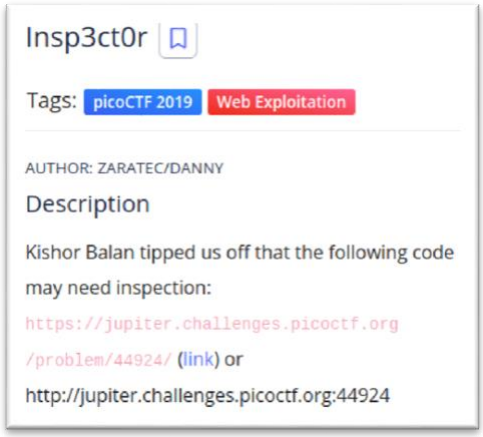
Web Exploitation

1



Solution

Looking at the website, it says HTML, CSS and JS have been used to make the website. Hence the flag is stored in parts in all the 3 components.

```
<button class="tablink" onclick="openTab('tababout', this, '#222')"
style="background-color: rgb(34, 34, 34);">How</button> event
<div id="tabintro" class="tabcontent" style="display: none;">…</div>
<div id="tababout" class="tabcontent" style="display: block;">
  <h3>How</h3>
  <p>…</p>
  <!-- Html is neat. Anyways have 1/3 of the flag: picoCTF{tru3 d3 -->
</div>
</div>
</body>
</html>
```

Inspecting the HTML, we come across the first part of the flag.

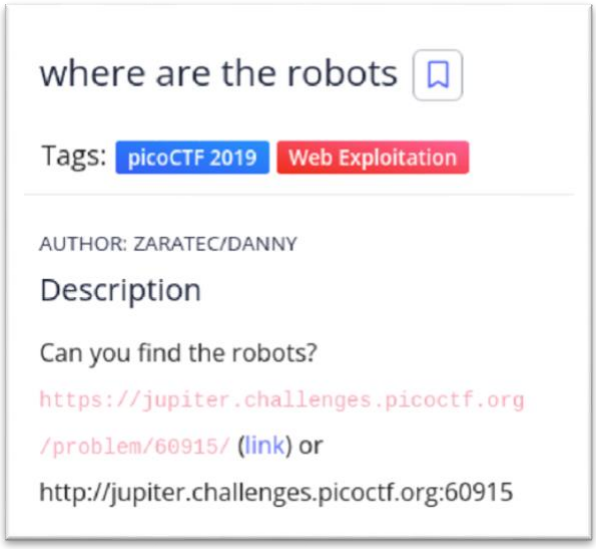
```
.tabcontent {
    color: #111;
    display: none;
    padding: 50px;
    text-align: center;
}

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

/* You need CSS to make pretty pages. Here's part 2/3 of the flag: t3ctlve_0r_ju5t */
```

Looking at the CSS, we come across the second part of the flag.

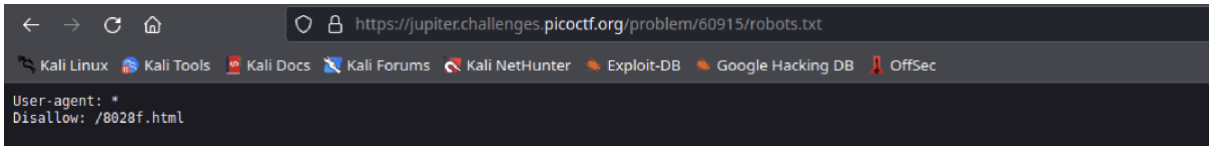
```
    if(elmnt.style != null) {
    elmnt.style.backgroundColor = color;
    }
}

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

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

The JS has the third part of the flag. Putting all the 3 together, we have the final flag.

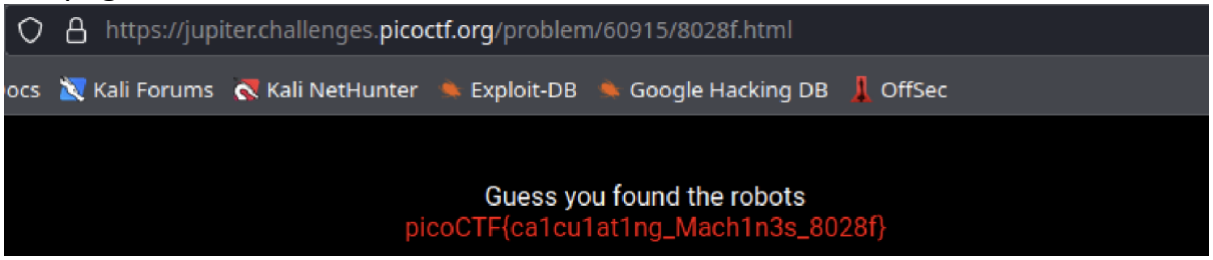
2.



Solution

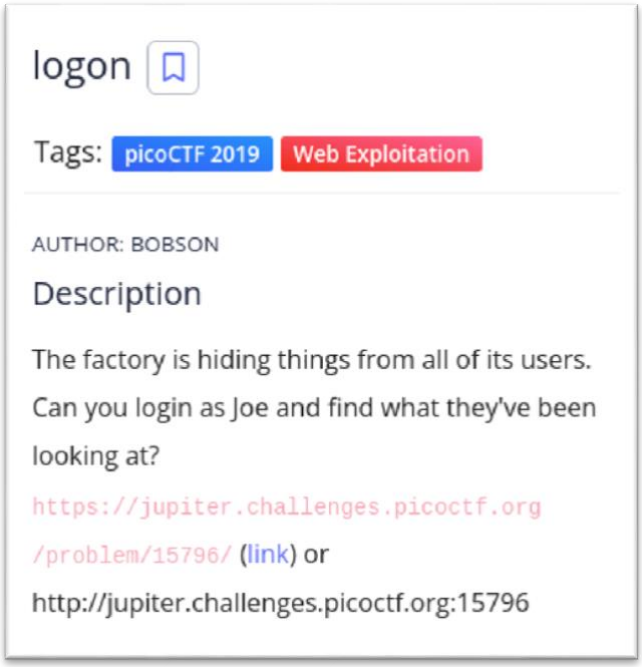


Accessing the robots.txt page using the URL, gives us a disallowed section of the page.

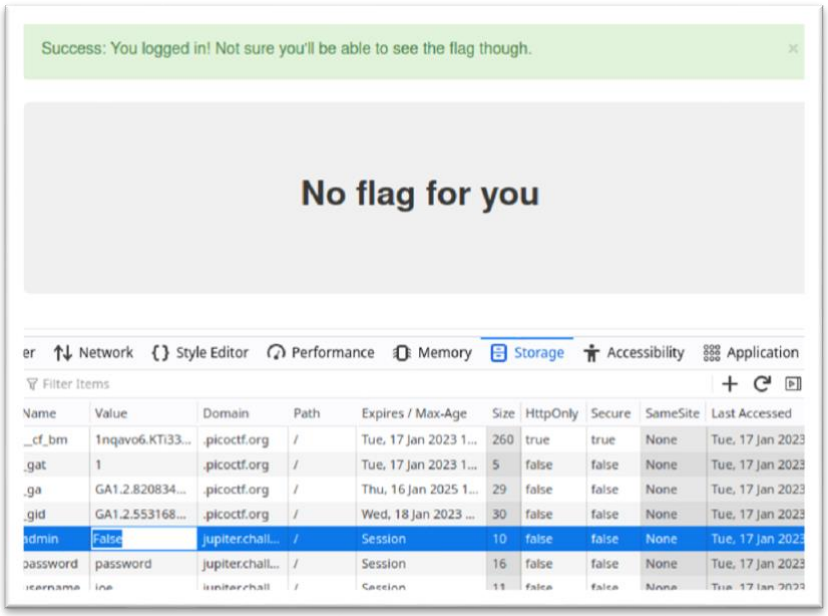


Accessing that page, we find the flag.

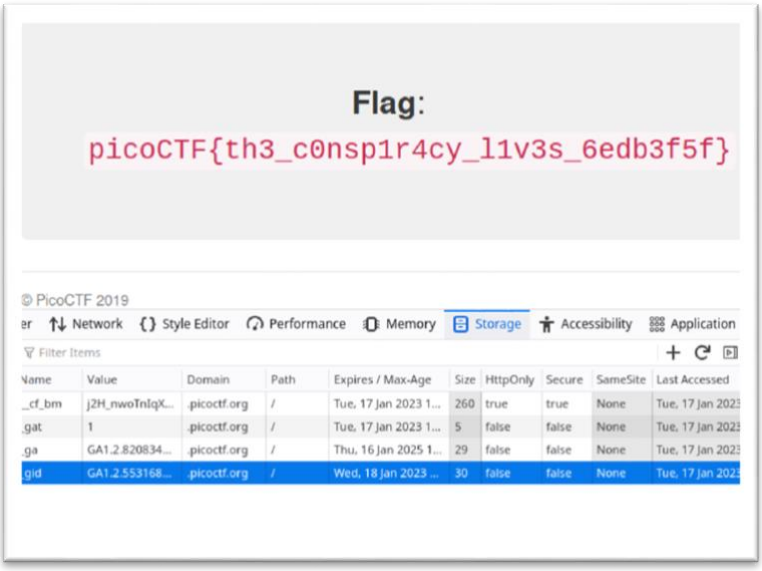
3.



Solution

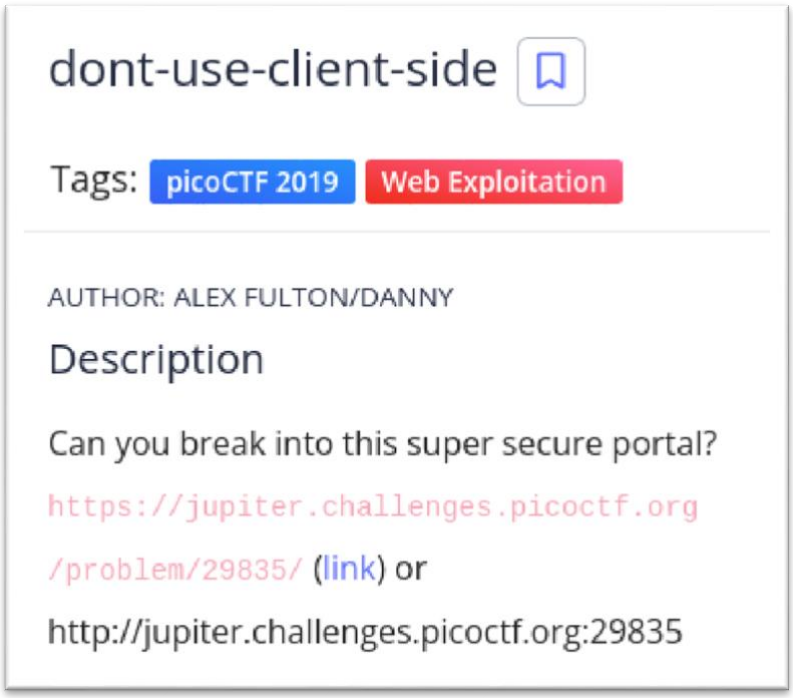


After trying to login as Joe with a random password, we get this admin declared as false.



Toggling it to true from false and then reloading the page, we find the password.

4.

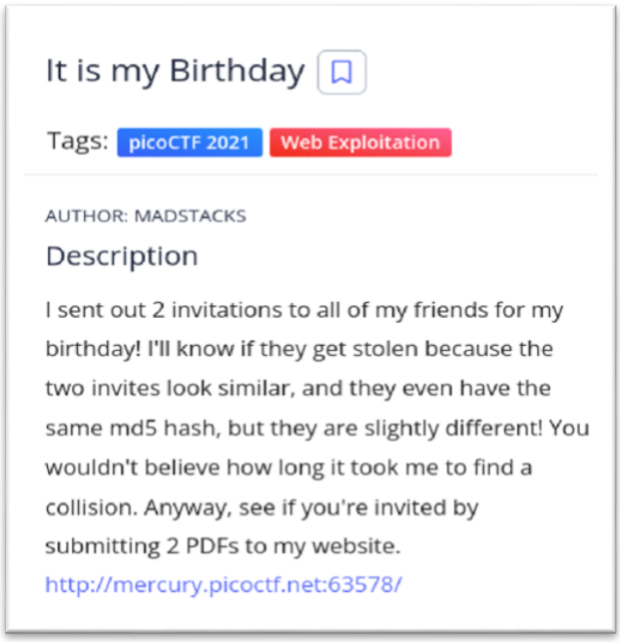


Solution

```
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")
```

The JavaScript of the page reveals the flag in a jumbled order. Rearranging it, we arrive at the flag.

5.



Solution

```
// FLAG: picoCTF{c0ngr4ts_u_r_1nv1t3d_5c8c5ce2}
```

Submitting 2 pdf's taken from the website - https://www.mathstat.dal.ca/~selinger/md5collision/ we get the flag displayed on the page.

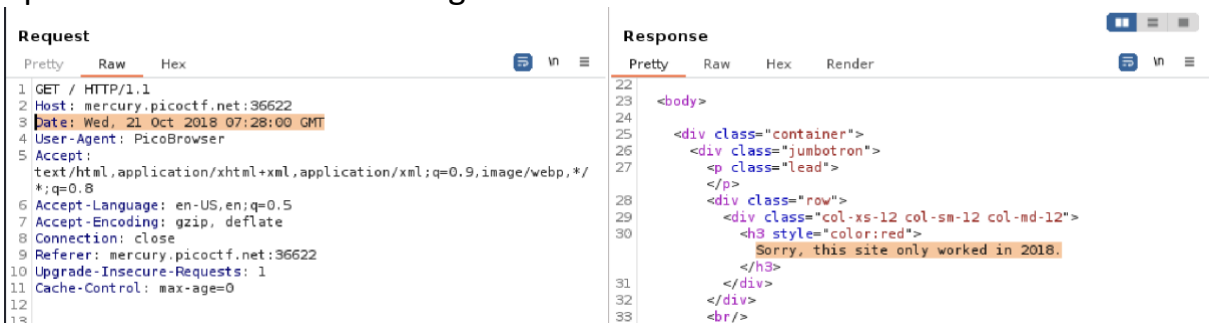
6.

Solution

Using burpsuite's repeater to change the credentials as needed.

The website shows that only people who use PicoBrowser are allowed in here. So changing the User-Agent to PicoBrowser.

The website now says it doesn't trust visitors from other sites. Hence we update the referrer to the original site name.



The website says that the site has worked only in 2018. So we update the date accordingly.

As the site says it doesn't trust users who can be tracked, we update DNT to 1

As the website is only for Sweden, a Swiss IP address is used as the generating IP.

The accept language must be changed from English to Swedish for this to work.

And finally, we are presented with the flag.

7.

picobrowser

Tags: picoCTF 2019 | Web Exploitation

AUTHOR: ARCHIT

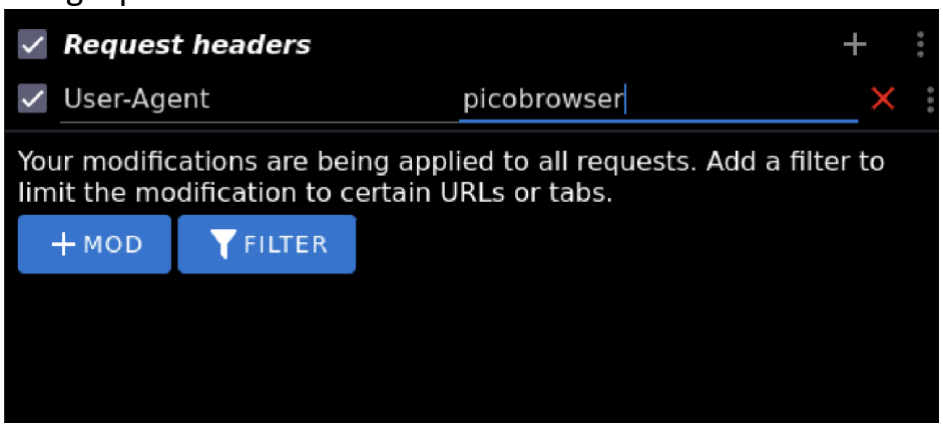
Description

This website can be rendered only by **picobrowser**, go and catch the flag!

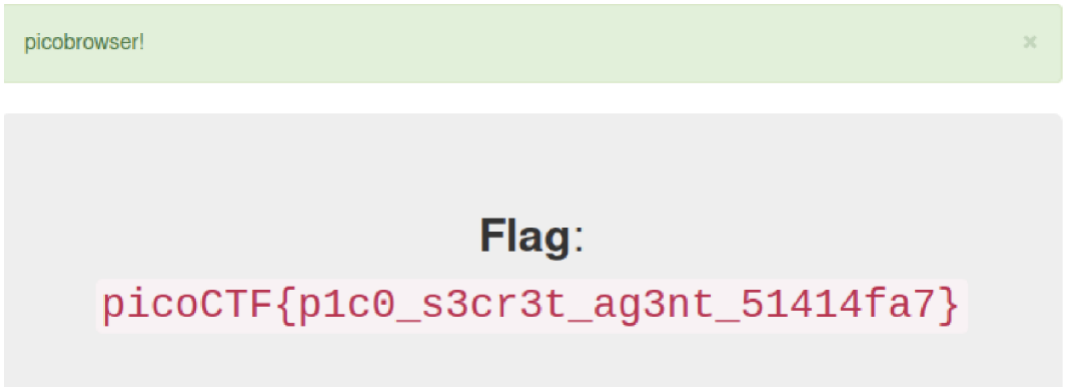
https://jupiter.challenges.picoctf.org/problem/50522/ (link) or http://jupiter.challenges.picoctf.org:50522

Solution

If we try to access the flag using the browser, it doesn't allow as we are not using a picobrowser.



Using an extension called modheader to change the user-agent to picobrowser



The flag is now displayed.