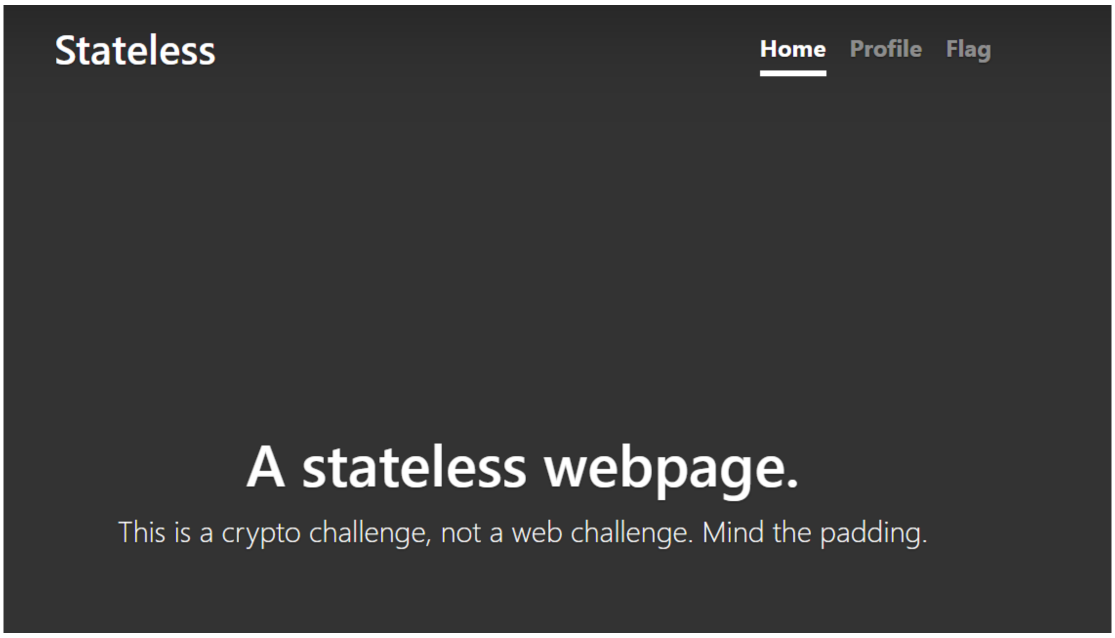
Stateless

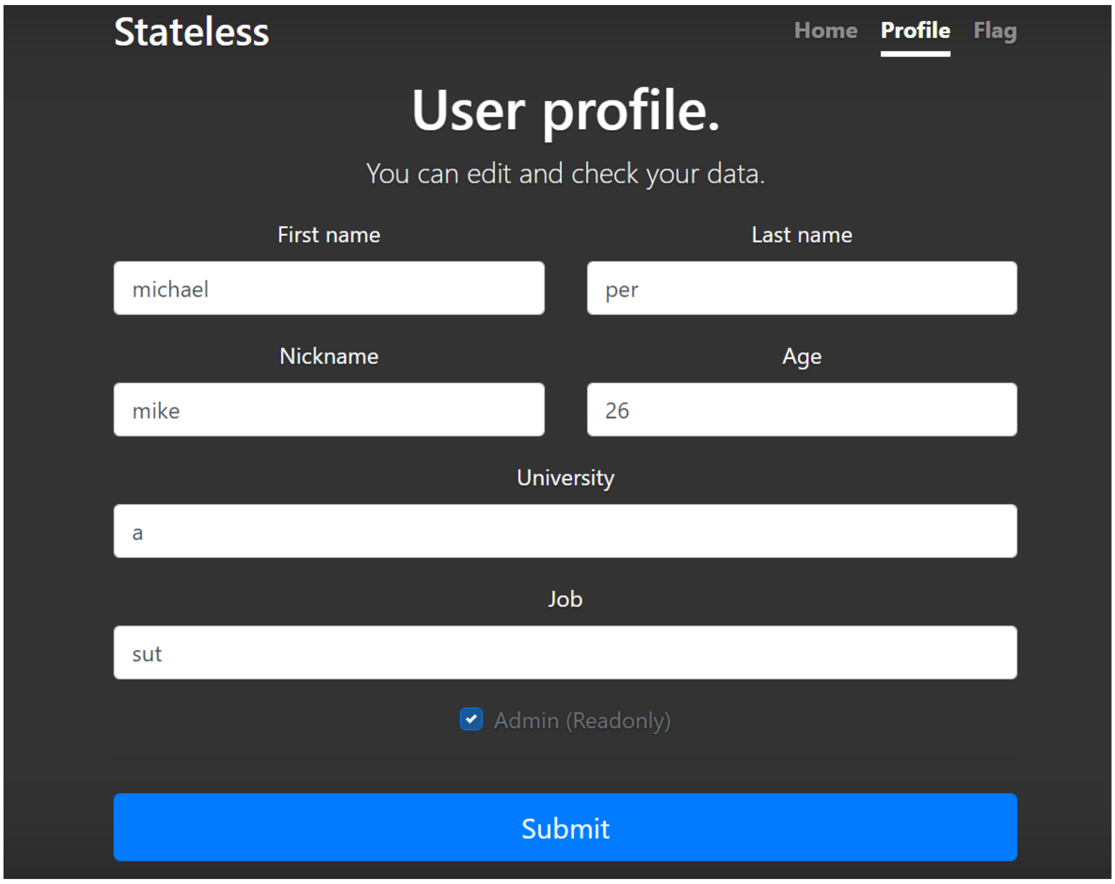
Recon phase:



Stateless

A stateless webpage.

This is a crypto challenge, not a web challenge. Mind the padding.



Stateless

User profile.

You can edit and check your data.

First name: michael

Last name: per

Nickname: mike

Age: 26

University: a

Job: sut

[✓] Admin (Readonly)

Hint – Mind the padding and the challenge is easy.

At this time I have no idea which encryption but I keep it in mind.

There is 3 pages:

Home – show us the hint.

Profile – construct information about me.

Flag - only admins can reach there.

Inside the page there is no authentication phase so, I assume that the authentication came from the cookies.

I investigate the meaning of the term "stateless" and I realized that like JWT the server save piece of data in client side cookie to store the state of the user, in our case: Admin or not.

When I rewrite the Profile with different parameters I saw that my cookie is changed.

Something else that I saw from first look on cookie is that there another encryption with base64 after the main encryption.

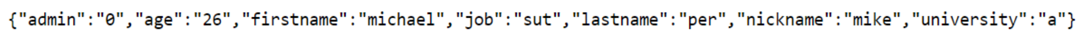
I decide to look more into the network flow to see if I can distinguish something about the data that send from me to the server and I saw that I sent the raw data like this:

"**firstname=michael&lastname=per&nickname=mike&age=26&university=a&job=sut**"

I assumed that the full string is like:

"**firstname=michael&lastname=per&nickname=mike&age=26&university=a&job=sut&admin=0**"

Because when I trying to reach profile page I got:



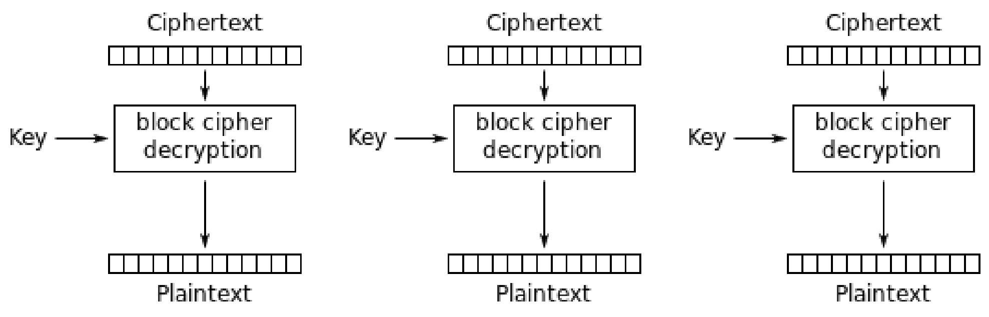
```
{"admin":"0","age":"26","firstname":"michael","job":"sut","lastname":"per","nickname":"mike","university":"a"}
```

Probably the server takes this input and reconstruct some encrypted output.

<u>Weapon phase:</u>

Before I select the right weapon I saw weird pattern that connected to the hint in the beginning, each time that I changed the parameters in the Profile page I got cookie with length that is a multiple of 32.

I search more in the internet and intersect the easy knowledge of the challenge and I came with the idea that the encryption is AEC with mode ECB.



Electronic Codebook (ECB) mode decryption

Now I know that the raw data that I sent parsed to blocks of strings with 32 bytes length.

After playing with AES in ECB mode with my self I sow that if the string have 16 bytes the algo generate automatically more 16 bytes with the same key and padding.

My attack:

All the string length before the &admin=0 will be modulo 16 so the last block plaintext contains only &admin=0 and padding

Before the last block inject &job= AAA…. Contains 2 blocks and another block string that ends with &admin=1 .

Copy the block encryption that ends with &admin=1 and copy paste without the block before and remove the last block.

Then I got the flag.