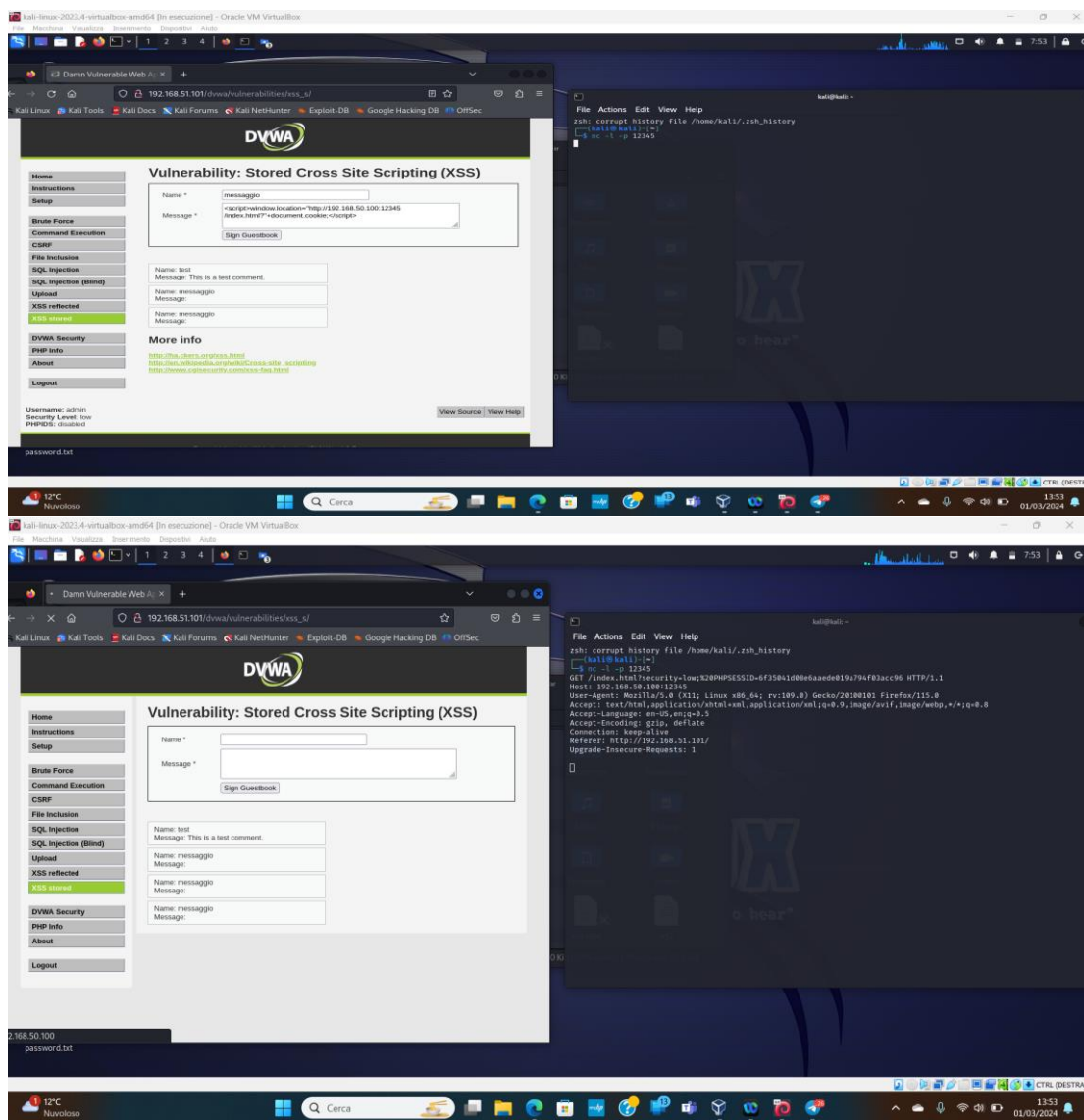


XSS stored

L'attacco XSS stored è una forma di attacco informatico che sfrutta le vulnerabilità nelle applicazioni web per inserire script dannosi nei dati memorizzati sul server e visualizzati agli utenti. Questi script dannosi vengono quindi eseguiti sul browser degli utenti che accedono alla pagina web, consentendo agli attaccanti di rubare informazioni sensibili o danneggiare l'esperienza dell'utente.

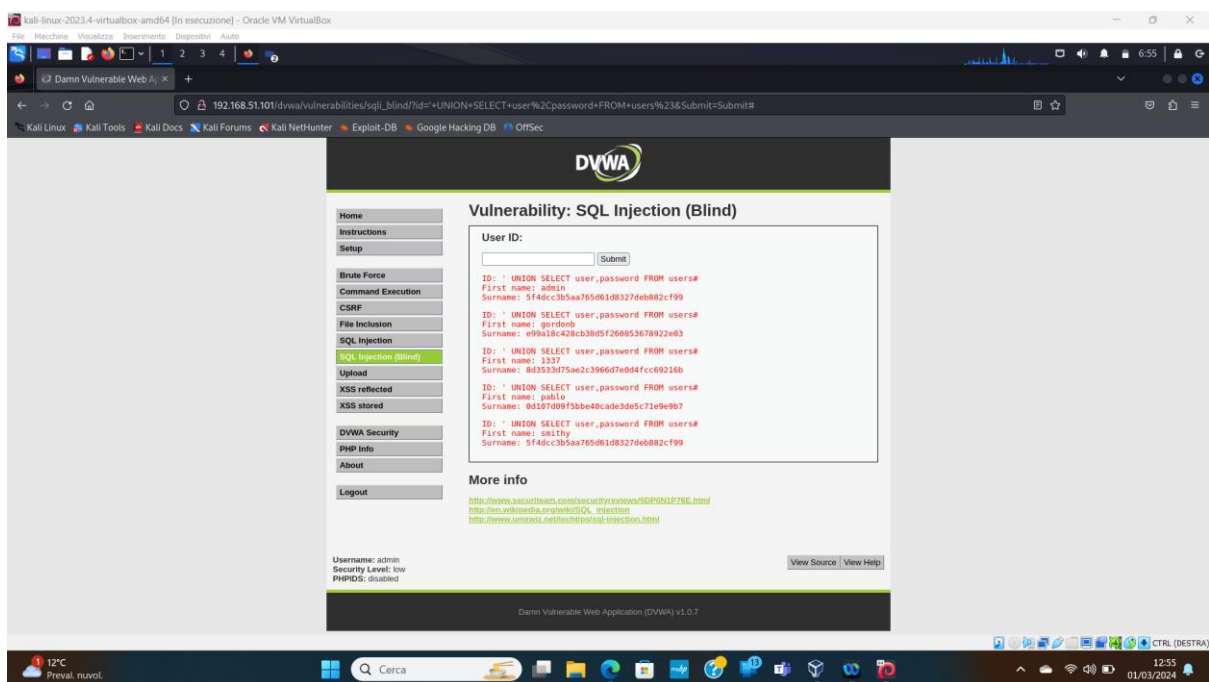
Oggi andremo a inserire uno script (modifichiamo tramite l'ispezione la quantità massima di caratteri da poter inserire) che andrà a rubare i cookie di accesso della vittima indirizzandola al nostro ip e porta in modo tale da usare kali per metterci in ascolto e inserire quei cookie rubati nell'ispezione della pagina bypassando il login.



SQL injection (blind)

Un attacco di SQL injection è una vulnerabilità informatica che consente a un aggressore di inserire codice SQL dannoso all'interno di un'applicazione web o di un'interfaccia utente che interagisce con un database. Questo tipo di attacco sfrutta le falle di sicurezza nelle query SQL, permettendo agli aggressori di manipolare il comportamento del database e ottenere informazioni non autorizzate o eseguire azioni non consentite. Oggi con questo attacco entreremo nel database della web app per cercare di estrapolare tutte le informazioni sensibili che essa ha conservato su questo database, forzandola a farci tornare tutte quelle informazioni a cui noi non dovremmo avere accesso.

Utilizziamo la union per unire due query, dopo aver fatto vari test siamo riusciti a capire il numero di colonne da far combaciare, e scrivendo ' UNION SELECT user,password FROM users# otteniamo il risultato della tabella users.



```
(kali@kali)-[~]
$ john --format=raw-md5 hash.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
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      (?)
abc123        (?)
letmein       (?)
Proceeding with incremental:ASCII
charley       (?)
4g 0:00:00:00 DONE 3/3 (2024-02-28 17:39) 11.42g/s 509022p/s 509022c/s 510668C/s stevy13..chertsu
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

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

Successivamente volendo abbiamo la possibilità di utilizzare john the ripper fornendo un file di hash che abbiamo catturato nel database.