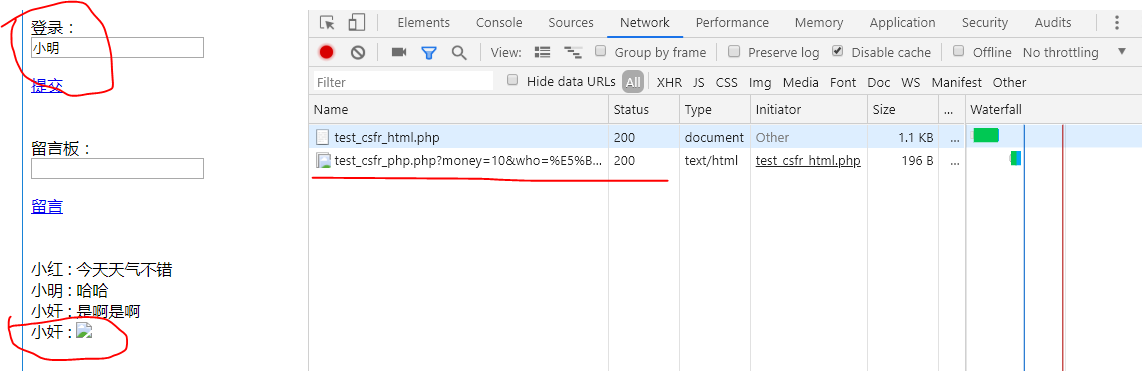
http://localhost.com/ajaxclass/test\_csfr\_php.php?money=10&who=小奸

但是返回了没有登录

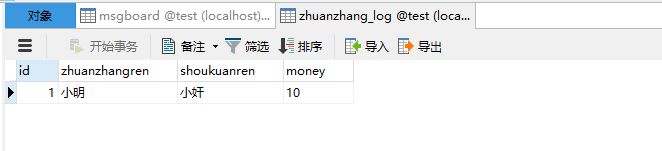
于是小奸想到了一个办法，他在留言板留言了这样的内容：

<img src=\"http://localhost.com/ajaxclass/test\_csfr\_php.php?money=10&who=小奸\" />

然后当其他人打开这个网页的时候......



意外就这么发生了......



这就是小奸用了XSS和CSFR攻击，

如果小奸输入了的是：

<img style=”z-index:9999;width:100%;height:100%” src=\”小奸的成人用品店广告图\” />

这样小奸用的就是纯粹的XSS攻击，

总的来说：

XSS攻击就是页面注入htnl、css、js代码，

CSFR就是在用户不知情的情况下被发送了恶意请求，这里只是其中一种手法，有些可能绕过后端的跨域检测或者其他漏洞直接发请求

然后稍微总结一下关于接口安全的防范措施：

1. 敏感接口用post不用get

POST 请求不会被缓存

POST 请求不会保留在浏览器历史记录中

2、验证请求来源

HTTP里面的Referer值就是请求来源，可以通过这个判断是否自己的域名或者对应的页面地址发出的请求。

后来我又发现表单直接post是没有这个Referer，而且可以跨域的，

然后我又想强制用请求提交，但是$\_SERVER里面判断是否请求提交那个标志好像不靠谱，而且Referer这个标志好像也是能伪造的

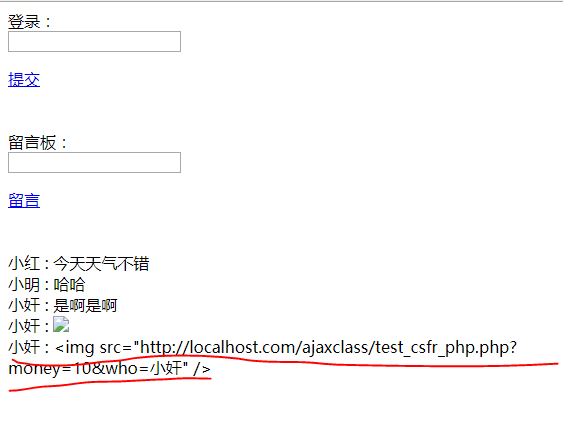
然后又发现了表单post的话可以用HTTP\_REFERER来验证，但还是那句能伪造的

等我哪天发现了防御方法再写，

而且别人还能通过上传漏洞上传了恶意代码的文件去攻击，那这来源检测也没用了

3、htmlspecialchars

对输入内容进行html转实体操作后，页面输出时就不会解析成html



4、addslashes

还有这个防sql注入的

5、用户使用接口权限注册

Token和refresh\_token

用户第一次注册接口使用权限时生成token和refresh\_token，token用于每次请求携带验证用户身份，refresh\_token用于换取token。之所以这样设计是因为token是每次请求都会用到的，所以容易被抓取，而refresh\_token是换取token的时候才用到，使用频率较低，所以没那么容易抓取到

6、页面使用权限接口注册

对于敏感重要的接口，页面每次生成时，向后端发送敏感接口的url注册，后端记录该页面和该接口url和用户信息(ip,id等)，并生成随机限时token保存后端并返回，前端页面记录token过期时间，够钟就重新发请求注册接口，页面调用该接口时附带该token到后端验证

7、页面参数加密

后端给前端提供敏感参数时，双向加密一波，比如抽奖，一等奖的奖品id是123456789，那别人就知道一等奖的id是这个了，就有机可乘了。

8、接口提交参数签名并加密

这个参数签名我还没理解清楚作用，好像是应对参数篡改的情况，具体还得有空实操一遍

9、公钥私钥的也还没理解清楚

10、https

参考：

<https://www.cnblogs.com/accumulater/p/6178166.html>

<https://zhuanlan.zhihu.com/p/22521378>

<https://www.cnblogs.com/shytong/p/5308667.html>

<https://blog.csdn.net/lynnlovemin/article/details/80638765>