# CSRF

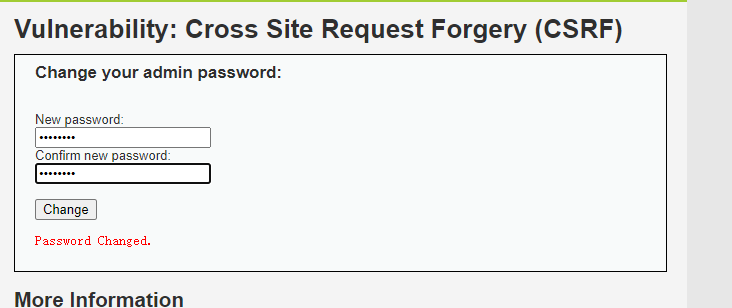
CSRF，全称Cross-site request forgery，翻译过来就是跨站请求伪造，是指利用受害者尚未失效的身份认证信息（cookie、会话等），诱骗其点击恶意链接或者访问包含攻击代码的页面，在受害人不知情的情况下以受害者的身份向（身份认证信息所对应的）服务器发送请求，从而完成非法操作（如转账、改密等）。CSRF与XSS最大的区别就在于，CSRF并没有盗取cookie而是直接利用。在2013年发布的新版OWASP Top 10中，CSRF排名第8。

# Level:Low



通过Low级别的源码可以看到，服务器收到修改密码的请求后，会检查参数password\_new与password\_conf是否相同，如果相同，就会修改密码，并没有任何的防CSRF机制

构造一个链接



输入修改的密码之后它的url链接是这样的



就直接在url链接上显示出来了

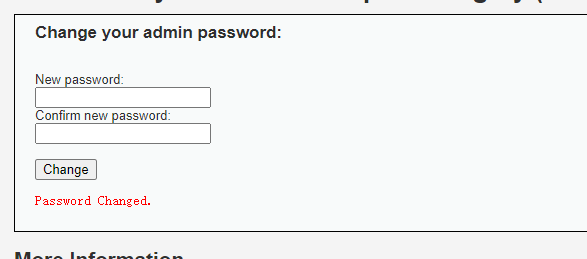
然后再把要改的密码给替换成一个自己知道的密码



弄成这样，但是这样的话太明显了，别人会察觉到不对劲，这个时候就要给它用url编码进行编码

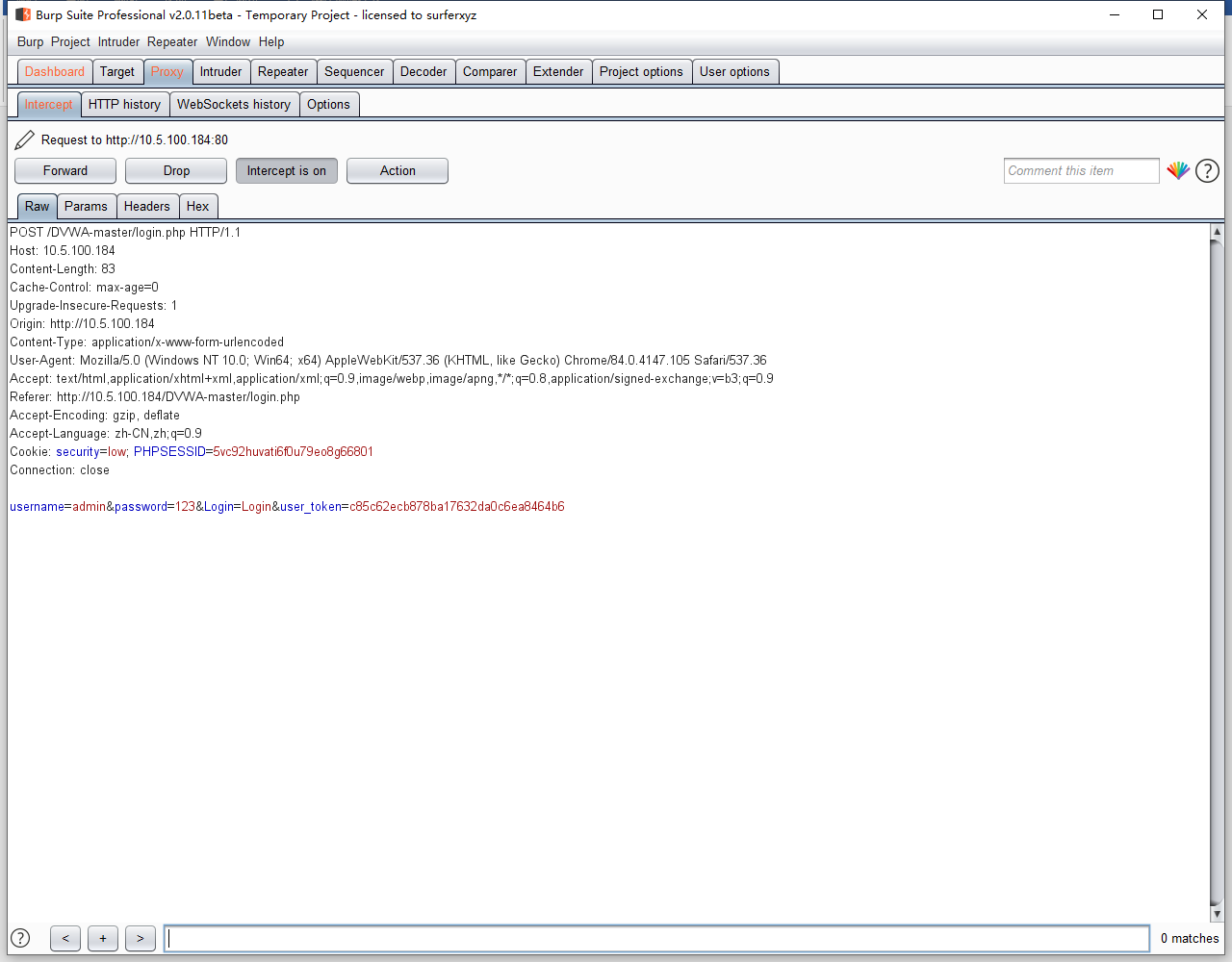
最后变成这样

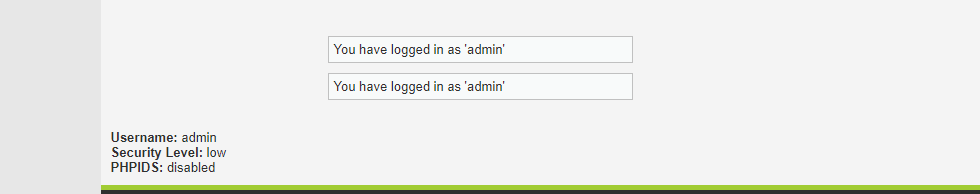
[http://10.5.100.184/DVWA-master/vulnerabilities/csrf/?%70%61%73%73%77%6f%72%64%5f%6e%65%77=%31%32%33&%70%61%73%73%77%6f%72%64%5f%63%6f%6e%66=%31%32%33&%43%68%61%6e%67%65=%43%68%61%6e%67%65#](http://10.5.100.184/DVWA-master/vulnerabilities/csrf/?%70%61%73%73%77%6f%72%64%5f%6e%65%77=%31%32%33&%70%61%73%73%77%6f%72%64%5f%63%6f%6e%66=%31%32%33&%43%68%61%6e%67%65=%43%68%61%6e%67%65)进行编码之后就不会那么明显了，点击之后



也会显示改密成功，现在它的密码就是123了，

去登录界面用bp抓包查看是否能登录成功





登录成功

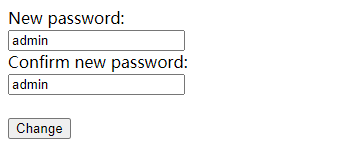
# Level:Medium

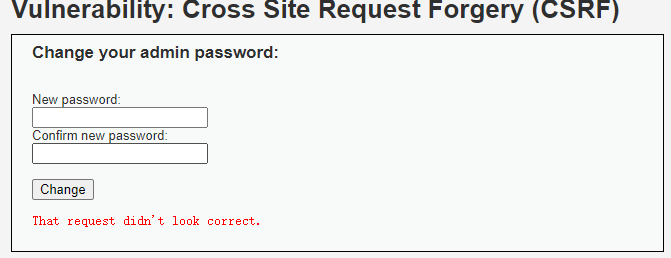


Medium级别的源码用变量$strips对比了变量$\_SERVERP‘HTTP\_REFERER’和变量$\_SERVERP’SERVERP\_NAME’的host请求头，都是本地，如果要是外部的地址就会修改不成功

创建一个表单

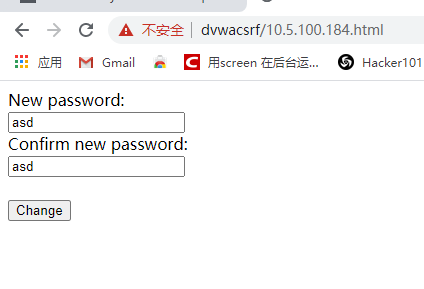


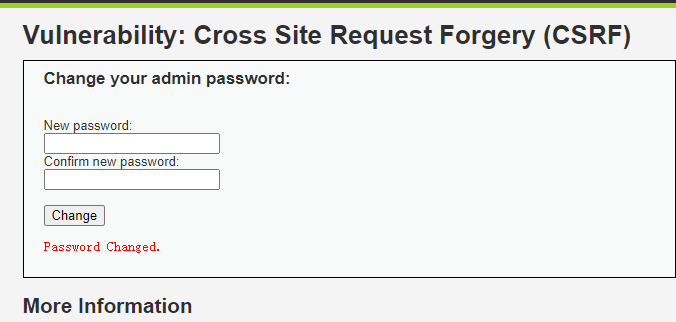
然后打开表单



执行失败是因为请求的地址和原来的不一样，这个时候可以修改

表单的名字改为它的ip地址





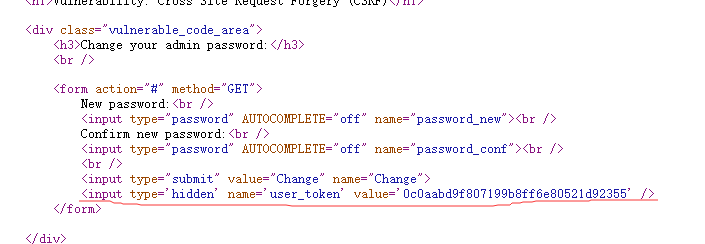
这样就可以修改成功了

# Level:High



High级别的源码的意思是这一次是检查用户的token才能改用户的密码

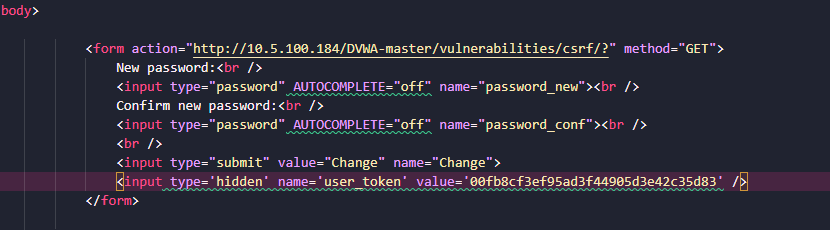
查看网页源码

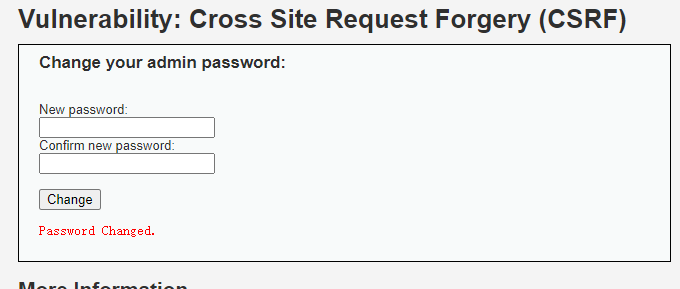


多出了user\_token

把这一段给自己的表单添加

然后给自己的表单添加





# Level:Impossible



可以看到，Impossible级别的代码利用PDO技术防御SQL注入，至于防护CSRF，则要求用户输入原始密码,在不知道原密码的情况下怎么用CSRF修改密码都是不允许的