Checklist

1. Test User Enumeration

a. without account lock

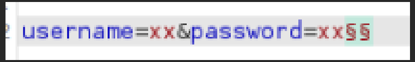
- enumerate a valid user name from usernames wordlist and brute-force this user's password from passwords wordlist

b. via account lock

To understand the account lockout

1. account locks me out based on login attempts

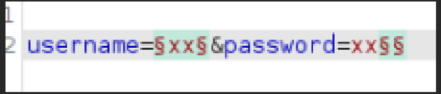
To verify , use the null payload option to generate 50 login attempts. and see if locked acccount or still 'Invalid username or password'



```
username=xx&password=xx§§
```

2. account locks me out based on actual usernames

To verify , try to login by each username with invalid password & the account will be closed to the correct user............................



```
username=§xx§&password=xx§§
```

Attack type: `Cluster Bomb`

- Payload 1: the provided candidate usernames
- Payload 2: Null payloads, 5 times

2. Broken brute-force protection

a. IP block

`I have one account and victim's username` :

- After three unsuccessful attempts, the login gets locked out for a minute

.

`solution` inject the credentials of my account on every third line and then use the Pitchfork attack on the Burp Intruder or Turbo intruder .

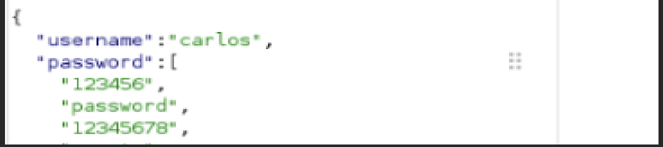
b. Multiple credentials per request

`I have a victim's username`

- After three unsuccessful attempts, the login gets locked out for a minute and noticed that `POST /login` request submits the login `credentials in JSON format`

.

`solution` set a hundred password parameters like this



```
{
  "username":"carlos",
  "password":[
    "123456",
    "password",
    "12345678",
```

by python script i can make any value of passwords in json format

c. Password change inside an account

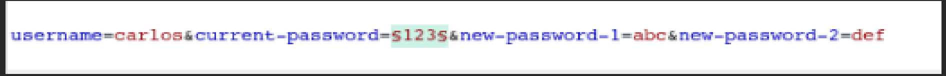
`I have one account and victim's username`

- After a successful login by my account and analyze password change functionality , Notice the behaviour

.

`solution` enter a valid current password, but two different new passwords, the message says New passwords do not match.

- We can use this message to enumerate correct passwords by Grep-Match



```
username=carlos&current-password=§123§&new-password-1=abc&new-password-2=def
```

3. Test 2FA

a. 2FA URL-bypass

`I have two accounts` :

First one : I have access to my email (that's mine)

Second: I don't have access to his email (that's a victim email)

- After a successful login by my account - it redirects to /my-account.

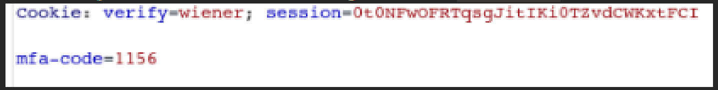
.

`solution`-instead of trying to find the 2FA code, manually change the URL of the victim email to / my-account after the first step of authentication?

b. 2FA broken logic

`I have one account and victim's username` :

- After a successful login by my account - The first interesting thing is that username provided in the `POST request` is reflected back as a cookie. In the request of the security code (`login2`)



```
Cookie: verify=wiener; session=0t0NFWOFRTqsgJitIKi0TZvdCWKxtFCI

mfa-code=1156
```

`solution`- Change verify value that has my username to victim's username

- Brute force the 2FA code by intruder & that happens If `/login2` only verifies the 2FA code without checking preceded by a credential check

c. 2FA brute-forcing bypass

`I have a victim account but I don't have access to his email`

- After analyzing , Brute-forcing the code directly does not work in this case. If I enter it wrong two times, the session appears to be terminated , the CSRF token is invalidated and The application lock the account out

.

`solution` -The login process involves multiple requests so the basic Burp Intruder does not help here.

- use macros and try to combine these requests into a single macro:

4. Test forgot password

Test Password reset functionality.

`I have one account and victim's username`

I will click the forgot password (to reset pass) and write victim's username then i have Three possibilities :

1. click send , intercept the request and and find `full resetlink`

If it was vuln to Excessive Data Exposure

.

2. click send , without intercept and find `check your email` for a reset password link

- Observe that a link containing a unique reset token
- If it was vuln to password reset poisoning

a. Send the POST /forgot-password request to Repeater.

b. Notice that the X-Forwarded-Host header is supported and you can use it to point the dynamically generated reset link to an attacker server.

.

3. click send without intercept and find `validation token`

- i will bruteforce the token by python script or intruder with note times of attempts