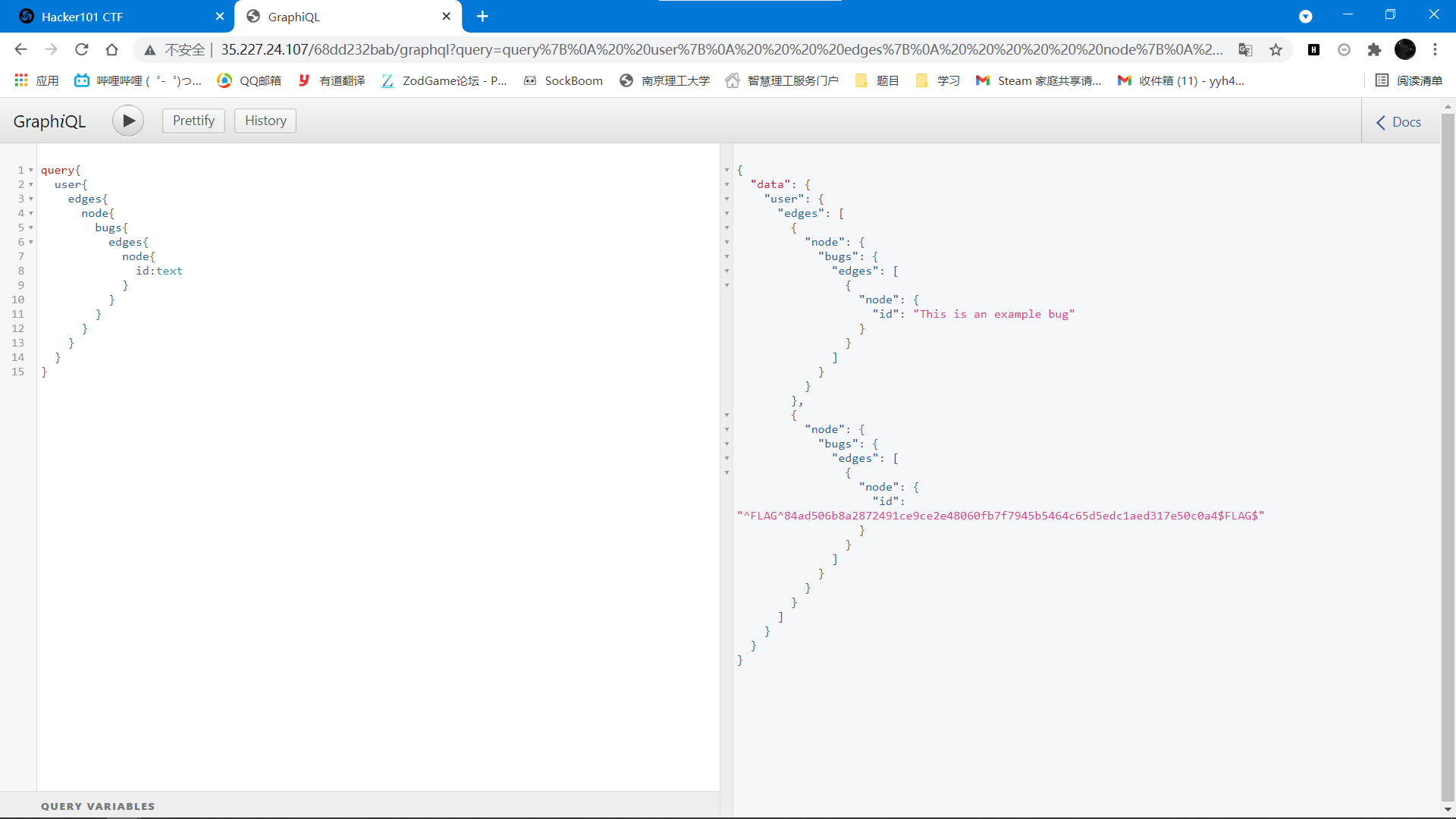
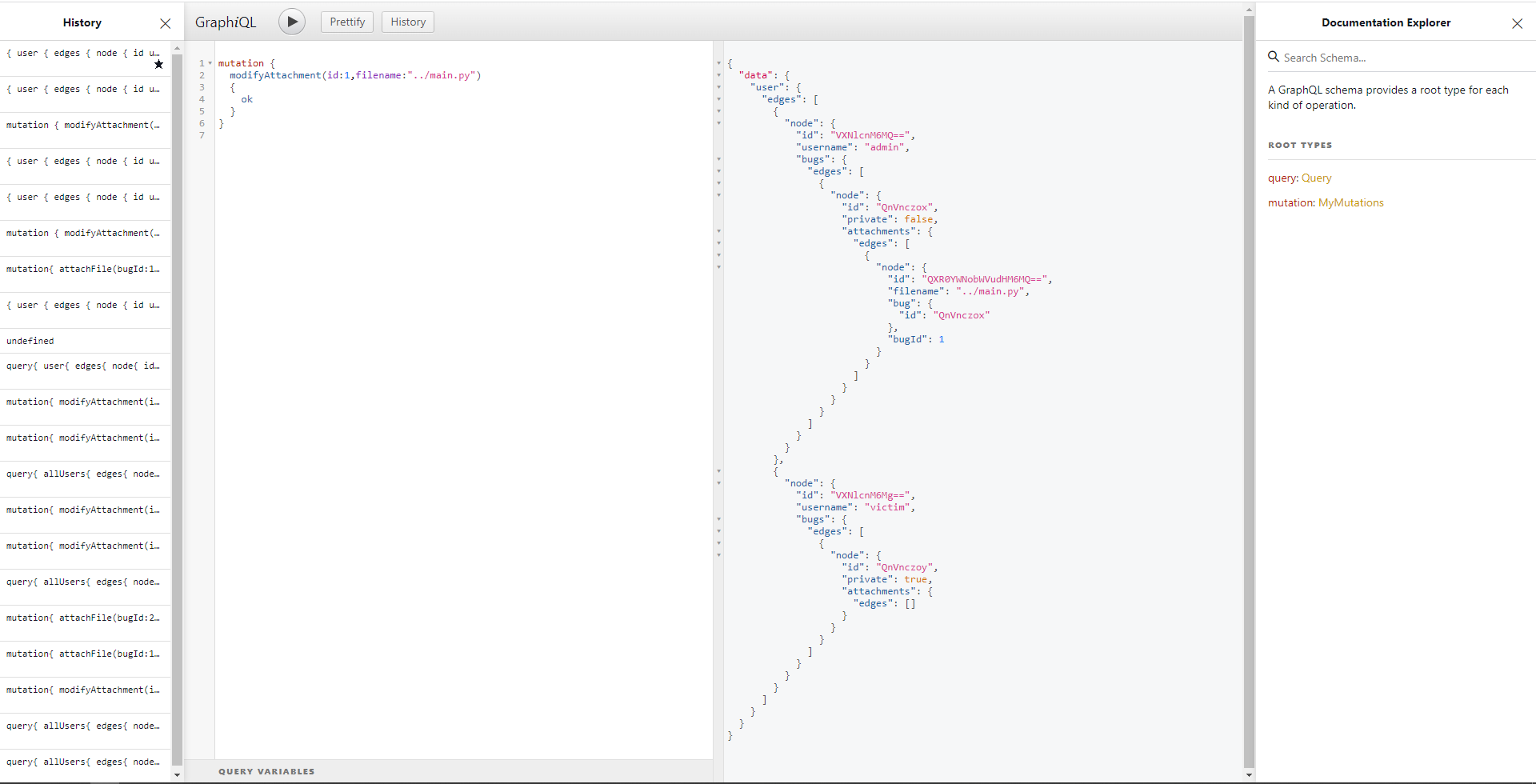
1.hacker101 bugdb v1

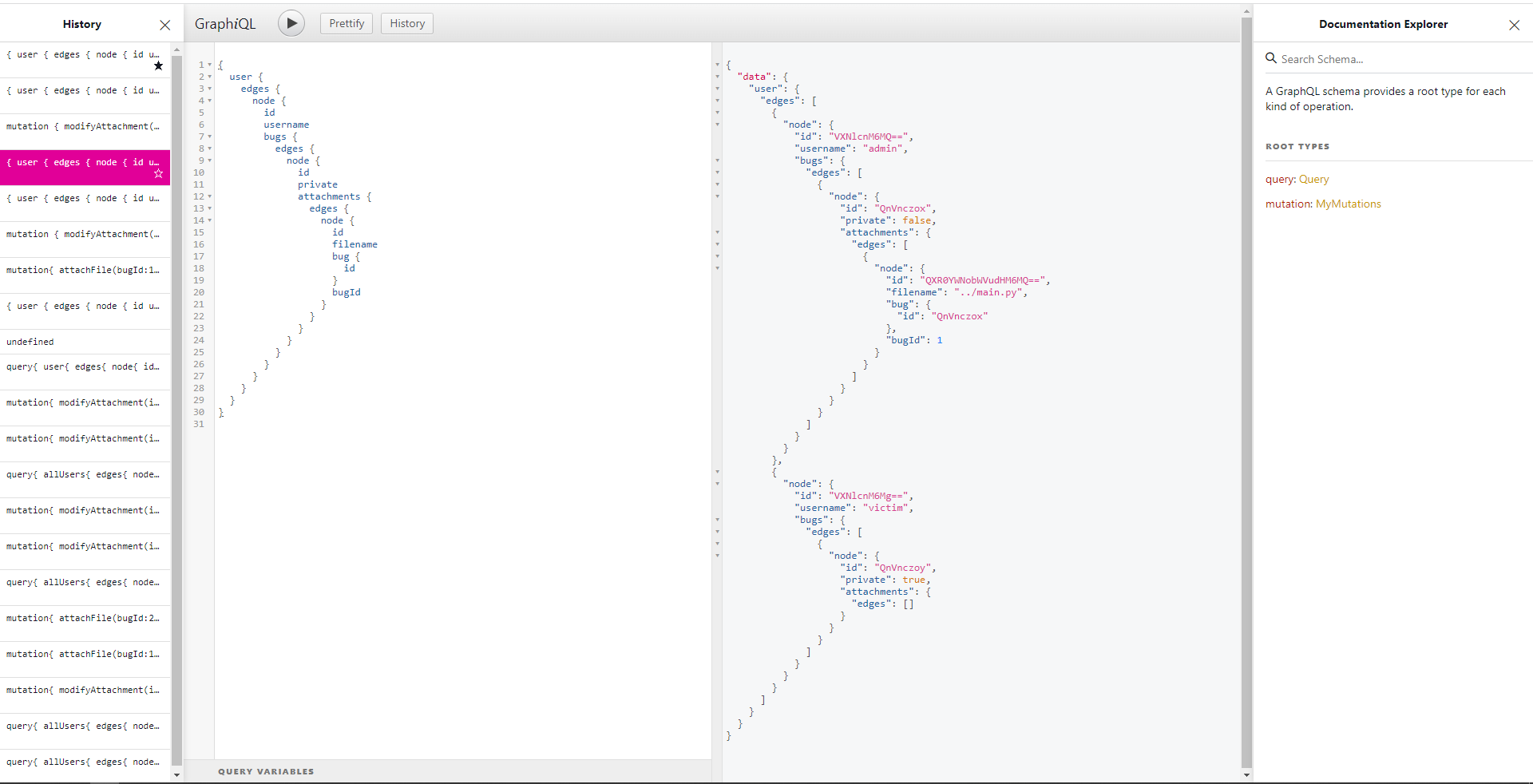


目录翻翻就翻到了？不太懂找的方法

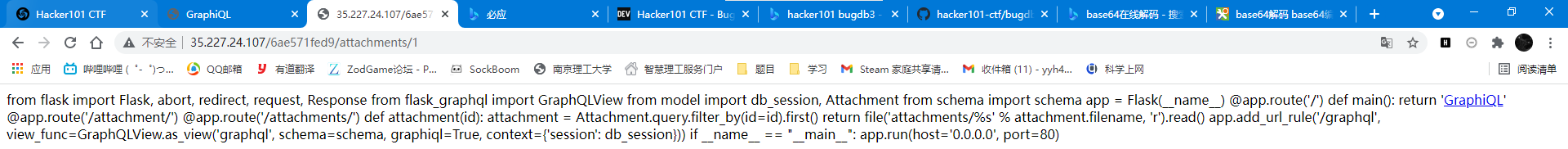
2.

3. hacker101 bugdb v3





刚开始查看不了attachment的内容，但是修改其内部文件名后就能查看了，发现filename确实被修改了，然后就懵逼了，然后查阅write up，说是根据题目hint提示" How do you access attachments? Hint: not via GraphQL"猜测查看attachments文件的方法是直接查看网站的attachments的路径，然后发现如下代码



from flask import Flask, abort, redirect, request, Response

from flask\_graphql import GraphQLView

from model import db\_session, Attachment

from schema import schema

app = Flask(\_\_name\_\_)

@app.route('/')

def main():

return 'GraphiQL'

@app.route('/attachment/')

@app.route('/attachments/')

def attachment(id):

attachment = Attachment.query.filter\_by(id=id).first()

return file('attachments/%s' % attachment.filename, 'r').read()

app.add\_url\_rule('/graphql', view\_func=GraphQLView.as\_view('graphql', schema=schema, graphiql=True, context={'session': db\_session}))

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host='0.0.0.0', port=80)

一段python代码，暂时还确实读不太懂，马上去学习，目前根据writeup推测../model.py下有某些信息，执行

mutation{

modifyAttachment(id:1, filename:"../model.py"){

ok

}

}

再次访问相同路径，根据内容再次修改路径

mutation{

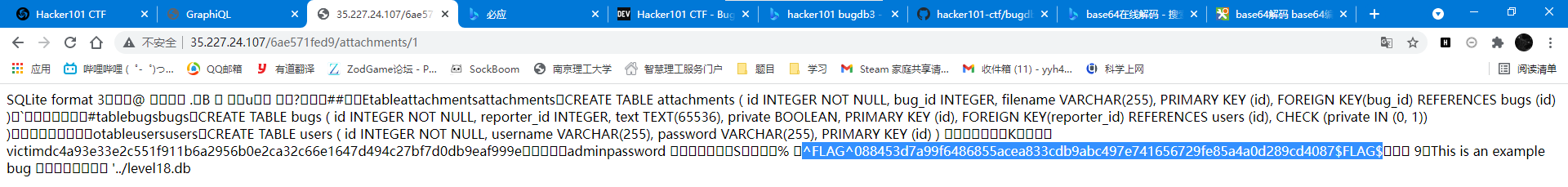
modifyAttachment(id:1, filename:"../level18.db"){

ok

}

}

Get the FLAG



总结：对网站路径仍不太敏感；对python本身以及和网页结合的相关知识仍了解甚少，上面那几个python文件基本看不懂

[HFCTF 2021 Final]hatenum

直接给出了源文件，重点登陆语句

$res = $this->conn->query("select \* from users where username='$username' and password='$password'");

过滤语句

function sql\_waf($str){

if(preg\_match('/union|select|or|and|\'|"|sleep|benchmark|regexp|repeat|get\_lock|count|=|>|<| |\\*|,|;|\r|\n|\t|substr|right|left|mid/i', $str)){

die('Hack detected');

}

}

function num\_waf($str){

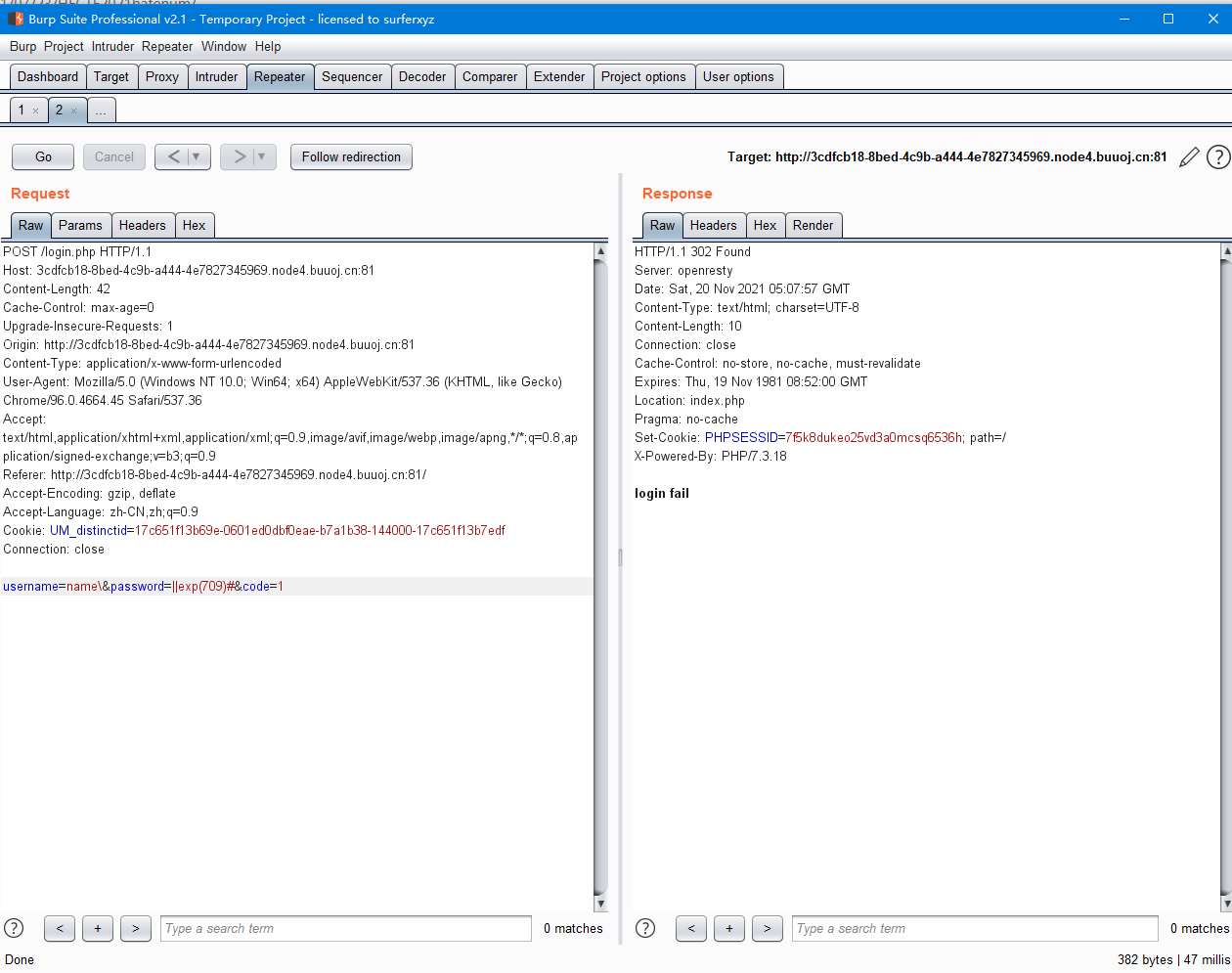
if(preg\_match('/\d{9}|0x[0-9a-f]{9}/i',$str)){

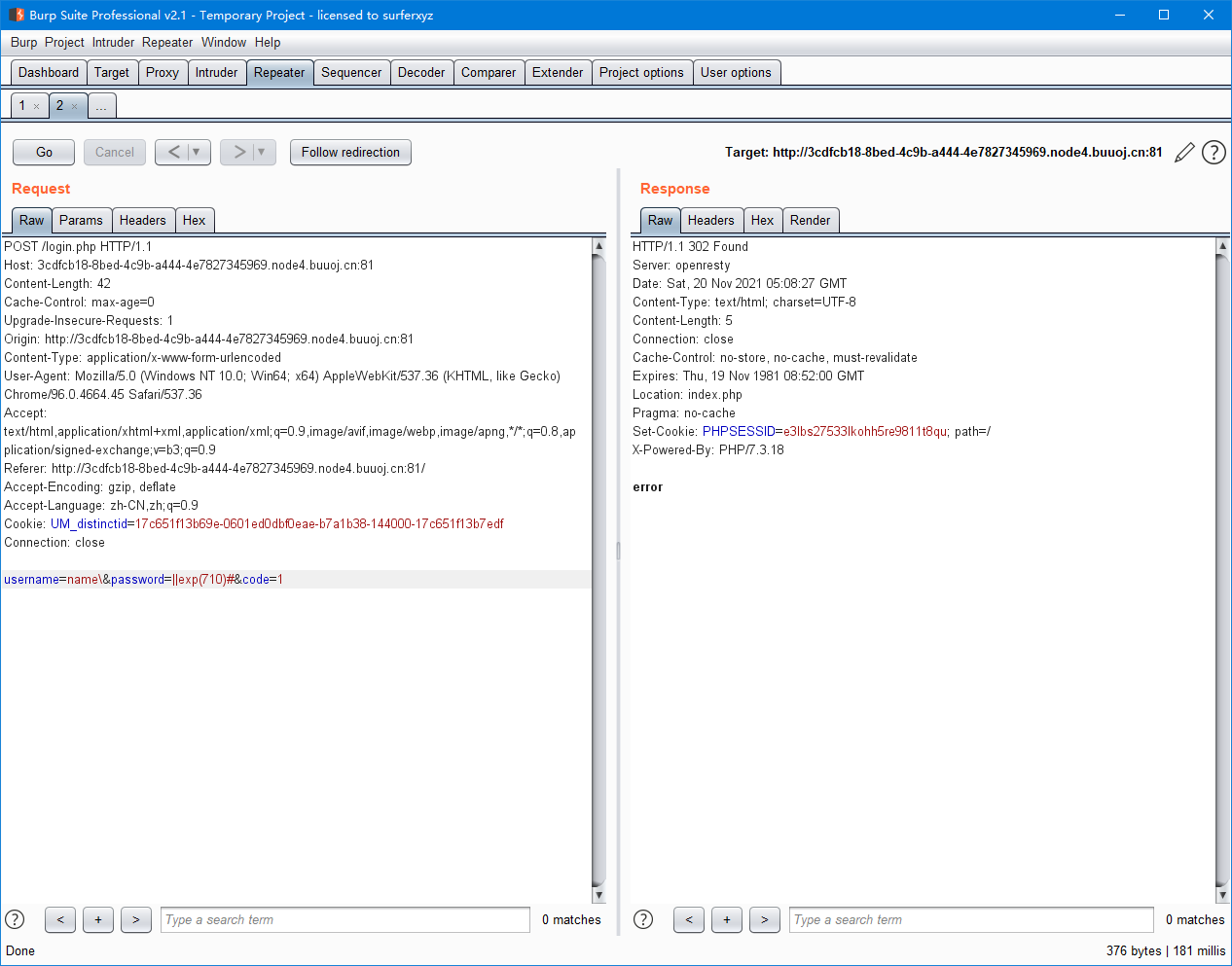
die('Huge num detected');

}

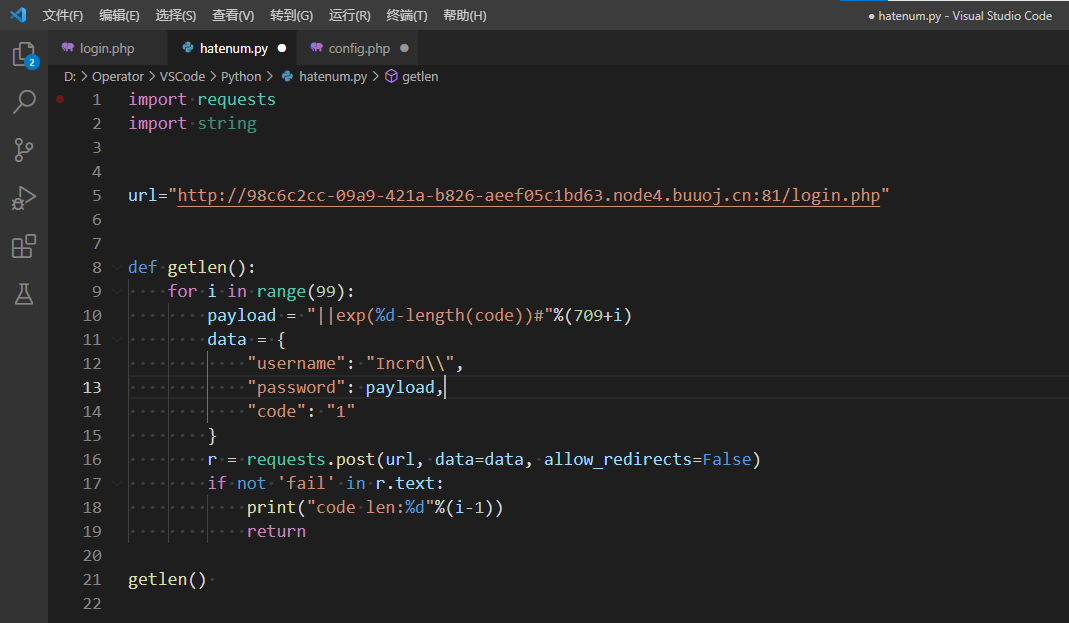
}

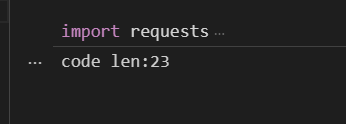
过滤了大部分sql语句和函数，但没有过滤 "\" and "#" ，故可在username处将引号用反斜杠转义，password处用井号注释之后部分，在password处构造得到一个注入点。此处采用报错盲注，虽然preg\_match过滤掉了大数字，不过我们仍然可以通过exp函数来进行溢出，exp(pow)返回e的pow次方，当exp()的参数大于709时就会造成溢出，报错





脚本





得到code长度为23

**rlike语句**

在MySQL中，RLIKE运算符用于确定字符串是否匹配正则表达式。它是REGEXP\_LIKE()的同义词，如果字符串与提供的正则表达式匹配，则结果为1，否则为0。

构造

password=||exp(710-(code rlike xxx))

语句匹配则返回login fail，否则返回error

**空格过滤**

1./\*\*/

2.加括号

3.将空格变成换页键（chr(0x0c)）,换行符,tab键 。

本题可以通过chr(0x0c)绕过

**单引号过滤**

通过binary语法将需要匹配的字符串转成十六进制形式即可

但是本题还限制了数字的长度，十进制和十六进制都不能达到9位，十六进制下按两位一个字符来算，也就是一次最多只能匹配四个字符，而code有23位