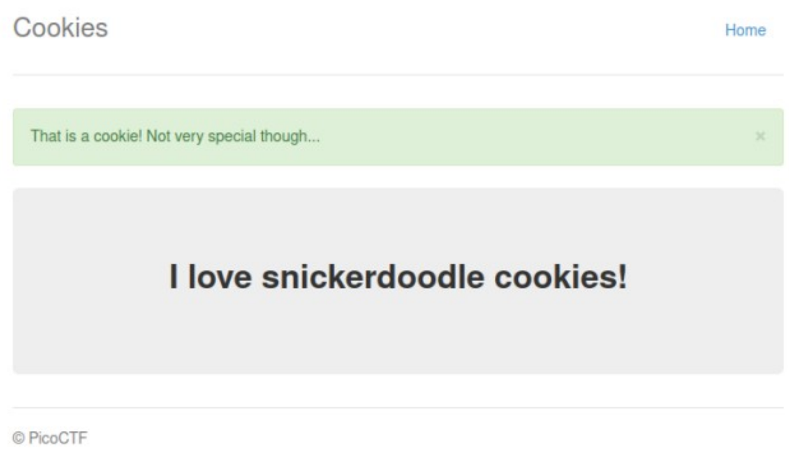
Computer Security

Homework 2

Sthefano Ulloa

Cookies

By inspecting the page I noticed that the value of the only cookie changed from -1 to 0 when I typed ‘snickerdoodle’. Also it showed the following message:



Cookies

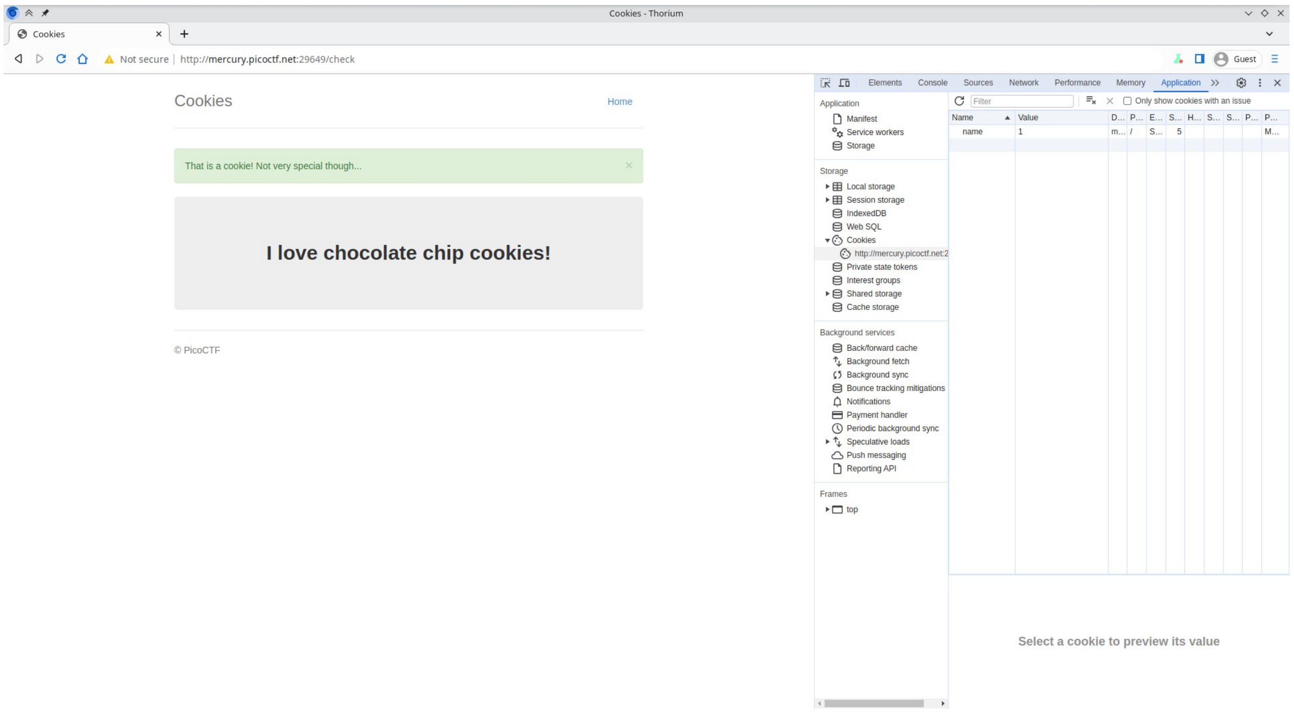
Home

That is a cookie! Not very special though...

I love snickerdoodle cookies!

© PicoCTF

And by changing the values of the cookie and refreshing the page the message changed as follows:



http://mercury.picoctf.net:29649/check

Cookies

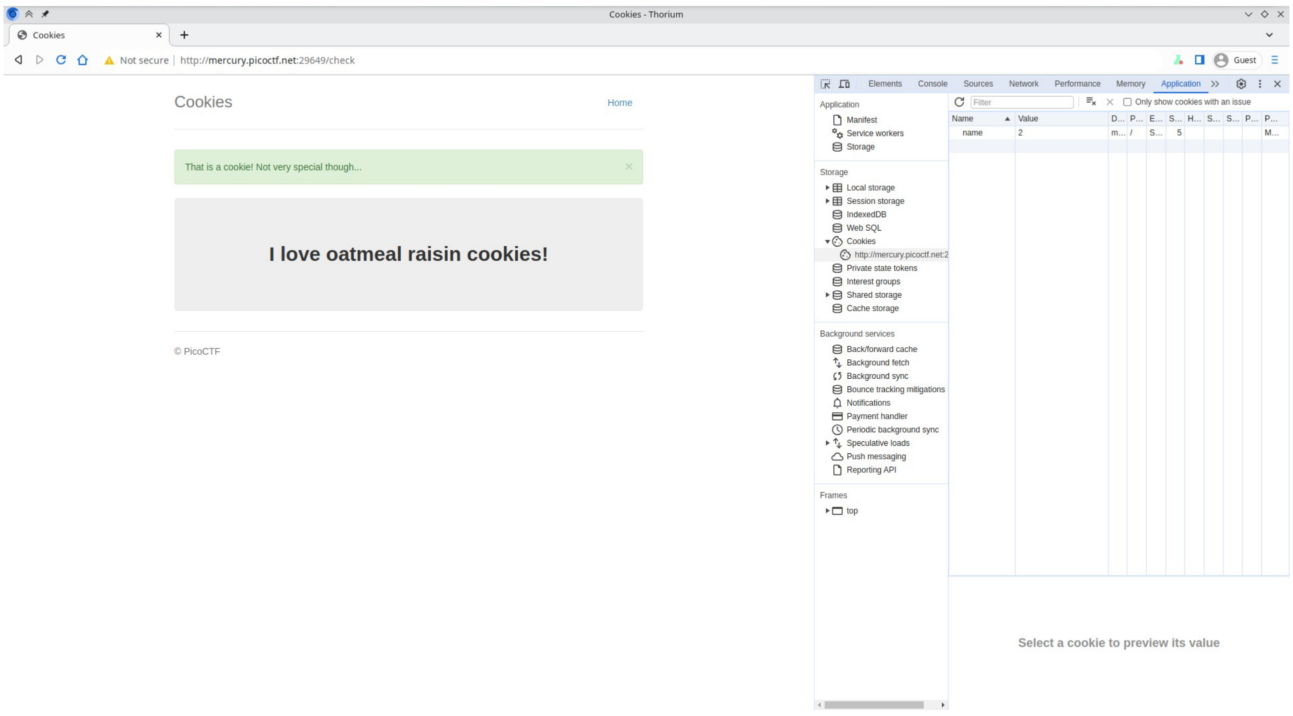
That is a cookie! Not very special though...

I love chocolate chip cookies!

© PicoCTF

Name	Value	D...	P...	E...	S...	H...	S...	S...	P...	P...
name	1	m...	/	S...	5					M...

Select a cookie to preview its value

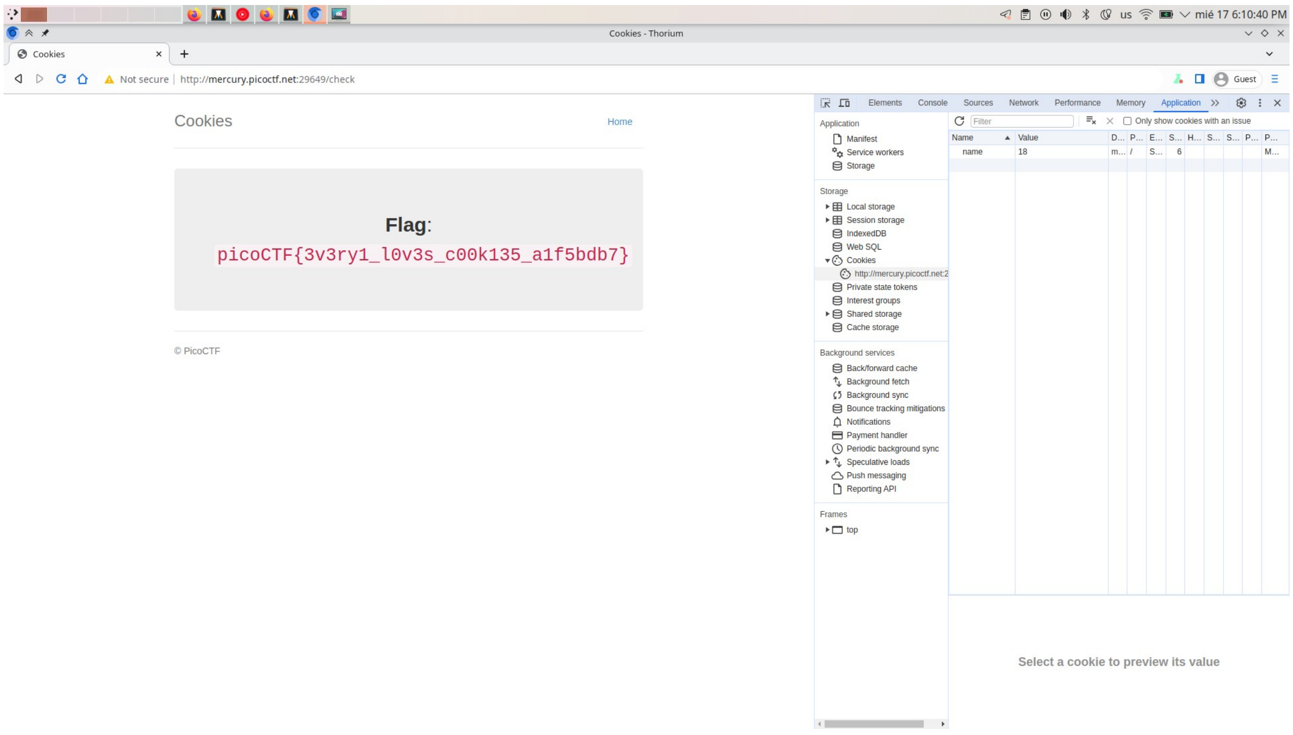

Cookies - Thorium
Cookies
Not secure | http://mercury.picoctf.net:29649/check
Cookies
Home
Elements
Console
Sources
Network
Performance
Memory
Application
Only show cookies with an issue
Select a cookie to preview its value
© PicoCTF

That is a cookie! Not very special though...

I love oatmeal raisin cookies!

Name	Value	D...	P...	E...	S...	H...	S...	S...	P...	P...
name	2	m...	/	S...	5					M...

I changed values until 18, where the message was the flag:


Cookies - Thorium
Cookies
Not secure | http://mercury.picoctf.net:29649/check
Cookies
Home
mié 17 6:10:40 PM
Application
Select a cookie to preview its value
© PicoCTF

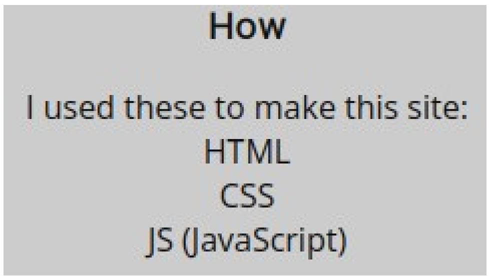
Flag:

```
picoCTF{3v3ry1_l0v3s_c00k135_a1f5bdb7}
```

Name	Value	D...	P...	E...	S...	H...	S...	S...	P...	P...
name	18	m...	/	S...	6					M...

Insp3ct0r

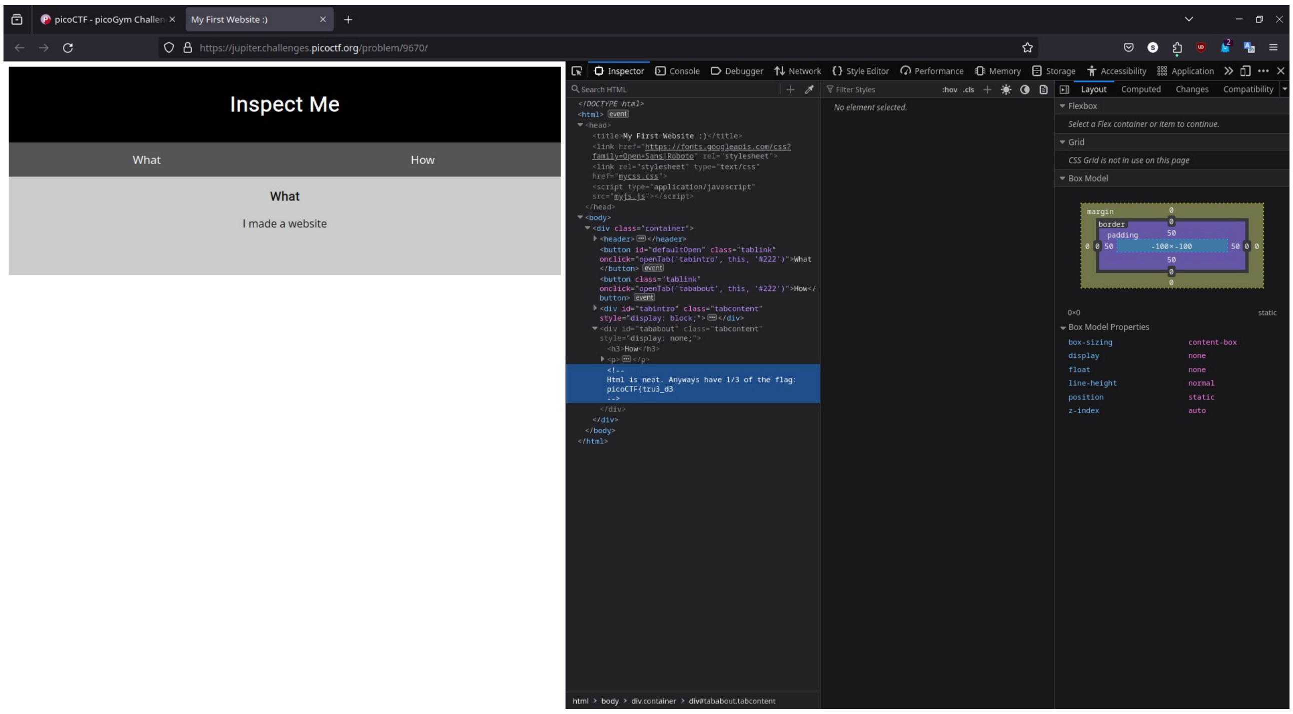
In this problem, it was sugested to inspect the code of the page, and also showed that there were three main elements: html, css and js:



How

I used these to make this site:
HTML
CSS
JS (JavaScript)

And in fact, the flag was divided in three parts, each one in the corresponding components:



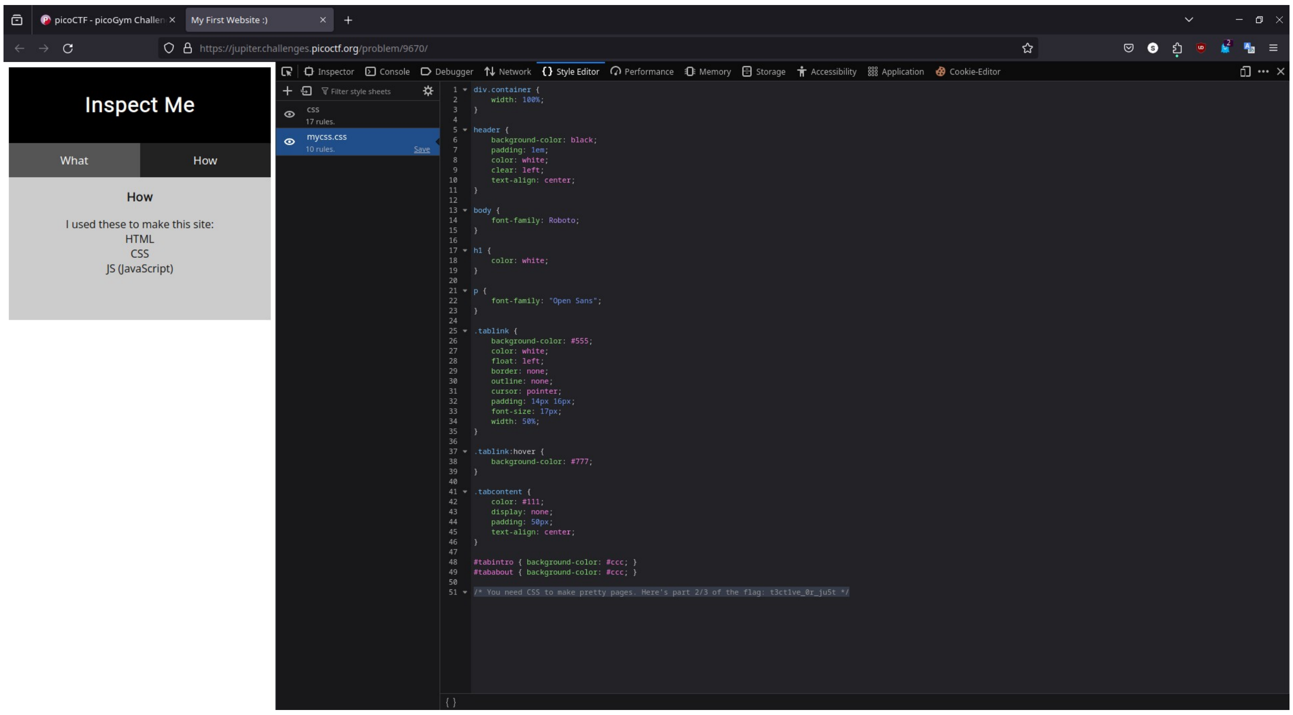
https://jupiter.challenges.picoctf.org/problem/9670/

Inspect Me

What

I made a website

```
<!DOCTYPE html>
<html> event
▾ <head>
    <title>My First Website :)</title>
    <link href="https://fonts.googleapis.com/css?
    family=Open+Sans|Roboto" rel="stylesheet">
    <link rel="stylesheet" type="text/css"
    href="mycss.css">
    <script type="application/javascript"
    src="myjs.js"></script>
  </head>
▾ <body>
  ▾ <div class="container">
    ▸ <header>…</header>
      <button id="defaultOpen" class="tablink"
      onclick="openTab('tabintro', this, '#222')">What
      </button> event
      <button class="tablink"
      onclick="openTab('tababout', this, '#222')">How</
      button> event
    ▸ <div id="tabintro" class="tabcontent"
      style="display: block;">…</div>
    ▾ <div id="tababout" class="tabcontent"
      style="display: none;">
        <h3>How</h3>
      ▸ <p>…</p>
        <!--
        Html is neat. Anyways have 1/3 of the flag:
        picoCTF{tru3_d3
        -->
      </div>
    </div>
  </body>
</html>
```



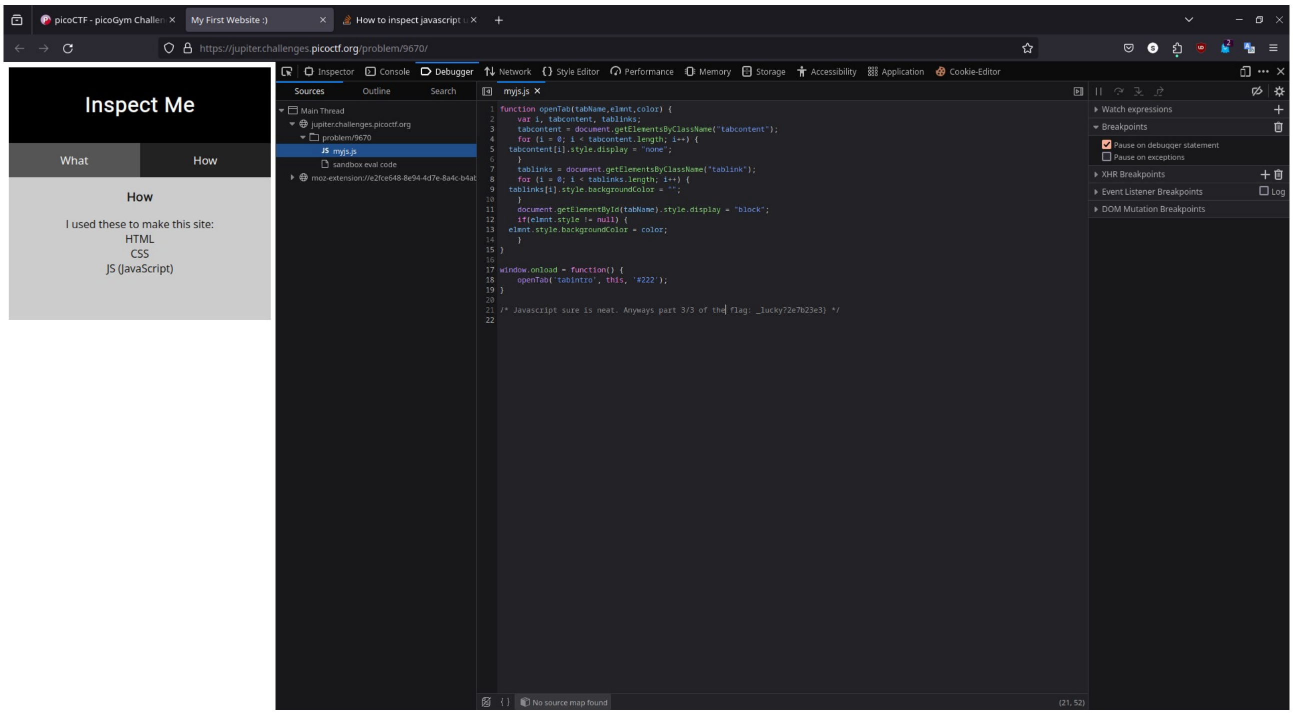
https://jupiter.challenges.picoctf.org/problem/9670/

Inspect Me

How

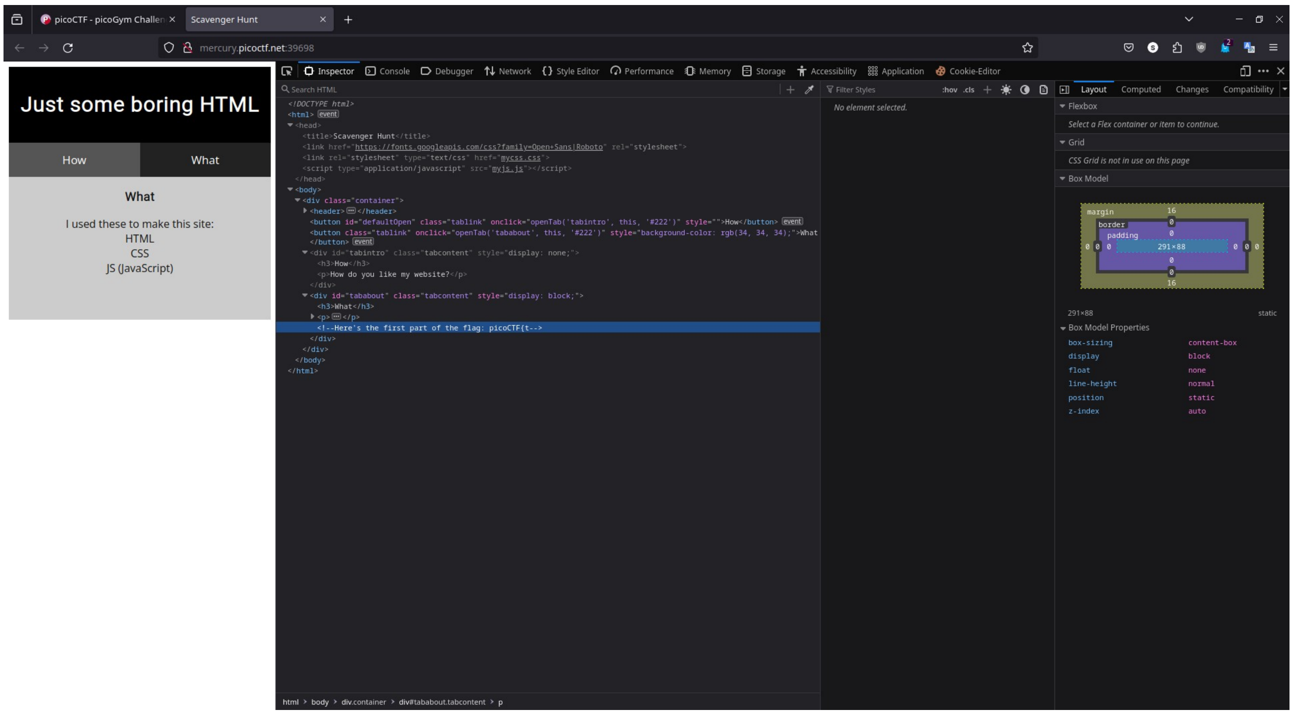
I used these to make this site:
HTML
CSS
JS (JavaScript)

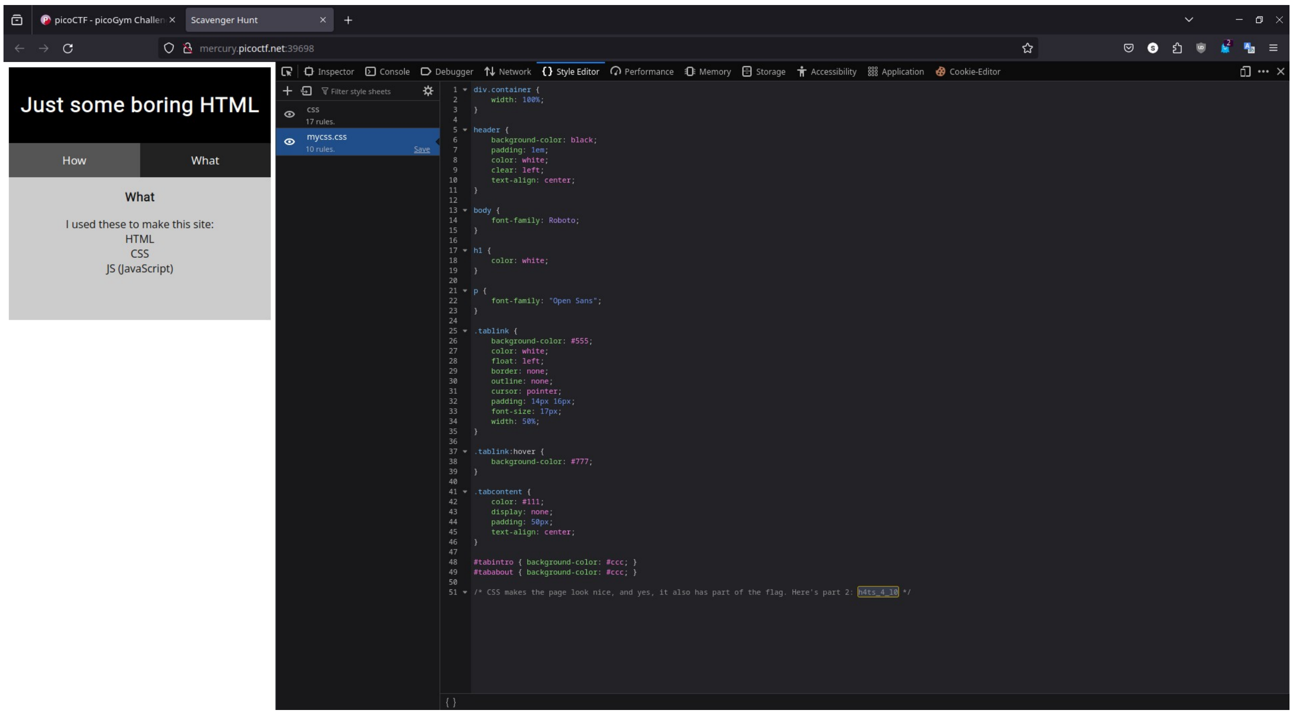
```
div.container {
    width: 100%;
}

header {
    background-color: black;
    padding: 1em;
    color: white;
    clear: left;
    text-align: center;
}

body {
    font-family: Roboto;
}

h1 {
    color: white;
}

p {
    font-family: "Open Sans";
}

.tablink {
    background-color: #555;
    color: white;
    float: left;
    border: none;
    outline: none;
    cursor: pointer;
    padding: 14px 16px;
    font-size: 17px;
    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; }

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


```
function openTab(tabName,elmnt,color) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
  tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablink");
    for (i = 0; i < tablinks.length; i++) {
  tablinks[i].style.backgroundColor = "";
    }
    document.getElementById(tabName).style.display = "block";
    if(elmnt.style != null) {
  elmnt.style.backgroundColor = color;
    }
}

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

/* Javascript sure is neat. Anyways part 3/3 of the flag: _lucky?2e7b23e3} */
```

Scavenger Hunt

This problem started similar to the previous one, part 1 and 2 where on the html and css code:


```
<!DOCTYPE html>
<html>
<head>
  <title>Scavenger Hunt</title>
  <link href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto" rel="stylesheet">
  <link rel="stylesheet" type="text/css" href="mycss.css">
  <script type="application/javascript" src="myjs.js"></script>
</head>
<body>
  <div class="container">
    <header>…</header>
    <button id="defaultOpen" class="tablink" onclick="openTab('tabintro', this, '#222')" style="">How</button>
    <button class="tablink" onclick="openTab('tababout', this, '#222')" style="background-color: rgb(34, 34, 34);">What
    </button>
    <div id="tabintro" class="tabcontent" style="display: none;">
      <h3>How</h3>
      <p>How do you like my website?</p>
    </div>
    <div id="tababout" class="tabcontent" style="display: block;">
      <h3>What</h3>
      <p>…</p>
      <!--Here's the first part of the flag: picoCTF{t-->
    </div>
  </div>
</body>
</html>
```



Just some boring HTML

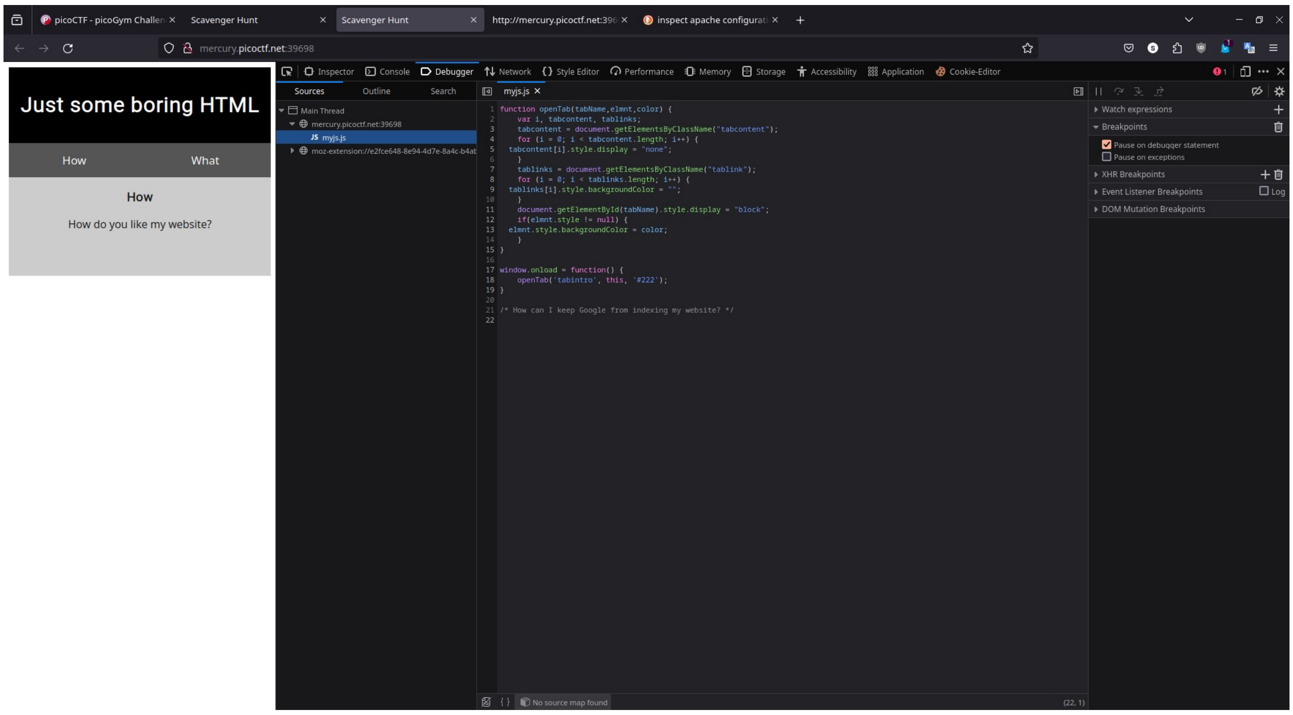
How What

What

I used these to make this site:
HTML
CSS
JS (JavaScript)

```css
div.container {
    width: 100%;
}

header {
    background-color: black;
    padding: 1em;
    color: white;
    clear: left;
    text-align: center;
}

body {
    font-family: Roboto;
}

h1 {
    color: white;
}

p {
    font-family: "Open Sans";
}

.tablink {
    background-color: #555;
    color: white;
    float: left;
    border: none;
    outline: none;
    cursor: pointer;
    padding: 14px 16px;
    font-size: 17px;
    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 */
```

But there wasn't a third one on the js, only a clue about not indexing the page on google:



Just some boring HTML

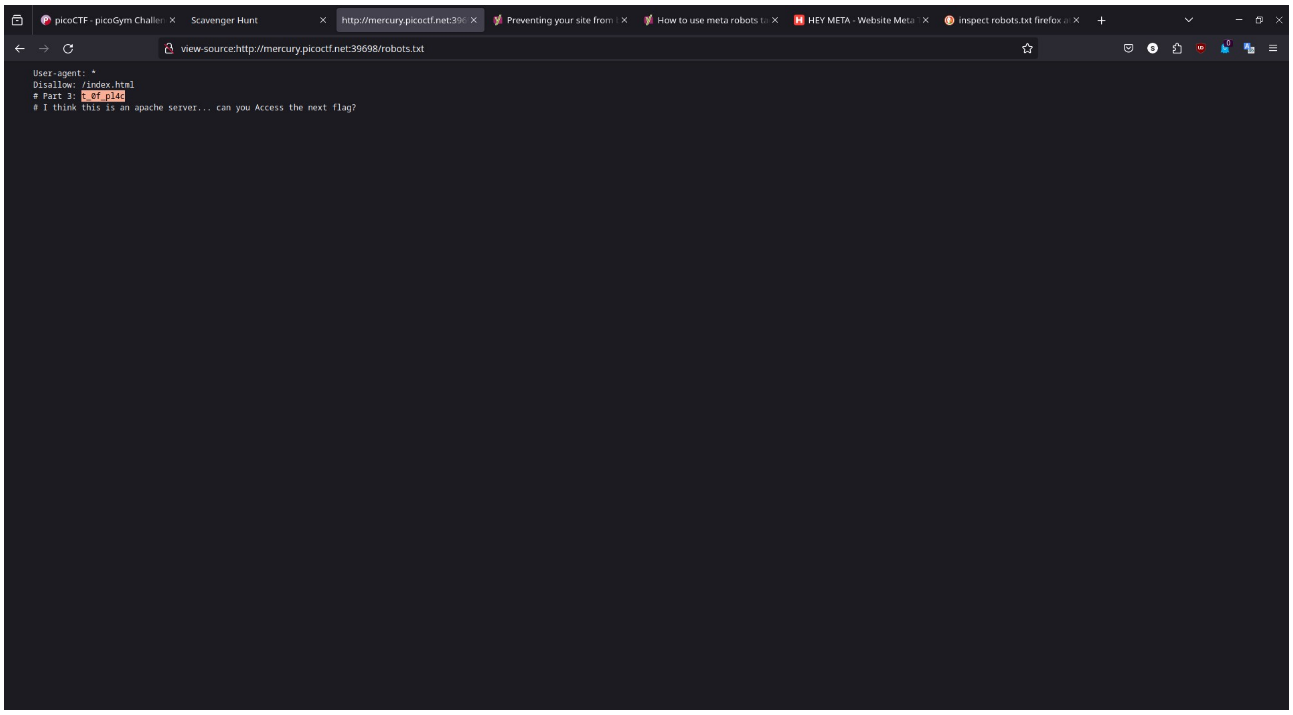
How What

How

How do you like my website?

```javascript
function openTab(tabName,elmnt,color) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
  tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablink");
    for (i = 0; i < tablinks.length; i++) {
  tablinks[i].style.backgroundColor = "";
    }
    document.getElementById(tabName).style.display = "block";
    if(elmnt.style != null) {
  elmnt.style.backgroundColor = color;
    }
}

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

/* How can I keep Google from indexing my website? */
```

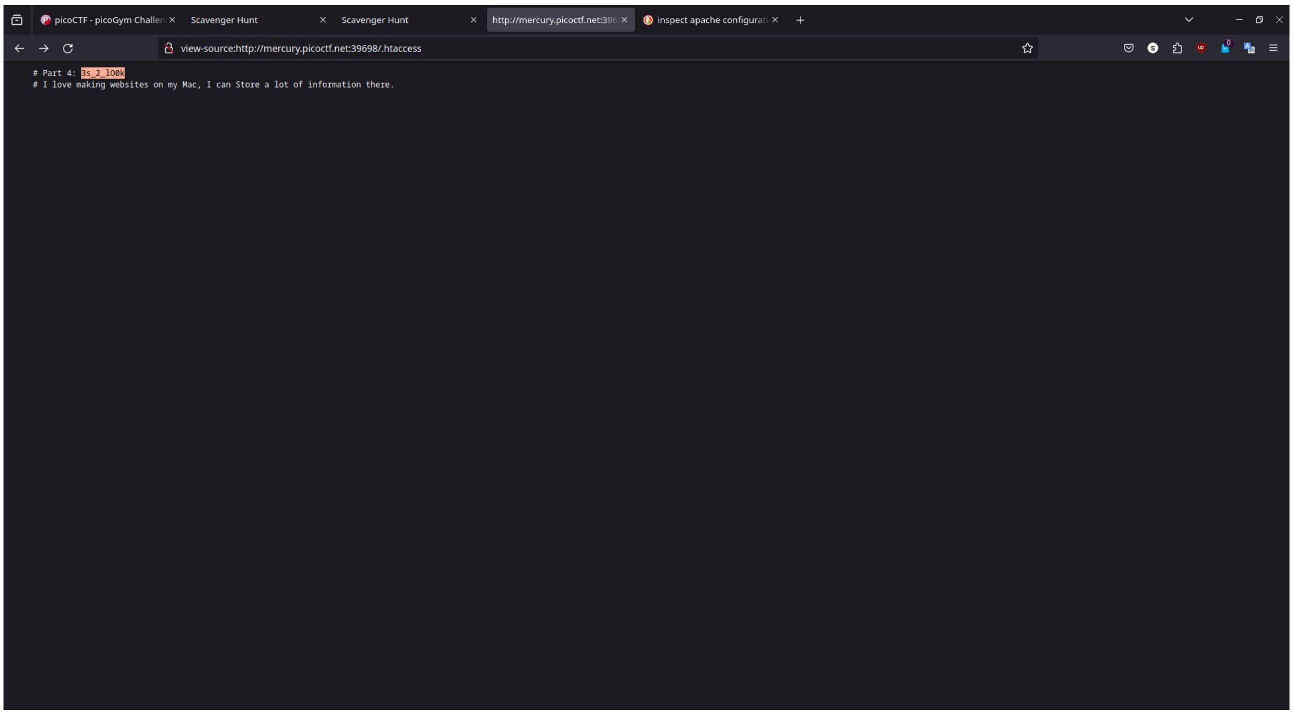
After searching that in google, I learn there is a source file called robot.txt where you can specify what parts of a page can be indexed to search engines (aka. Robots or crawlers). So by adding '/robots.txt', I got to see this file content:



view-source:http://mercury.picoctf.net:39698/robots.txt

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

The next part was related to apache and I thought that it may be another file similar to this robots.txt so after researching this a lot I found that there is a file called '.htaccess' user for several configurations on the server like URL Rewriting, Access Control, Error Handling, etc. So, with the same approach as the previous, the fourth part of the flag was found:



view-source:http://mercury.picoctf.net:39698/.htaccess

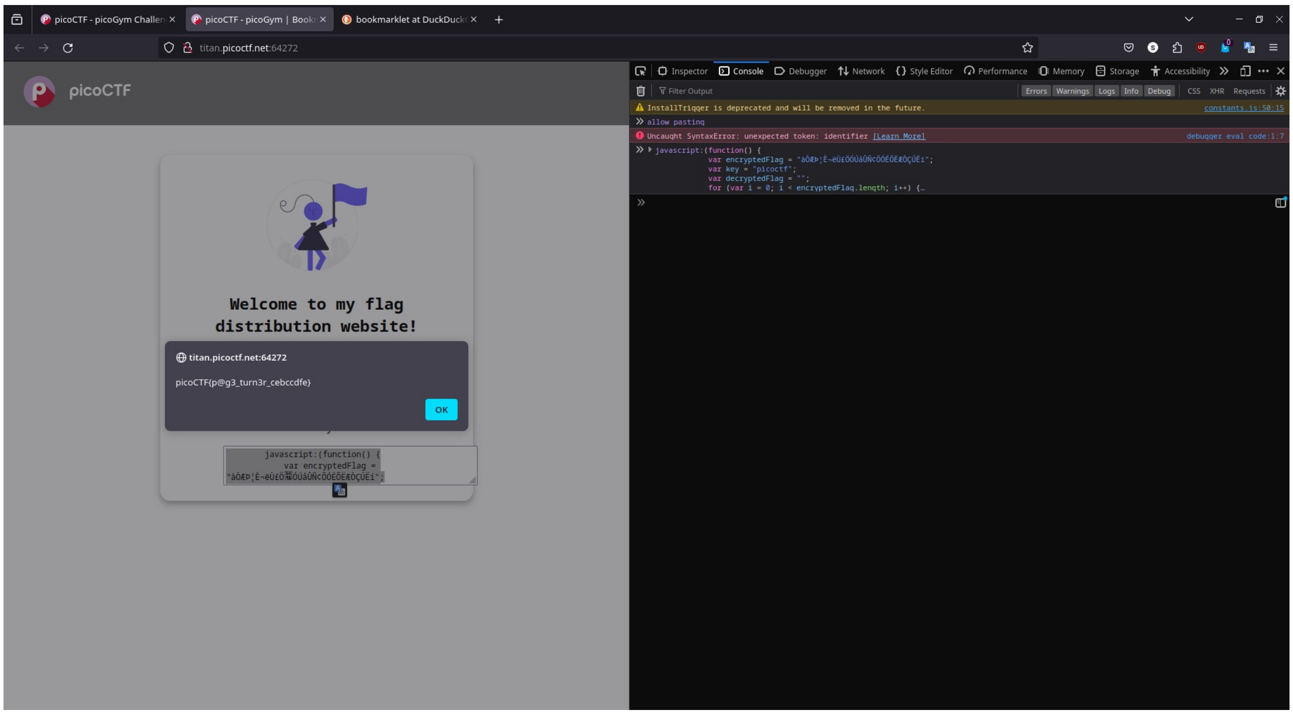
```
# Part 4: 3s_2_lO0k
# I love making websites on my Mac, I can Store a lot of information there.
```

Then, the last part seemed related to Apple devices but I wasn't able to find something related to special files that are used for web configuration or other uses specific to those devices. So all I got of the flag was 'picoCTF{th4ts_4_l0t_0f_pl4c3s_2_lO0k'.

Note: I gave up for this problem and looked for the solution, turns out it was a file related to Apple desktop settings. Not intuitive at all when the problem is related to web exploitation and if you haven’t used a Mac ever (my case). So I completed the flag with that knowledge, but if you think it is unfair, substract the necessary points from my grade.

Bookmarklet

This problem was very easy, the bookmarklet was a javascript function and its execution and printing (by using a browser alert). So, by running it in the inspect page console the alert showed the flag:



Welcome to my flag distribution website!

titan.picoctf.net:64272

picoCTF{p@g3_turn3r_cebccdfe}

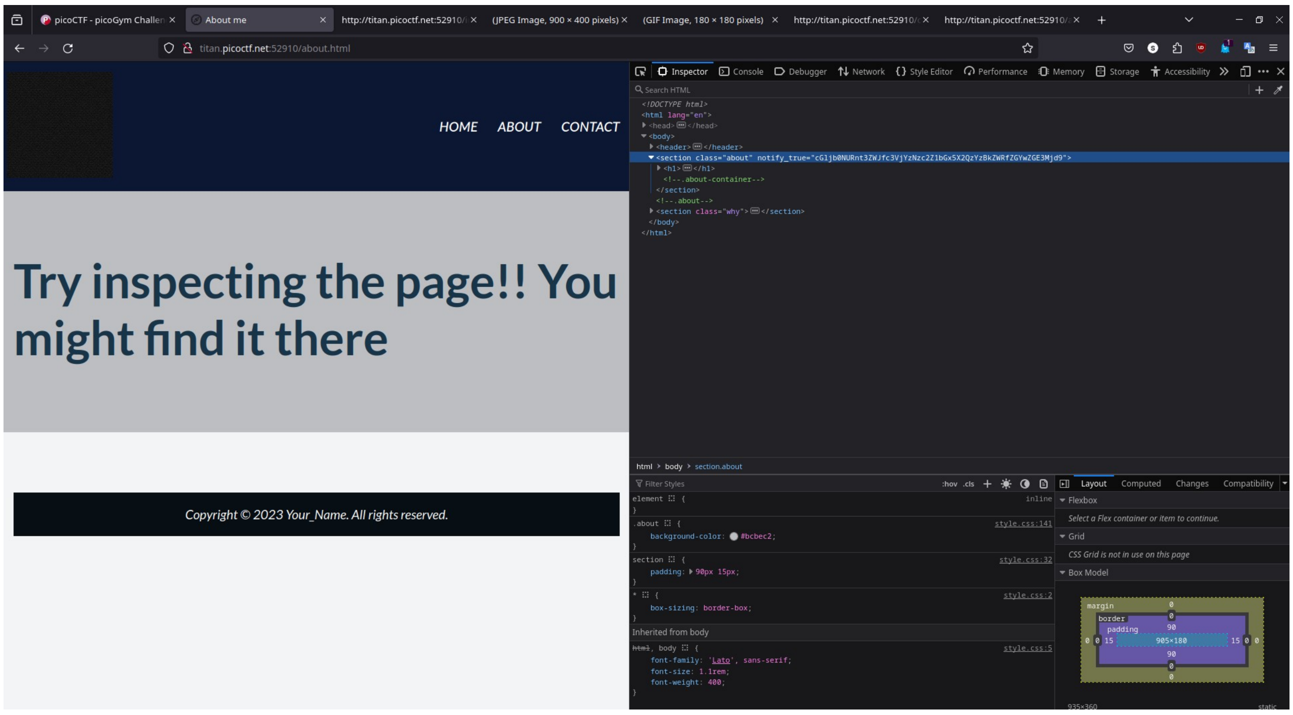
```
InstallTrigger is deprecated and will be removed in the future.      constants.js:50:15
» allow pasting
Uncaught SyntaxError: unexpected token: identifier [Learn More]       debugger eval code:1:7
» ▸ javascript:(function() {
            var encryptedFlag = "àÒÆÞ¦È¬ëÙ£ÖÓÚåÛÑ¢ÕÓÉÕËÆÒÇÚËí";
            var key = "picoctf";
            var decryptedFlag = "";
            for (var i = 0; i < encryptedFlag.length; i++) {…
```

Also the flag was encrypted with a Ceasar cypher because the decription involved converting chars to ASCII I suppose and then substracting a value from the key and using module to ensure the result is in the range of the alphabet:

```
javascript:(function() {
    var encryptedFlag = "àÒÆÞ¦È¬ëÙ£Ö□ÓÚåÛÑ¢ÕÓÉÕËÆÒÇÚËí";
    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);
})();
```

WebDecode

For this problem the hint was to inspect the site. As I didn’t found something in plain sight (html, css, js) I thought that maybe that suspicious gif had the flag hidden. But turns out it was a suspicious string that made no sense:

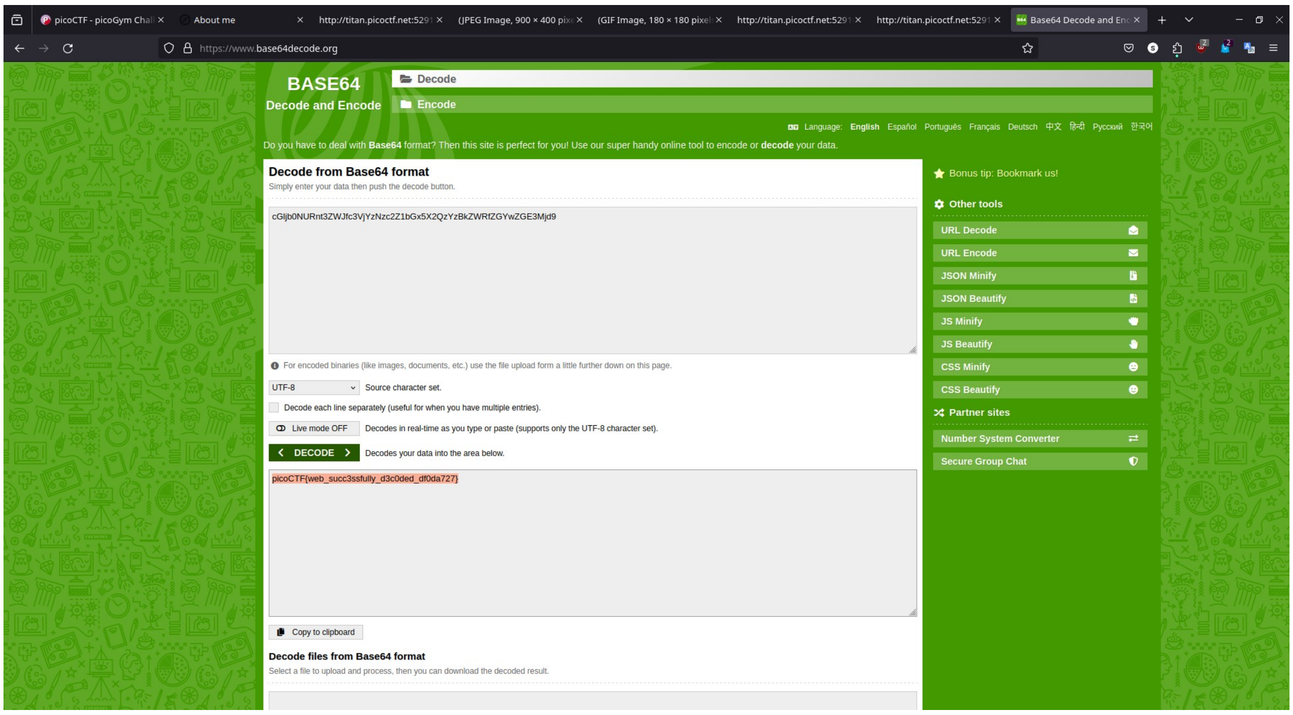

titan.picoctf.net:52910/about.html

Try inspecting the page!! You might find it there

Copyright © 2023 Your_Name. All rights reserved.

```
<!DOCTYPE html>
<html lang="en">
  <head>…</head>
  <body>
    <header>…</header>
    <section class="about" notify_true="cGljb0NURnt3ZWJfc3VjYzNzc2Z1bGx5X2QzYzBkZWRfZGYwZGE3Mjd9">
      <h1>…</h1>
      <!--.about-container-->
    </section>
    <!--.about-->
    <section class="why">…</section>
  </body>
</html>
```

As a test I decoded it in case it was in base 64 and turns out it was the case:


https://www.base64decode.org

Do you have to deal with **Base64** format? Then this site is perfect for you! Use our super handy online tool to encode or **decode** your data.

Decode from Base64 format

Simply enter your data then push the decode button.

```
cGljb0NURnt3ZWJfc3VjYzNzc2Z1bGx5X2QzYzBkZWRfZGYwZGE3Mjd9
```

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

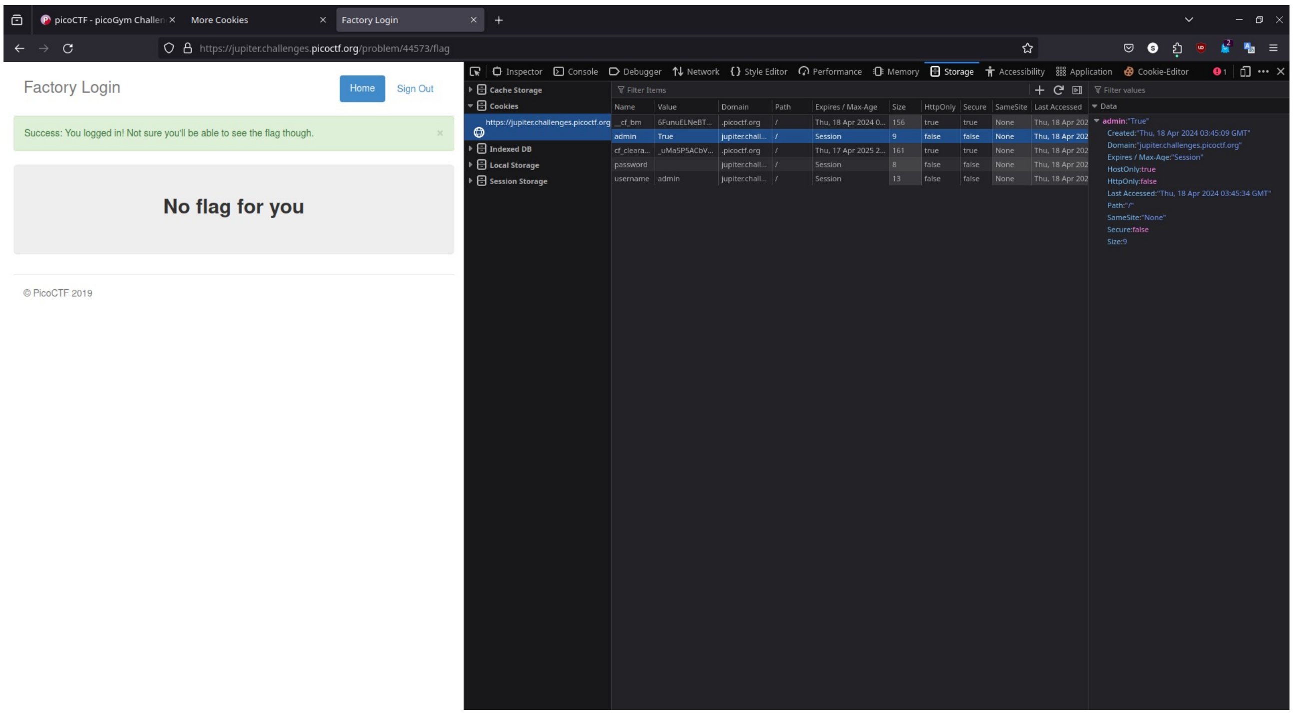
```
picoCTF{web_succ3ssfully_d3c0ded_df0da727}
```

Decode files from Base64 format

Select a file to upload and process, then you can download the decoded result.

logon

For this problem, a login page is presented. I tried to make a sql injection on the form for the user 'joe' without succes. However after reading the hint I tried used admin without a password and it logged it. After that I noticed that there was a cookie named admin (not related to the user because there was a 'name' one also) with value False, so I changed it to True:



https://jupiter.challenges.picoctf.org/problem/44573/flag

Factory Login

Success: You logged in! Not sure you'll be able to see the flag though.

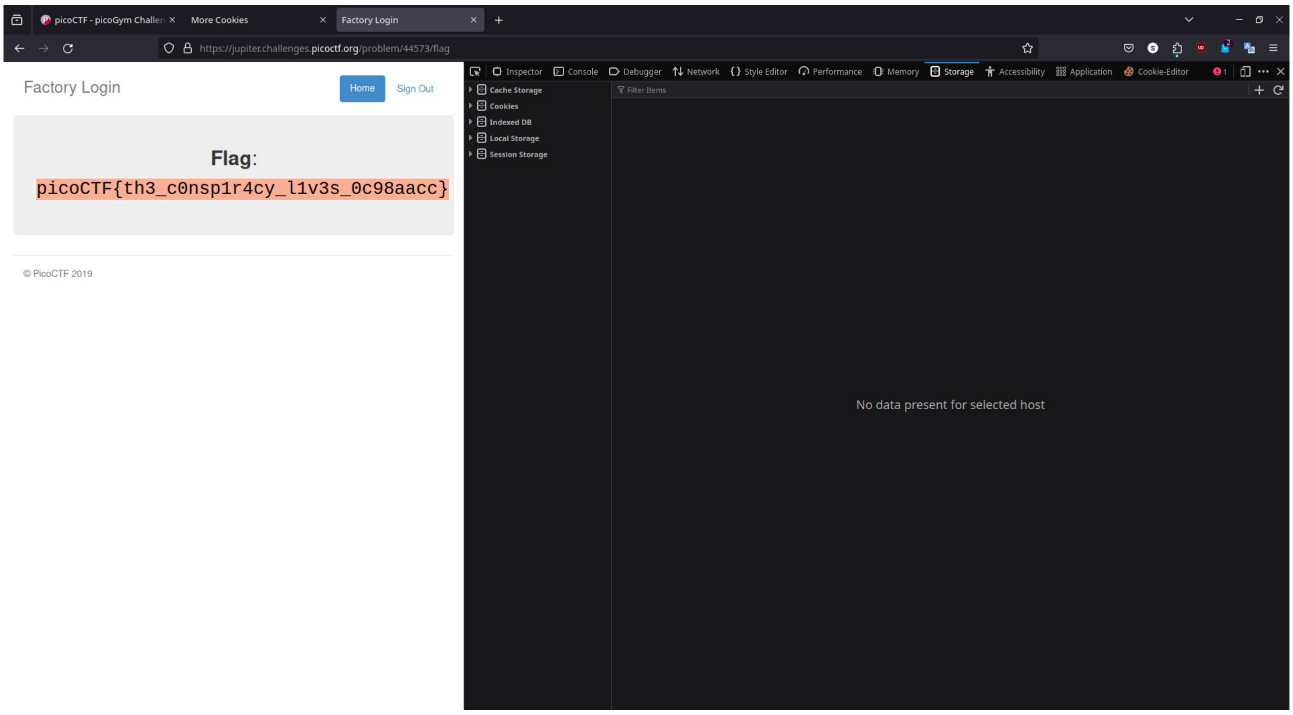
No flag for you

© PicoCTF 2019

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
__cf_bm	6FunuELNeBT...	.picoctf.org	/	Thu, 18 Apr 2024 0...	156	true	true	None	Thu, 18 Apr 202
admin	True	jupiter.chall...	/	Session	9	false	false	None	Thu, 18 Apr 202
cf_cleara...	_uMa5P5ACbV...	.picoctf.org	/	Thu, 17 Apr 2025 2...	161	true	true	None	Thu, 18 Apr 202
password		jupiter.chall...	/	Session	8	false	false	None	Thu, 18 Apr 202
username	admin	jupiter.chall...	/	Session	13	false	false	None	Thu, 18 Apr 202

```
admin:"True"
  Created:"Thu, 18 Apr 2024 03:45:09 GMT"
  Domain:"jupiter.challenges.picoctf.org"
  Expires / Max-Age:"Session"
  HostOnly:true
  HttpOnly:false
  Last Accessed:"Thu, 18 Apr 2024 03:45:34 GMT"
  Path:"/"
  SameSite:"None"
  Secure:false
  Size:9
```

After reloading, instead of the 'No flag' message, the flag appeared:



https://jupiter.challenges.picoctf.org/problem/44573/flag

Factory Login

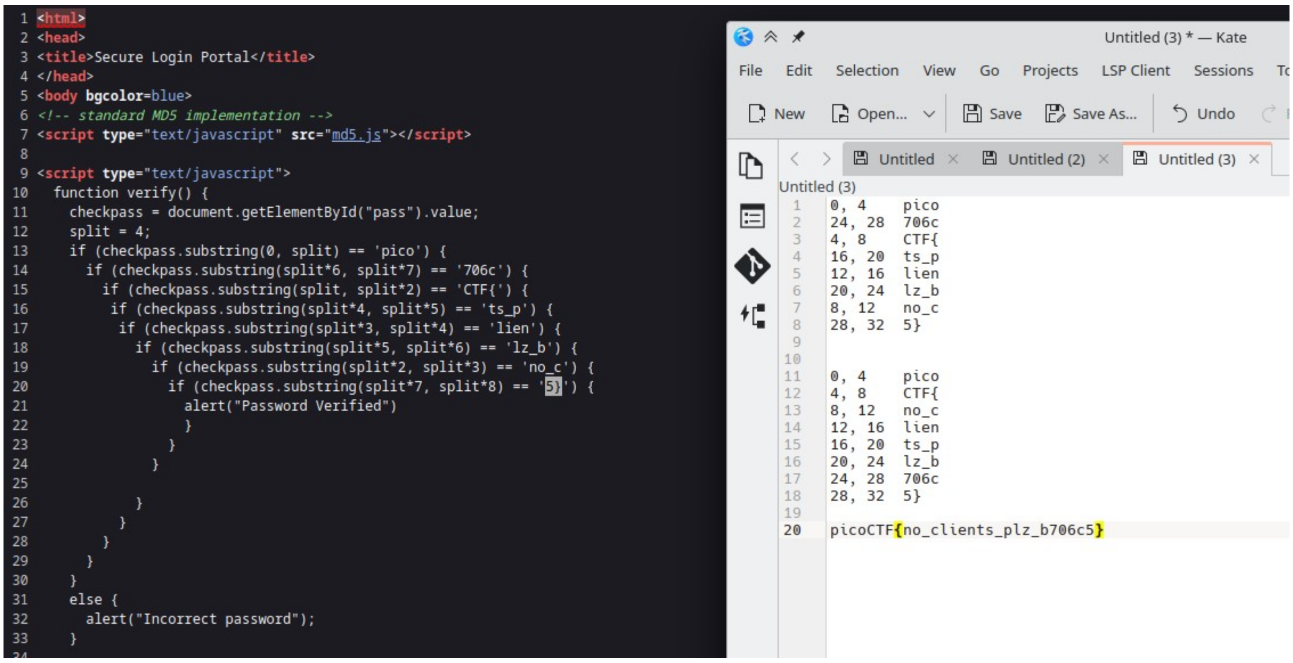
Flag: picoCTF{th3_c0nsp1r4cy_l1v3s_0c98aacc}

© PicoCTF 2019

No data present for selected host

dont-use-client-side

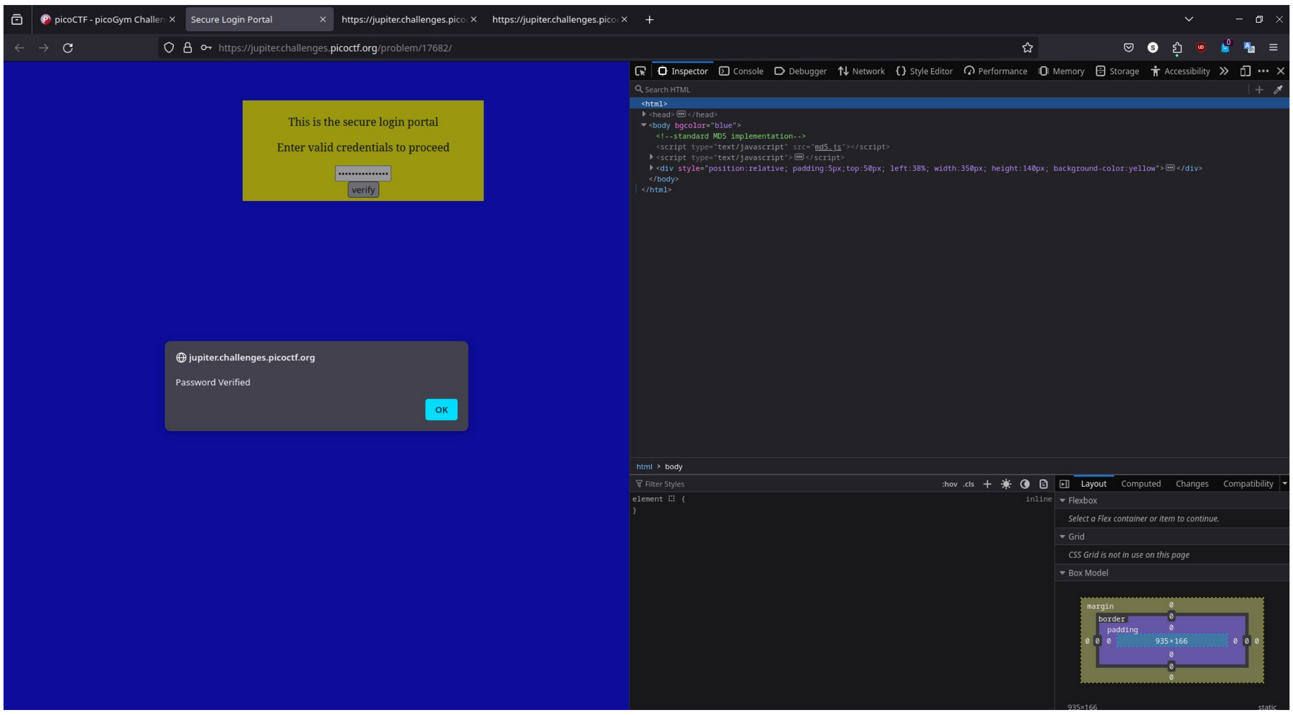
For this problem we are presented a login portal, and by inspecting the source code we see that the verification function is inplain sight. It just checks the password in a unordered way, so we only need to reorder the parts in the correct order to get the password that happens to be the flag:



```
<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) == '706c') {
        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_b') {
              if (checkpass.substring(split*2, split*3) == 'no_c') {
                if (checkpass.substring(split*7, split*8) == '5}') {
                  alert("Password Verified")
                  }
                }
              }
      
            }
          }
        }
      }
    }
    else {
      alert("Incorrect password");
    }
```

```
Untitled (3) * — Kate
Untitled (3)
0, 4    pico
24, 28  706c
4, 8    CTF{
16, 20  ts_p
12, 16  lien
20, 24  lz_b
8, 12   no_c
28, 32  5}


0, 4    pico
4, 8    CTF{
8, 12   no_c
12, 16  lien
16, 20  ts_p
20, 24  lz_b
24, 28  706c
28, 32  5}

picoCTF{no_clients_plz_b706c5}
```

Also, here’s the validation we get after using that string:



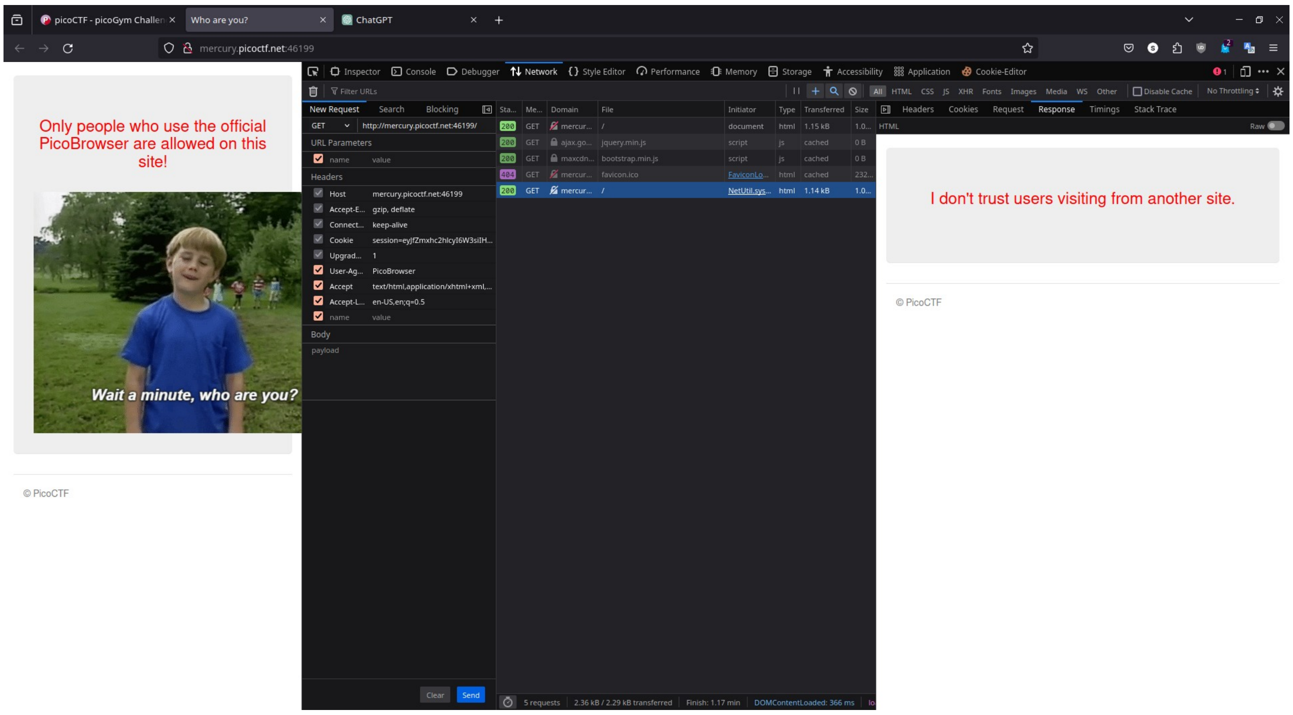
```
https://jupiter.challenges.picoctf.org/problem/17682/

This is the secure login portal
Enter valid credentials to proceed

jupiter.challenges.picoctf.org
Password Verified
```

```
<html>
  <head>…</head>
  <body bgcolor="blue">
    <!--standard MD5 implementation-->
    <script type="text/javascript" src="md5.js"></script>
    <script type="text/javascript">…</script>
    <div style="position:relative; padding:5px;top:50px; left:38%; width:350px; height:140px; background-color:yellow">…</div>
  </body>
</html>
```

Who are you?

Note: I used firefox again because its inspect tools allows you to modify and resend http requests.

First the site tells you only people using PicoBrowser can access the site, so we have to spoof this. Turns out the http requests headers carry all kinds of information including this. So in firefox we select the original request, modify the user-agent (browser and resend it):



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

© PicoCTF

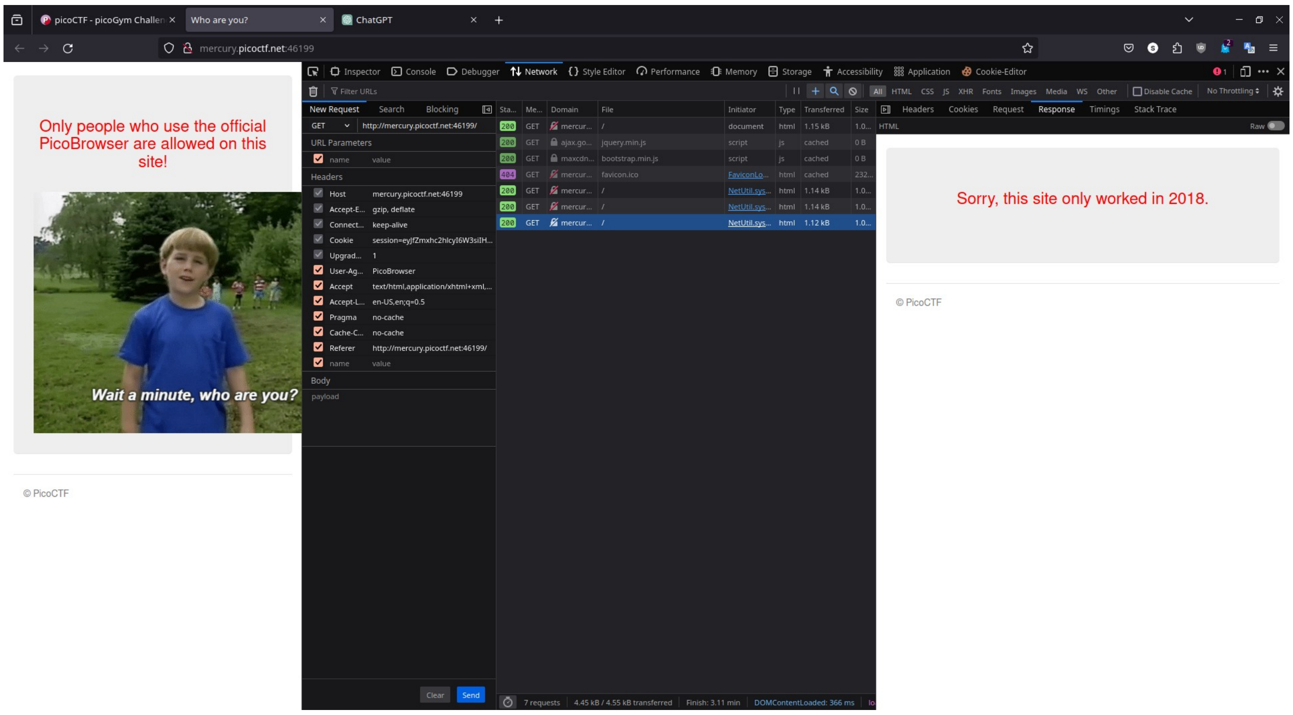
```
GET  http://mercury.picoctf.net:46199/

Headers
Host          mercury.picoctf.net:46199
Accept-E...   gzip, deflate
Connect...    keep-alive
Cookie        session=eyJfZmxhc2hlcyI6W3siIH...
Upgrad...     1
User-Ag...    PicoBrowser
Accept        text/html,application/xhtml+xml,...
Accept-L...   en-US,en;q=0.5
```

```
Sta...  Me...  Domain      File              Initiator     Type  Transferred  Size
200     GET    mercur...   /                 document      html  1.15 kB      1.0...
200     GET    ajax.go...  jquery.min.js     script        js    cached       0 B
200     GET    maxcdn...   bootstrap.min.js  script        js    cached       0 B
404     GET    mercur...   favicon.ico       FaviconLo...  html  cached       232...
200     GET    mercur...   /                 NetUtil.sys...html  1.14 kB      1.0...

5 requests   2.36 kB / 2.29 kB transferred   Finish: 1.17 min   DOMContentLoaded: 366 ms
```

I don't trust users visiting from another site.

© PicoCTF

After that, we have to ensure that the requests includes the origin or something similar, so analizyng the other requests I found there is a referer one:



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

© PicoCTF

```
GET  http://mercury.picoctf.net:46199/

Headers
Host          mercury.picoctf.net:46199
Accept-E...   gzip, deflate
Connect...    keep-alive
Cookie        session=eyJfZmxhc2hlcyI6W3siIH...
Upgrad...     1
User-Ag...    PicoBrowser
Accept        text/html,application/xhtml+xml,...
Accept-L...   en-US,en;q=0.5
Pragma        no-cache
Cache-C...    no-cache
Referer       http://mercury.picoctf.net:46199/
```

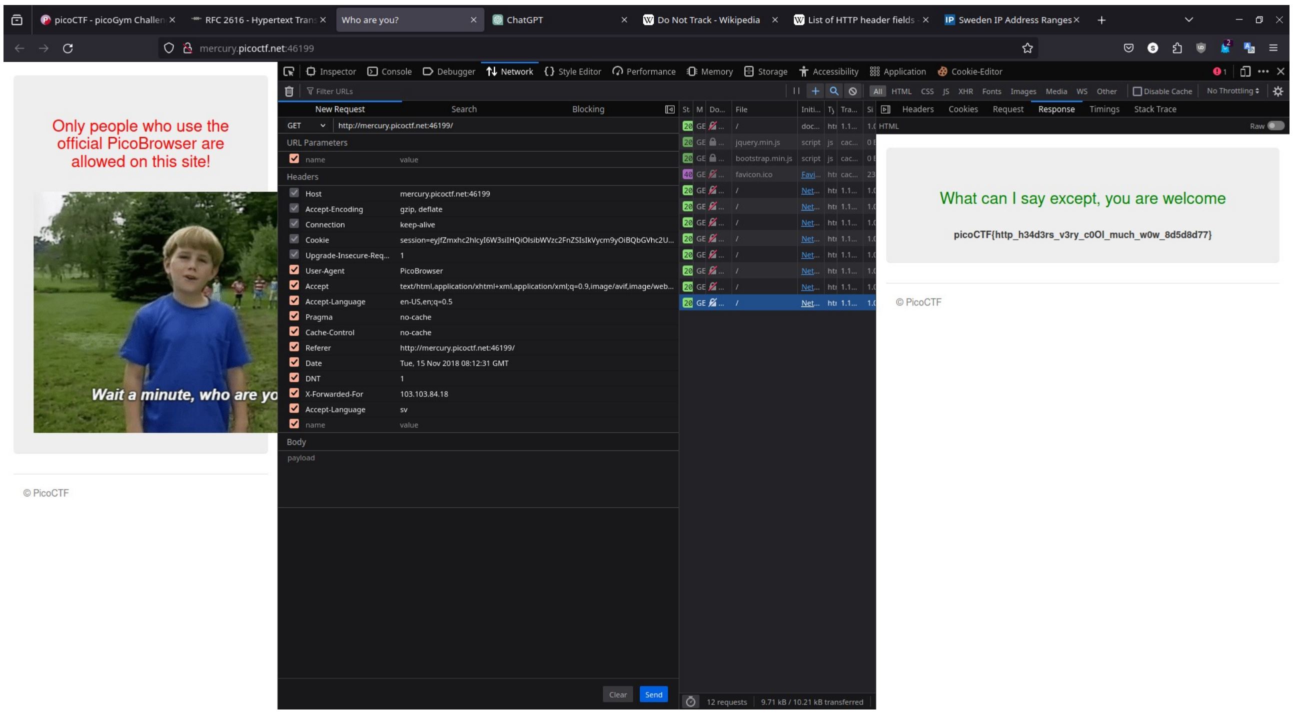
```
Sta...  Me...  Domain      File              Initiator     Type  Transferred  Size
200     GET    mercur...   /                 document      html  1.15 kB      1.0...
200     GET    ajax.go...  jquery.min.js     script        js    cached       0 B
200     GET    maxcdn...   bootstrap.min.js  script        js    cached       0 B
404     GET    mercur...   favicon.ico       FaviconLo...  html  cached       232...
200     GET    mercur...   /                 NetUtil.sys...html  1.14 kB      1.0...
200     GET    mercur...   /                 NetUtil.sys...html  1.14 kB      1.0...
200     GET    mercur...   /                 NetUtil.sys...html  1.12 kB      1.0...

7 requests   4.45 kB / 4.55 kB transferred   Finish: 3.11 min   DOMContentLoaded: 366 ms
```

Sorry, this site only worked in 2018.

© PicoCTF

From here, the procedure was the same: find the appropiate header and modify its value. I used this wikipedia page to find the usefull ones: https://en.wikipedia.org/wiki/List_of_HTTP_header_fields

And at the end this are all the headers I needed to modify:


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

© PicoCTF

```
GET   http://mercury.picoctf.net:46199/
URL Parameters
  name                        value
Headers
  Host                        mercury.picoctf.net:46199
  Accept-Encoding             gzip, deflate
  Connection                  keep-alive
  Cookie                      session=eyJfZmxhc2hlcyI6W3siIHQiOlsibWVzc2FnZSIsIkVycm9yOiBQbGVhc2U...
  Upgrade-Insecure-Req...     1
  User-Agent                  PicoBrowser
  Accept                      text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web...
  Accept-Language             en-US,en;q=0.5
  Pragma                      no-cache
  Cache-Control               no-cache
  Referer                     http://mercury.picoctf.net:46199/
  Date                        Tue, 15 Nov 2018 08:12:31 GMT
  DNT                         1
  X-Forwarded-For             103.103.84.18
  Accept-Language             sv
  name                        value
Body
  payload
```

12 requests 9.71 kB / 10.21 kB transferred

What can I say except, you are welcome

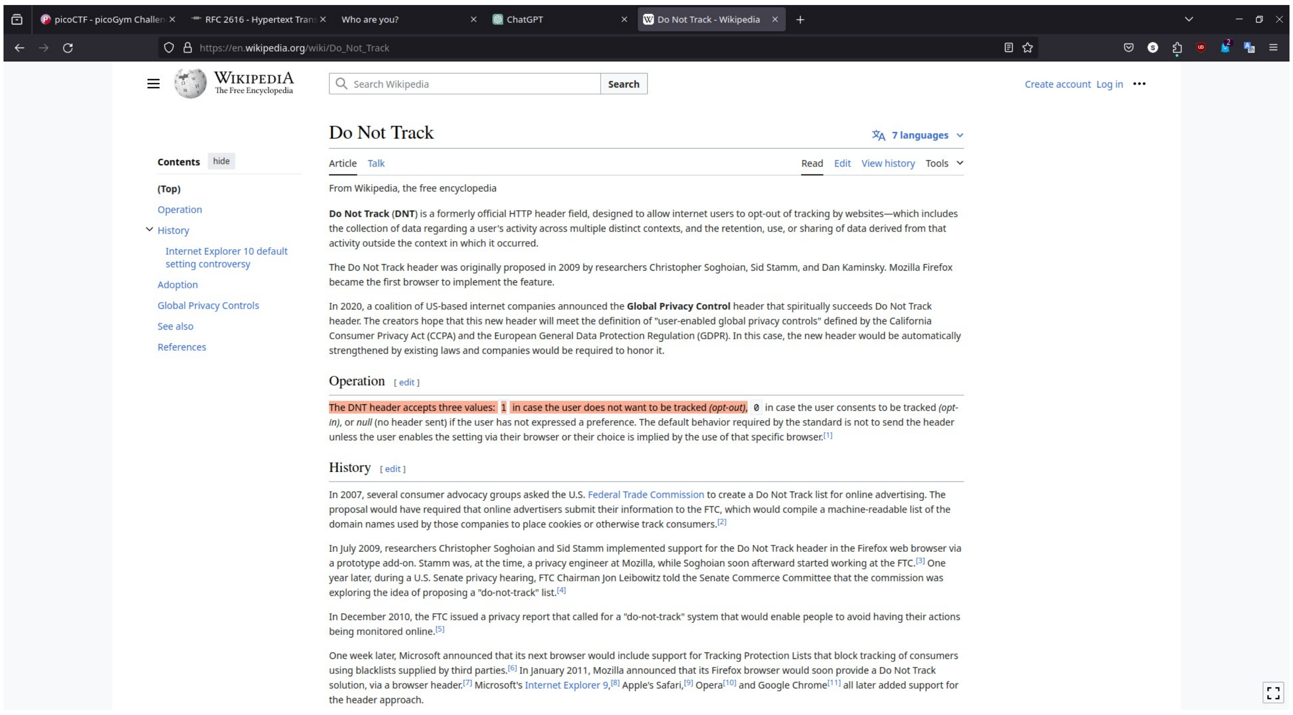
picoCTF{http_h34d3rs_v3ry_c0Ol_much_w0w_8d5d8d77}

© PicoCTF

The explanation for the headers:

Date: date, used the example from the hint site

DNT: Do Not Track, 1 equals opt-out


https://en.wikipedia.org/wiki/Do_Not_Track

Do Not Track

From Wikipedia, the free encyclopedia

Do Not Track (**DNT**) is a formerly official HTTP header field, designed to allow internet users to opt-out of tracking by websites—which includes the collection of data regarding a user's activity across multiple distinct contexts, and the retention, use, or sharing of data derived from that activity outside the context in which it occurred.

The Do Not Track header was originally proposed in 2009 by researchers Christopher Soghoian, Sid Stamm, and Dan Kaminsky. Mozilla Firefox became the first browser to implement the feature.

In 2020, a coalition of US-based internet companies announced the **Global Privacy Control** header that spiritually succeeds Do Not Track header. The creators hope that this new header will meet the definition of "user-enabled global privacy controls" defined by the California Consumer Privacy Act (CCPA) and the European General Data Protection Regulation (GDPR). In this case, the new header would be automatically strengthened by existing laws and companies would be required to honor it.

Operation

The DNT header accepts three values: 1 in case the user does not want to be tracked *(opt-out)*, 0 in case the user consents to be tracked *(opt-in)*, or *null* (no header sent) if the user has not expressed a preference. The default behavior required by the standard is not to send the header unless the user enables the setting via their browser or their choice is implied by the use of that specific browser.[1]

History

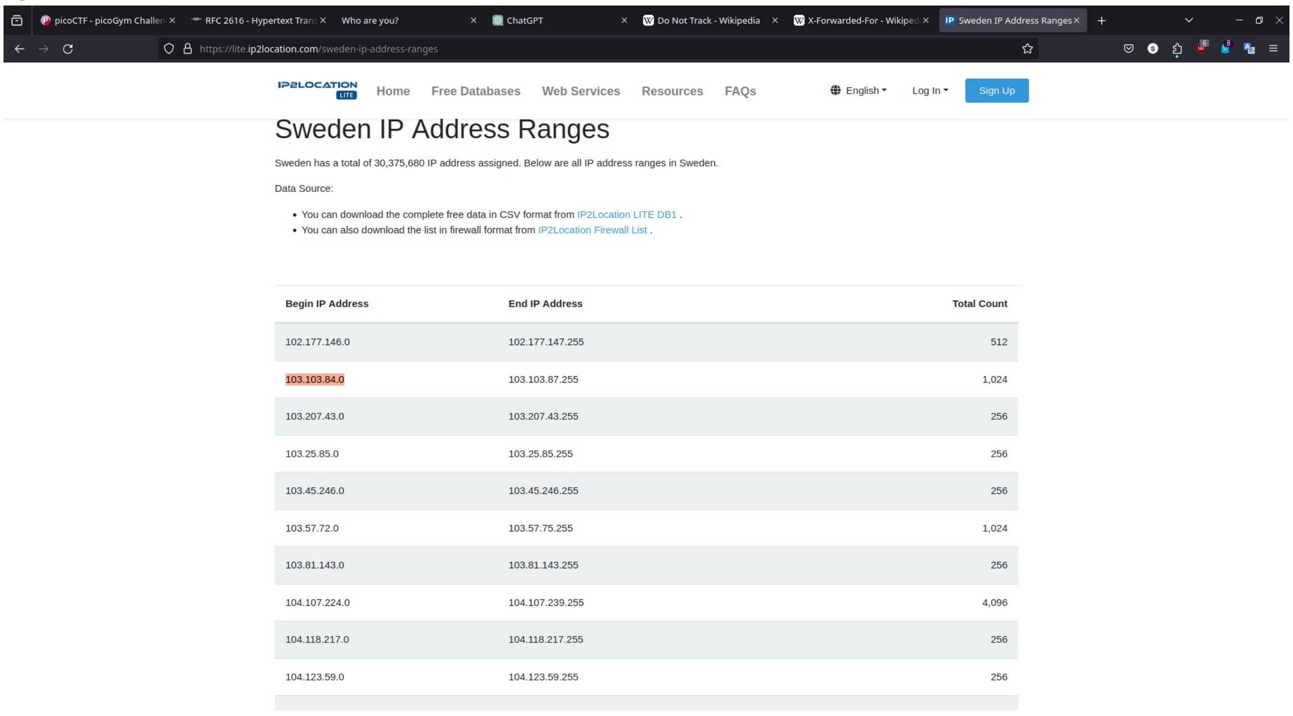
In 2007, several consumer advocacy groups asked the U.S. Federal Trade Commission to create a Do Not Track list for online advertising. The proposal would have required that online advertisers submit their information to the FTC, which would compile a machine-readable list of the domain names used by those companies to place cookies or otherwise track consumers.[2]

In July 2009, researchers Christopher Soghoian and Sid Stamm implemented support for the Do Not Track header in the Firefox web browser via a prototype add-on. Stamm was, at the time, a privacy engineer at Mozilla, while Soghoian soon afterward started working at the FTC.[3] One year later, during a U.S. Senate privacy hearing, FTC Chairman Jon Leibowitz told the Senate Commerce Committee that the commission was exploring the idea of proposing a "do-not-track" list.[4]

In December 2010, the FTC issued a privacy report that called for a "do-not-track" system that would enable people to avoid having their actions being monitored online.[5]

One week later, Microsoft announced that its next browser would include support for Tracking Protection Lists that block tracking of consumers using blacklists supplied by third parties.[6] In January 2011, Mozilla announced that its Firefox browser would soon provide a Do Not Track solution, via a browser header.[7] Microsoft's Internet Explorer 9,[8] Apple's Safari,[9] Opera[10] and Google Chrome[11] all later added support for the header approach.

X-Forwarded-For: accepts an IP and from that it recognizes where the request is coming from (geolocation), so used a sweden one



Sweden IP Address Ranges

Sweden has a total of 30,375,680 IP address assigned. Below are all IP address ranges in Sweden.

Data Source:

- You can download the complete free data in CSV format from IP2Location LITE DB1 .
- You can also download the list in firewall format from IP2Location Firewall List .

Begin IP Address	End IP Address	Total Count
102.177.146.0	102.177.147.255	512
103.103.84.0	103.103.87.255	1,024
103.207.43.0	103.207.43.255	256
103.25.85.0	103.25.85.255	256
103.45.246.0	103.45.246.255	256
103.57.72.0	103.57.75.255	1,024
103.81.143.0	103.81.143.255	256
104.107.224.0	104.107.239.255	4,096
104.118.217.0	104.118.217.255	256
104.123.59.0	104.123.59.255	256

Accept-Language: Takes the language we want the response to be in. Swedish is ‘sv’