Guilherme Alt Chagas Merklein

PicoCTF 2024 - Web Exploitation

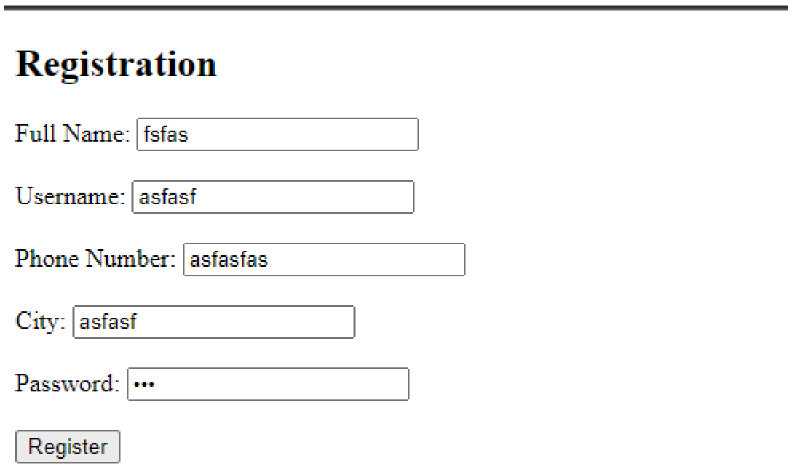
flag: picoCTF{#0TP_Bypvss_SuCc3$S_c94b61ac}

26/03/2024

IntroToBurp

Idk if it was supposed to be like this or I just got lucky.

The challenge starts with a simple web page with a form that lets you submit information about you and choose a password (image 1). At first I just entered random things on the forms, and after submitting I got redirected to a /dashboard endpoint, which was asking for an OTP. I just entered "OTP" but it refused, and I couldn't get it to accept my OTP.



Registration

Full Name:	fsfas
Username:	asfasf
Phone Number:	asfasfas
City:	asfasf
Password:	•••

Image 1: webpage a form to submit.

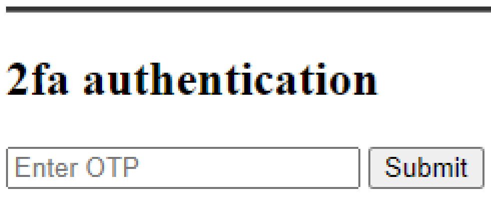
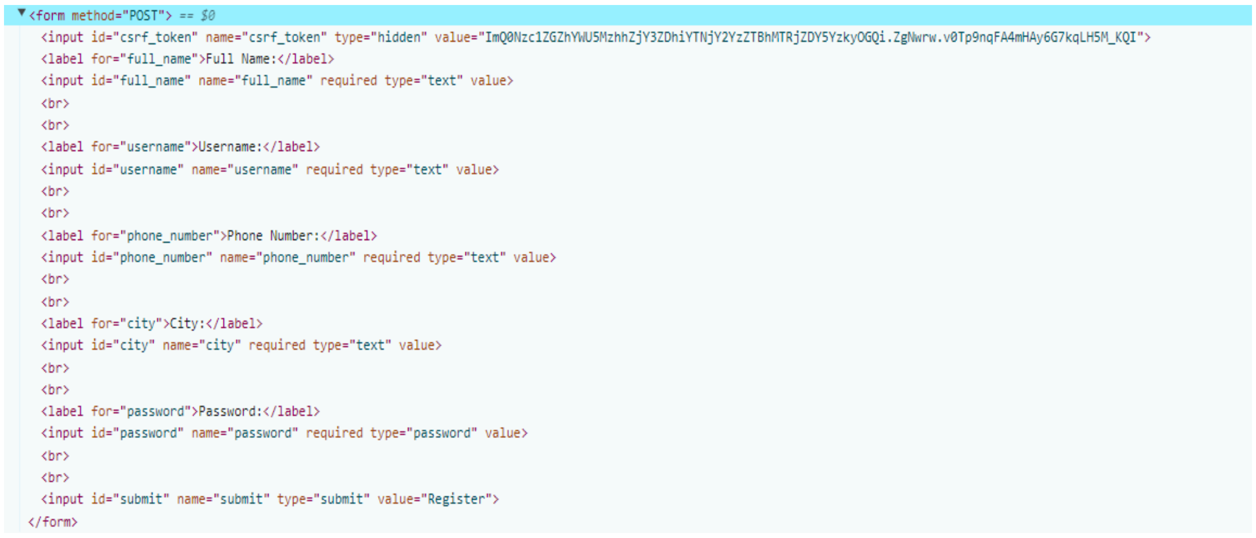


Image 2: Webpage at endpoint /dashboard, asking for OTP.

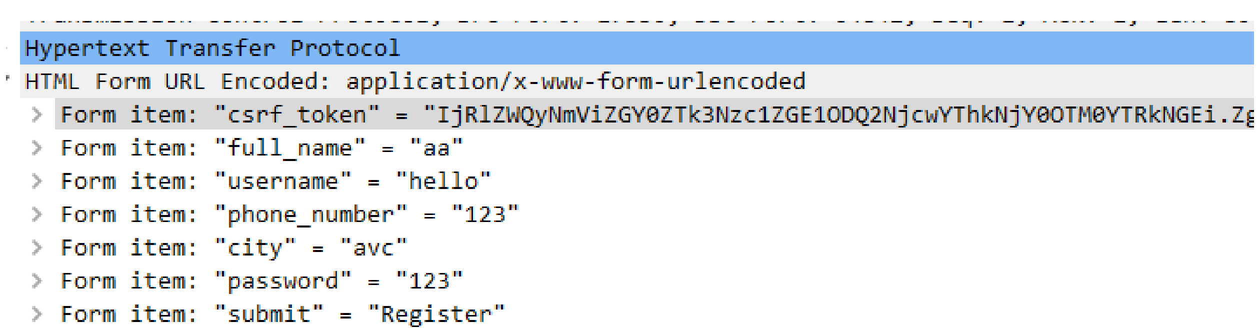
Next step, as usual, was inspecting the elements. I went to the forms page and found something interesting: there were some hidden inputs that were being sent along with the register request (image 3). Also the data format for the HTTP POST request is form-urlencoded.

I started wireshark to capture those packets and see what data was being transfered, and indeed those fields were observed (image 4).



```
▼<form method="POST"> == $0
   <input id="csrf_token" name="csrf_token" type="hidden" value="ImQ0Nzc1ZGZhYWU5MzhhZjY3ZDhiYTNjY2YzZTBhMTRjZDY5YzkyOGQi.ZgNwrw.v0Tp9nqFA4mHAy6G7kqLH5M_KQI">
   <label for="full_name">Full Name:</label>
   <input id="full_name" name="full_name" required type="text" value>
   <br>
   <br>
   <label for="username">Username:</label>
   <input id="username" name="username" required type="text" value>
   <br>
   <br>
   <label for="phone_number">Phone Number:</label>
   <input id="phone_number" name="phone_number" required type="text" value>
   <br>
   <br>
   <label for="city">City:</label>
   <input id="city" name="city" required type="text" value>
   <br>
   <br>
   <label for="password">Password:</label>
   <input id="password" name="password" required type="password" value>
   <br>
   <br>
   <input id="submit" name="submit" type="submit" value="Register">
 </form>
```

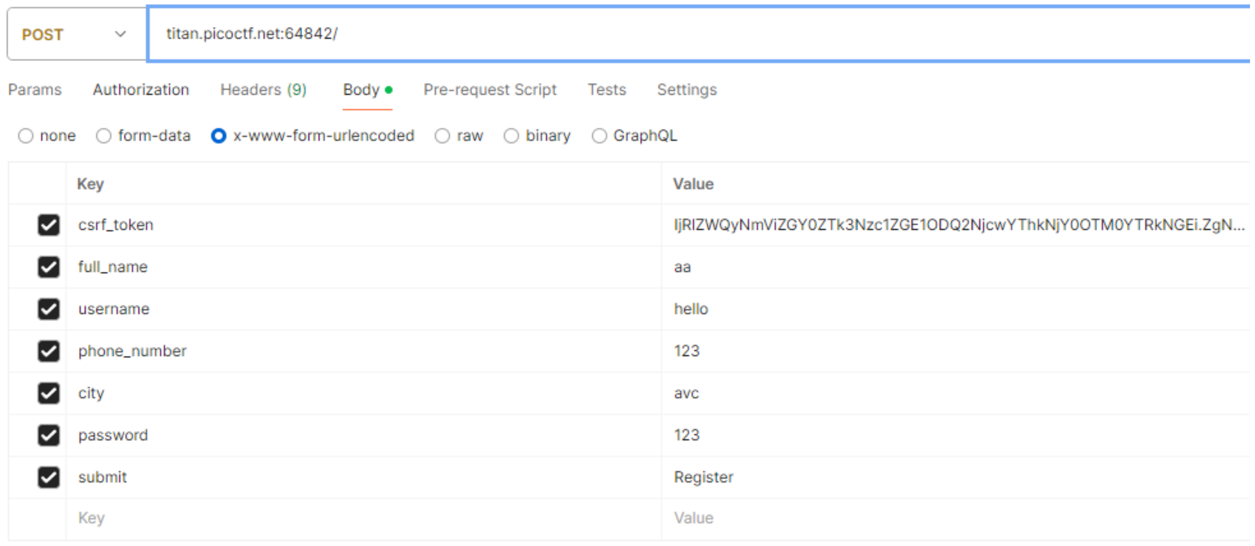
Image 3: Hidden inputs on the forms.



```
Hypertext Transfer Protocol
HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "csrf_token" = "IjRlZWQyNmViZGY0ZTk3Nzc1ZGE1ODQ2NjcwYThkNjY0OTM0YTRkNGEi.Zg
  > Form item: "full_name" = "aa"
  > Form item: "username" = "hello"
  > Form item: "phone_number" = "123"
  > Form item: "city" = "avc"
  > Form item: "password" = "123"
  > Form item: "submit" = "Register"
```

Image 4: Wireshark captured packet.

First thing I did after this was try to see if csrf_token was base-64 encoded, but that was not it. I then remembered Postman had the form-urlencoded data format, so I started Postman and tried to replicate the HTTP POST request that the webpage sends on submit (Image 5). I sent it and it worked, but I didn't know what to do next, so I got *lucky*.

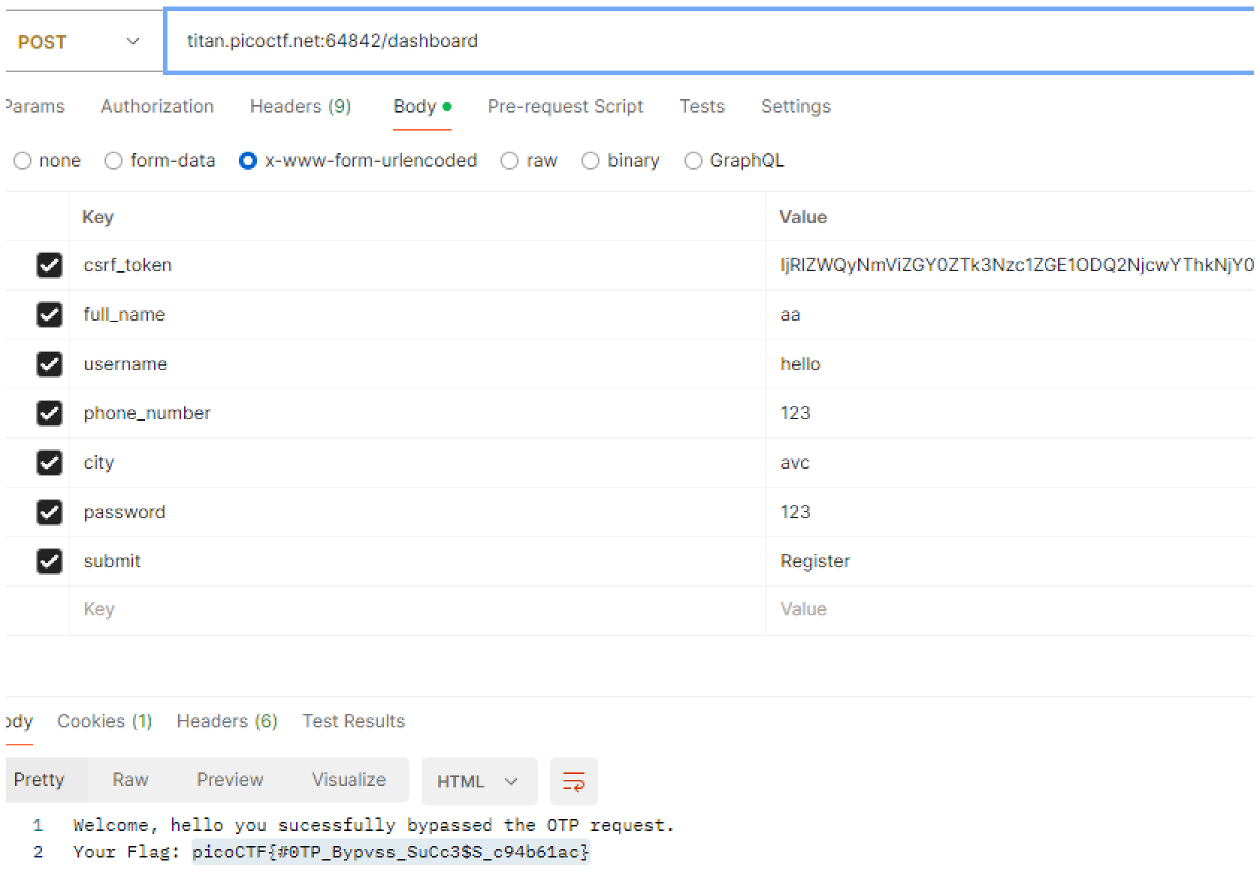


POST titan.picoctf.net:64842/

	Key	Value
✓	csrf_token	IjRlZWQyNmViZGY0ZTk3Nzc1ZGE1ODQ2NjcwYThkNjY0OTM0YTRkNGEi.ZgN...
✓	full_name	aa
✓	username	hello
✓	phone_number	123
✓	city	avc
✓	password	123
✓	submit	Register
	Key	Value

Image 5: Replicated request via Postman.

For some reason I tried to send the exact same request on the endpoint /dashboard, with the exact same fields. The response I got surprised me (Image 6).



POST titan.picoctf.net:64842/dashboard

Params | Authorization | Headers (9) | Body ● | Pre-request Script | Tests | Settings

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key	Value
✓	csrf_token	IjRlZWQyNmViZGY0ZTk3Nzc1ZGE1ODQ2NjcwYThkNjY0
✓	full_name	aa
✓	username	hello
✓	phone_number	123
✓	city	avc
✓	password	123
✓	submit	Register
	Key	Value

Body | Cookies (1) | Headers (6) | Test Results

Pretty | Raw | Preview | Visualize | HTML

```
Welcome, hello you sucessfully bypassed the OTP request.
Your Flag: picoCTF{#0TP_Bypvss_SuCc3$S_c94b61ac}
```

Image 6: Response on /dashboard endpoint.

Good for me, I guess.