### **Title : dont-use-client-side**

**Category : Web Exploitation**

**Description:**

 Can you break into this super secure portal? <https://jupiter.challenges.picoctf.org/problem/37821/> or [link](https://jupiter.challenges.picoctf.org/problem/37821/)

**Hints:**

* Never trust the client

**Points:** 100

**Source Code**

Here’s the source code for the index.html page:

<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) == 'a3c8') {

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\_1') {

if (checkpass.substring(split\*2, split\*3) == 'no\_c') {

if (checkpass.substring(split\*7, split\*8) == '9}') {

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>

For this challenge we only need to focus on the code within the “script” tags. The JavaScript calls a function called verify() to check the password we enter against the correct password. There’s a bit of code obfuscation going on with the checkpass.substring() functions, but nothing we can’t handle.

Essentially, the correct password is split at different locations to make reading it a bit trickier. We could manually read the password flag, but it’s much more fun to write a script!

**Solution**

Here’s a janky bash script I wrote that (sort of) prints the flag:

#!/usr/bin/bash

grep "substring" index.html | sed 's/ //g' | sort | cut -d "'" -f 2

I used sed to strip the leading whitespace, and sort to roughly place the flag pieces in the correct order.

And the output:

$ ./get-flag.sh

pico

no\_c

lien

ts\_p

lz\_1

a3c8

9}

CTF{

Flag: picoCTF{no\_clients\_plz\_1\_a3c89}

Title : Insp3ct0r

**Category : Web Exploitation**

Points: 50

Description:

Kishor Balan tipped us off that the following code may need inspection:

Link: https://jupiter.challenges.picoctf.org/problem/41511/

Hints

### **Solution:**

View source code

The first flag is at the bottom of the index.html file

<!-- Html is neat. Anyways have 1/3 of the flag: \*\*\* -->

The second flag is at the bottom of the mycss.css file

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

The third flag is at the bottom of the myjs.js file

/\* Javascript sure is neat. Anyways part 3/3 of the flag: \*\*\* \*/

Result :

The flag is the combination of the 3 parts.

Title: where are the robots

**Category : Web Exploitation**

Points: 100

Description:

Can you find the robots?

Link: <https://jupiter.challenges.picoctf.org/problem/56830/>

Hints

### **Solution**

Open source page and go to /robots.txt

https://jupiter.challenges.picoctf.org/problem/36474/robots.txt

Look for the Disallow: /1bb4c.html

Replace “/robots.txt” with “/1bb4c.html” in url

Then you will get the CTF Flag

DONE